Ministère de l'enseignement Supérieur et de la recherche Scientifique

وزارة التعليم العالي والبحث العلمي

Badji Mokhtar Annaba University
Université Badji Mokhtar – Annaba
Faculté de Technologie

Département Informatique

جامعة باجي مختار – عنابـــة

كـلـيــة التكنـولـوجيـا

قســم الاعلام الالـي

Thèse

Présentée pour obtenir le diplôme de

Doctorat Troisième Cycle

Filière : Informatique

Spécialité : Gestion et Analyse des Données Massives

Par :

Seloua Hadiby

Thème :

La fouille de données à la découverte de motif

Thèse soutenue **le 29 /11/2023** devant le jury composé de :

N°	Nom et prénom	Grade	Etablissement	Qualité
01	MOHAMED BENALI YAMINA	Prof.	Université Badji Mokhtar -Annaba	Rapporteur
02	BAHI HALIMA	Prof.	Université Badji Mokhtar -Annaba	Président
04	AZIZI NABIHA	Prof.	Université Badji Mokhtar -Annaba	Examinateur
05	ABAINIA KHAIREDDINE	MCA.	Université 8 Mai 1945 Guelma	Examinateur

Acknowledgement

First, I would like to acknowledge that this thesis is the culmination of extensive hard work and dedicated study. So, I would like to express my heartfelt appreciation to my supervisor, Professor Mohamed Ben Ali Yamina, who provided me with the opportunity to undertake this thesis and offered invaluable assistance throughout the research process. Without her unwavering support and involvement at every stage, this project would not have come to fruition. I am immensely grateful for her insightful feedback, which has sharpened my thinking. I extend my deepest gratitude for her unwavering support and understanding over the course of the past five years.

Secondly, It is said that no one in this world wishes to see you better than themselves except for your parents. Thank you, dear father, thank you, beloved mother, for everything you have done for me. I reserve a great thanks to my entire family for their unwavering support and encouragement throughout my journey.

Finally, I would like to express my sincere gratitude to the Ministry of Higher Education and the University of Badji Mokhtar-Annaba for giving me the opportunity to pursue my postgraduate studies.

الملخص

لا يزال السرطان أحد أكثر الأمراض فتكًا في عصرنا، حيث يتميز بالنمو غير الطبيعي وانتشار الخلايا في الجسم . تعد المعلوماتية الحيوية أمرًا بالغ الأهمية في تعزيز تطوير الأدوية للسرطان ، وذلك باستخدام البيانات الجينومية والبروتينية لتحديد أهداف الأدوية المحتملة وتصميم العلاجات المستهدفة .تركز هذه الأطروحة على اكتشاف الأدوية ، وتحديداً الفحص الافتراضي باستخدام النماذج التنبؤية لتصنيف الجزيئات النشطة وغير النشطة مع مستقبلات محددة .المستقبلات التي تمت دراستها في هذه الأطروحة هي CDK1 ، بروتين كيناز مفرط التعبير في أنواع مختلفة من السرطان وهدف محتمل لعلاج السرطان ، وBRCA1 ، وهو جين مثبط للورم مهم في الحفاظ على استقرار الجينوم ، مع الطفرات المرتبطة بزيادة مخاطر الإصابة بالثدي والمبيض ، وأنواع السرطان الأخرى .تقترح هذه الدراسة طرقًا جديدة للفحص الافتراضي استنادًا إلى التعلم العميق ، باستخدام البصمات الصيدلانية كوصف جزيئي لبناء نماذج تنبؤية قادرة على التنبؤ بنشاط الجزيئات مع المستقبلات المذكورة .يتم تقييم الطرق المقترحة باستخدام العديد من المقاييس ومقارنتها بالطرق الشائعة الاستخدام في هذا المجال .تظهر النتائج إمكانات واعدة لاستخدام التعلم العميق مع بصمات الأدوية في التنبؤ بنشاط الجزيئات لاكتشاف الأدوية.

الكلمات المفتاحية: اكتشاف الأدوية ، الفحص الافتراضي ، التنبؤ بالنشاط ، التعلم العميق ، بصمة الصيدلة.

Résumé

Le cancer reste l'une des maladies les plus meurtrières de notre époque, caractérisée par une croissance et une propagation anormales des cellules dans le corps. La bioinformatique est essentielle pour faire progresser le développement de médicaments contre le cancer, en utilisant des données génomiques et protéomiques pour identifier des cibles médicamenteuses potentielles et concevoir des thérapies ciblées. Cette thèse se concentre sur la découverte de médicaments, en particulier le criblage virtuel, en utilisant des modèles prédictifs pour classer les molécules actives et inactives avec des récepteurs spécifiques. Les récepteurs étudiés dans cette thèse sont CDK1, une protéine kinase surexprimée dans divers types de cancer et une cible potentielle pour le traitement du cancer, et BRCA1, un gène suppresseur de tumeur crucial dans le maintien de la stabilité du génome, avec des mutations associées à des risques accrus de cancer du sein, ovarien, et d'autres cancers. Cette étude propose de nouvelles approches de criblage virtuel basées sur l'apprentissage profond, utilisant les empreintes digitales de pharmacophores comme descripteurs moléculaires pour construire des modèles prédictifs capables de prédire l'activité des molécules avec les récepteurs mentionnés. Les méthodes proposées sont évaluées à l'aide d'un ensemble de métriques et comparées aux méthodes couramment utilisées dans le domaine. Les résultats montrent un potentiel prometteur pour l'utilisation de l'apprentissage en profondeur avec des empreintes digitales de pharmacophores dans la prédiction de l'activité de molécules pour la découverte de médicaments.

Mots clés : Découverte de médicaments, Criblage virtuel, prédiction d'activité, apprentissage en profondeur, empreinte pharmacophore.

Abstract

Cancer remains one of the deadliest diseases of our time, characterized by abnormal growth and spreading of cells in the body. Bioinformatics is critical in advancing drug development for cancer, utilizing genomic and proteomic data to identify potential drug targets and design targeted therapies. This thesis focuses on drug discovery, specifically virtual screening using predictive models to classify active and inactive molecules with specific receptors. The receptors studied in this thesis are CDK1, a protein kinase over expressed in various types of cancer and a potential target for cancer therapy, and BRCA1, a tumor suppressor gene crucial in maintaining genome stability, with mutations associated with increased risks of breast, ovarian, and other cancers. This study proposes new approaches to virtual screening based on deep learning, using pharmacophore fingerprints as molecular descriptors to build predictive models capable of predicting the activity of molecules with the mentioned receptors. The proposed methods are evaluated using several metrics and compared to commonly used methods in the field. The results show promising potential for using deep learning with pharmacophore fingerprints in predicting the activity of molecules for drug discovery.

Keywords: Drug discovery, Virtual screening, Activity prediction, Deep learning, Pharmacophore fingerprint.

Contents Table

Chapter 3: Virtual Screening for Activity Prediction in Drug Discovery

Contents Table

Tables List

Tables List

Figures List

2DPF	2D Pharmacophore Fingerprint
3DPF	3D Pharmacophore Fingerprint
3D-QSAR	Three-Dimensional Quantitative Structure-Activity Relationship
ACC	Accuracy
ADAM	Adaptive Moment Estimation
ADMET	Absorption, Distribution, Metabolism, Excretion, AndToxicity
Aro	Aromatic
AUC-PRC	Area Under Precision-Recall Curve
AUC-ROC	Area Under The Receiver Operating Characteristic Curve
CDK1	Cyclin-dependent kinase 1
CNN	Convolution Neural Network
DBN	Deep Belief Network
Df	Degree Of Freedom
DNA	deoxyribonucleic acid
DNN	Deep neural network
DT	Decision Tree
FDA	US Food and Drug Administration
FN	False Negatives
FP	False Positives
GB	Gradient boosting
GCN	Graph Convolutional Network
GRU	Gated recurrent units
HA, HBA	Hydrogen Bond Acceptor
HD, HBD	Hydrogen Bond Donor

HTS	high-throughput screening
Hyd	Hydrophobic
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LR	logistic regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCC	Matthews Correlation Coefficient
MD	Molecular dynamics
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NB	Naive Bayes
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
RF	random forests
RMSprop	Root Mean Square Propagation
RNA	The nucleic acid ribonucleic acid
RNN	Recurrent Neural Networks
SEN	Sensitivity
SMILES	Simplified Molecular Input Line Entry System
SP E	Specificity
SVM	Support Vector Method
TN	True Negatives
TP	True Positives

VS	Virtual Screening
XGBoost	Extreme GB

General Introduction

Biological and chemical data collection and storage have been an integral part of the field of biology for many years. As technology has advanced, the amount of data that can be collected has increased exponentially. In recent years, there has been a surge in the amount of data generated from biological and chemical experiments, particularly in genomics, proteomics, and metabolomics. This explosion of data has created a challenge for biologists who struggle to manage, analyze, and interpret the vast amounts of data being generated. Traditional methods of data storage, such as spreadsheets and paper notebooks, have become inadequate to handle this deluge of data. One of the recent techniques to manage and store data is the use of databases, which are electronic repositories that can store large amounts of data in a structured manner. Databases have revolutionized the way biologists store and access biological data, enabling them to store and retrieve data in a more efficient and structured manner.

Bioinformatics has become an essential tool in the development of biology and biotechnology. The use of computational methods and tools to analyze large biological datasets has revolutionized the way scientists study and understand biological systems. Bioinformatics techniques have enabled scientists to make significant strides in understanding complex biological processes, such as DNA sequencing, protein structure and function, gene expression, and the genetic basis of disease. Furthermore, bioinformatics has facilitated the development of new drugs and therapies by enabling researchers to identify potential drug targets and predict drug efficacy. The importance of bioinformatics in the development of science cannot be overstated, as it has played a pivotal role in accelerating our understanding of biological systems and has paved the way for groundbreaking discoveries and innovations in the field of biology.

One of the key applications of bioinformatics in the field of drug discovery is virtual screening. Virtual screening is a computational technique that involves the use of bioinformatics tools to identify potential drug candidates from a vast library of compounds. Virtual screening has revolutionized the drug discovery process by enabling researchers to quickly and efficiently screen large numbers of compounds and identify potential drug candidates more quickly and accurately than traditional screening methods. This approach has significantly reduced the time and cost associated with drug discovery and has led to the development of several important drugs.

Our work is placed within the framework of virtual screening for activity prediction of molecules. Activity prediction is a crucial step in drug discovery and plays a significant role in identifying potential drug candidates. The process of predicting the biological activity of molecules involves using computational tools and techniques to model the interaction between a molecule and a target protein. This enables researchers to identify compounds that have a high affinity and specificity for a target protein and are likely to have a therapeutic effect. Activity prediction is particularly important in the early stages of drug discovery when researchers need to screen large libraries of compounds to identify potential drug candidates. Without accurate activity prediction models, the drug discovery process would be slow and

inefficient, as researchers would need to synthesize and test each compound individually, which is time-consuming and expensive. Furthermore, accurate activity prediction models can help reduce the number of compounds that need to be tested, thereby minimizing the risk of adverse side effects and increasing the likelihood of success in clinical trials. In summary, activity prediction is a critical component of drug discovery and has played a significant role in the development of new and effective drugs.

Machine learning methods have been widely used in virtual screening for activity prediction, enabling researchers to accurately predict the biological activity of new compounds against a specific target protein. Machine learning algorithms use statistical models to identify patterns and relationships between molecular features and biological activities. These models can be trained on large datasets of molecular structures and their corresponding activities, enabling them to learn from complex, high-dimensional data and make accurate predictions. Machine learning methods have been shown to outperform traditional methods for activity prediction, as they can capture complex nonlinear relationships between molecular descriptors and biological activities. Moreover, machine learning models can be continuously improved and refined as more data becomes available, leading ening for activity prediction has significantly accelerated the drug discovery process, enato more accurate predictions over time. The use of machine learning methods in virtual screbling researchers to screen large libraries of compounds efficiently and identify potential drug candidates.

1. Problem and objectives

In the field of cancer research, two genes, CDK1 and BRCA1, are of particular interest due to their critical role in cancer development. CDK1, a protein kinase that regulates cell cycle progression, is overexpressed in many types of cancer and is a potential target for cancer therapy. However, targeting CDK1 presents several challenges, including the risk of off-target effects, toxicity, and drug resistance. Similarly, BRCA1, a tumor suppressor gene that plays a crucial role in maintaining genome stability, is also a promising target for drug development. However, the complexity of the BRCA1 pathway and the interplay between various signaling pathways in cancer cells make it difficult to identify specific targets for drug development.

Traditional methods for drug discovery, such as high-throughput screening, have limitations, including a low success rate, a focus on a single target or pathway, and a lack of diversity in chemical structures available for screening. Virtual screening is a promising approach for identifying potential drugs targeting CDK1 and BRCA1. However, this approach also presents challenges, including the need for large and diverse datasets, the selection of appropriate molecular descriptors, and the optimization of machine learning algorithms.

The objective of our work is to develop novel approaches for virtual screening based on pharmacophore fingerprints using deep learning to predict potential new drugs targeting the CDK1 and BRCA1 genes. Pharmacophore fingerprints are a type of molecular descriptor that captures the key chemical patterns and features of a molecule, such as functional groups,

shapes, and electrostatic potentials that are important for its biological activity. Unlike other types of descriptors, such as structural fingerprints, which encode only the presence or absence of certain molecular features, pharmacophore fingerprints represent the spatial arrangement of these features in two or three dimensions. This allows for a more accurate representation of the molecular interactions that are involved in biological activity. We aim to leverage the large amount of molecular data available in well-known chemical libraries to develop accurate predictive models. So we have proposed two approaches using deep learning with 2D pharmacophore fingerprints for virtual screening to predict the activity of molecules with CDK1, Additionally, we have proposed two other approaches using deep learning with 3D pharmacophore fingerprints to predict the activity of molecules with both CDK1 and BRCA1 genes. In order to evaluate and compare the performance of the proposed predictive models, we have made comparisons with the most used machine learning methods in this field.

2. Document content

This document is organized into five chapters, the first three of which are theoretical and include, respectively, an introduction to bioinformatics, Data Mining and Classification Models, and Virtual Screening for Activity prediction in Drug discovery**.** The last two chapters concern the experimental study carried out, which consists of the prediction of the biological activity of chemical compounds using 2D pharmacophore fingerprint in chapter four and 3D pharmacophore fingerprint in final chapter.

Chapter 1. Introduction to Bioinformatics

This chapter provides an introduction to the interdisciplinary field of bioinformatics, including its definition and historical background. It discusses the components of bioinformatics, such as data, databases, and database mining tools and techniques. The chapter explores the applications of bioinformatics in life sciences and technologies, such as genomics and drug discovery. It also covers the challenges and limitations of bioinformatics, including the need for advanced computational methods and the ethical and legal issues surrounding its use.

Chapter 2. Data Mining and Classification Models

This chapter covers the fundamental concepts and techniques of data mining and classification models. We begin with an explanation of data mining and its role in today's world of big data. We then discuss various data mining techniques, which are: supervised learning techniques (classification and regression) and unsupervised learning techniques (association rules, clustering, and anomaly detection). Data preprocessing for mining is also explored, covering the methods of data cleaning, integration, transformation, and reduction. Lastly, we delve into classification models that are commonly used in data mining, such as decision trees, logistic regression, support vector machines and neural networks . This chapter

serves as an introduction to the key concepts and techniques used in data mining and classification, providing a foundation for understanding how to analyze and interpret large datasets.

Chapter 3. Virtual Screening for Activity prediction in Drug discovery

This chapter focuses on virtual screening as a powerful tool for predicting the activity of drug candidates in the early stages of drug discovery. We begin with an overview of the drug discovery process, and then introduce the concept of virtual screening and its goals, discussing the various categories of virtual screening techniques and the molecular descriptions used in this process. The applications of virtual screening in drug discovery are then presented, along with a summary of the most important work using machine learning in virtual screening for activity prediction. Finally, we discuss the challenges and limitations of virtual screening, such as the accuracy of the prediction models and the need for more effective screening methods. , this chapter provides a comprehensive overview of the virtual screening process, its applications, and its potential as a tool for accelerating the drug discovery process.

Chapter 4. Deep Learning – based Virtual screening using 2DPF for Activity prediction

This chapter is devoted to our first main contributions in the field of virtual screening for activity prediction in drug discovery, which involves the use of deep learning-based approaches using 2D pharmacophore fingerprints. Our study focused on the cyclin-dependent kinase 1 (CDK1) receptor. The chapter begins by discussing the reasons behind the selection of CDK1 as our target receptor, including its importance in drug discovery and the limited research on the use of machine learning methods for activity prediction with this receptor. We then present our first proposed approach for virtual screening using 2D pharmacophore fingerprints, which includes the generation of the fingerprints, feature selection, and two predictive models: a deep neural network and a convolutional neural network. The experimental results, including data sets and overall performance, are also presented. Finally, we discuss our second proposed approach, which builds on the first and further improves the predictive model. The chapter concludes with a discussion of the limitations and potential future directions of our work.

Chapter 5. Deep Learning-Based Virtual screening using 3DPF for Activity Prediction

This chapter focuses on our second main contributions, which is the application of deep learning-based virtual screening using 3D pharmacophore fingerprints for activity prediction in drug discovery. We begin by discussing the importance of 3D structures of molecules and their conformations. We then present our first proposed approach, which aims at activity prediction with CDK1. This approach involves the generation of a 3D pharmacophore fingerprint, the selection of discriminating pharmacophores, and the architecture of the CNN model and the obtained experimental results. Next, we discuss our second proposed approach, which is a deep neural network-based virtual screening for activity prediction with BRCA1 using the same kind of fingerprint. We begin by introducing the breast cancer gene (BRCA1),

its importance in drug discovery, and the proposed approach for activity prediction with BRCA1. This approach involves the generation of a pharmacophore model, the generation of 3D pharmacophore fingerprint based on the proposed model, and the architecture of the proposed predictive model. We also provide experimental results for this approach. this chapter demonstrates the potential of using 3D pharmacophore fingerprints and deep learning-based methods for activity prediction in drug discovery.

Chapter 1
Introduction to bioinformatics

Bioinformatics is the application of information processing techniques to the management of biological data; bioinformatics encompasses a wide range of tools and methodologies which plays a critical role in advancing research in various scientific fields by enabling the analysis and interpretation of large and complex biological datasets.

This chapter offers an introductory overview of the interdisciplinary field of bioinformatics, it is structured as follows: the second section talks about the definition, historical background, and goals of bioinformatics. The third section presents the main components of bioinformatics including data and techniques. In the fourth section, we mention the most applications of bioinformatics in life sciences and technologies. The challenges and limitations of bioinformatics are covered in section five.

1. 1. Introduction

The great and rapid advances in the field of molecular biology and chemistry have led to exponential growth in the amount of biological and chemical data. Two main problems have emerged: First, the efficient storage of this information, and second, its processing in order to extract useful knowledge. This situation has led the scientific community in both fields to think about finding the technical means necessary to store, process, analyze, and interpret the enormous amount of information after the old manual methods became unable to keep up with this rapid data growth. Eventually, a new science called bioinformatics was founded resulted from the fusion of biology and information technology. Bioinformatics depends on the use of computer science and the exploitation of various computational methods to obtain new biological knowledge. It has combined many disciplines such as molecular biology, genetics, mathematics, and statistics to be able to progress. At first, his tasks were as simple as analyzing sequences through comparisons, later, his tasks evolved in parallel with the increase in various biological data and the use of modern methods to deal with heterogeneous data, perform complex analyzes, and obtain accurate predictions. Bioinformatics deals with different types of data, the most important: DNA sequences, protein sequences and structures, RNA sequences, Genomic sequence tags, Expressed sequence tags. Bioinformatics is a multidisciplinary field used in many life sciences, so it has numerous applications, like Medicine, pharmacology, Genetics, Agriculture, Livestock, and Wastes. Bioinformatics is a rapidly growing field, and its applications are likely to continue expanding in the future as more biological data becomes available and new technologies are developed for analyzing and interpreting that data.

In this chapter, we will provide an overview of the key concepts, and techniques in bioinformatics, the types of data, and the most important databases used in this field. We will also introduce some of the common database tools such as BLAST, GenBank, and the Protein Data Bank. Finally, we will mention some important applications of bioinformatics in life sciences and technologies and the most challenges and limitations of bioinformatics.

1. 2. A historical and general vision of bioinformatics

1. 2. 1. What is bioinformatics?

The field of bioinformatics is interdisciplinary and combines computer science with mathematics, statistics, chemistry, and engineering (Figure 1.1). It strives to create methods for storing,

retrieving, analyzing, integrating, exploiting, understanding, and interpreting biological data by using informatics methods and concepts. Computer studies of biological data are used to derive the knowledge. It is used in a wide range of fields of biology and medicine in the actual world.

1. 2. 2. A brief Historical background of bioinformatics

Despite not using the name "bioinformatics" to characterize their work, leading researchers of computational biology had a clear vision for how computing technology, mathematics, and molecular biology could be productively combined to address basic challenges in the life sciences [1]. The beginning of the 1960s saw the availability of computers for academic and scientific reasons, which gave rise to the field of bioinformatics. At that time, scientists started using computers to attempt to provide answers to fundamental questions in biology. At the time, a woman named Margaret Dayhoff, an experienced researcher, suggested using mathematical techniques to examine mutation probability and amino acid frequency in biological sequences. In the period 1945-1955, the idea that proteins carry information encoded in a linear sequence of amino acids appeared which is a great achievement in protein biochemistry, as Frederick Sanger and his colleagues [2] were able to extract the first successful sequence of a complete protein, which are insulin Using experimental methods to sequence protein microstructures. In this time, the development of computational biology was aided by three crucial reasons .First, the availability of the source of data, which is a huge amount of sequences of amino acids and many questions raised about them, which were impractical to process manually. Second, the notion that macromolecules contain information became a key tenet of the molecular biology conceptual framework. Third, the lack of computers was no longer a significant barrier to the advancement of computational biology and the rise of the Internet and supercomputers [1].

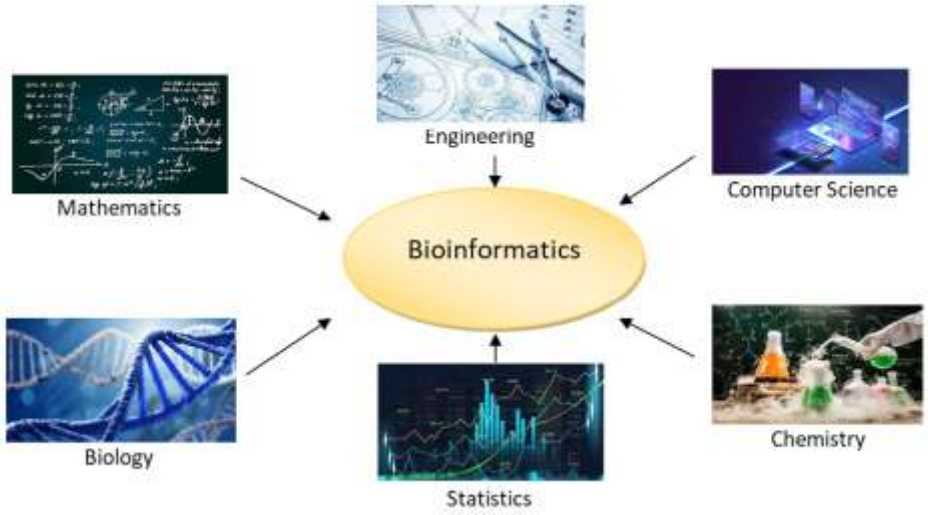


Figure1.1. Concept of bioinformatics.

Since the seventies of the last century, there have been techniques for sequencing DNA; but, these techniques were costly and time-consuming. For example, the early 1990s saw the commencement of the project of the human genome, and it wasn't officially finished until 2003. Although this project was very useful and it provided us with a genetic blueprint of the human being, sequencing

was expensive [3]. Thereafter, clever tactics and computational methods were combined and promising results were achieved, including: shotgun sequencing appeared in1970, the whole nucleotide sequence of the Haemophilus influenzae bacteria is deciphered successfully in 1995 [4], and the complete genome sequence of Drosophila melanogaster was obtained in 2000 [2]. The creation of Next-Generation Sequencing technologies and the appearance of other high-throughput technologies such as microarray and RNA-Seq were the results of all these developments. As a result, modern bioinformatics has become mainly focused on creating new models, algorithms, and tools based on dealing with biological data using computational methods, after it was previously only a tool for biological analysis in order to gain new knowledge.

1. 2. 3. Goals of bioinformatics

A variety of sciences are combined in the interdisciplinary field of bioinformatics to address complicated biological issues. Among the objectives of bioinformatics are:

- Develop computational tools and methods for organizing, analyzing, and interpreting large-scale biological data.
- Help with genomic data analysis, including finding genetic variants and annotating genes for their functions.
- Support the development of new drugs and therapies by enabling the identification of new drug targets and the optimization of drug efficacy and safety.
- Facilitate the analysis of proteomic and metabolomic data, which can provide insights into the biochemical processes that occur within cells and tissues.
- Analyze environmental and ecological data, such as metagenomics data, which can provide insights into the diversity and function of microbial communities.
- Provide researchers and clinicians with the tools and insights they need to make sense of complex biological data, and to develop new treatments and therapies for a wide range of diseases.

The goal of bioinformatics is to use computational and statistical methods to gain insights into biological systems and solve complex biological problems.

1. 3. Components of bioinformatics

Bioinformatics comprises three components: Data, Database, Database Mining tools and techniques (Figure 1.2). Before detailing the types data, three main concepts must be explained which are: atom, molecule, and macromolecule, because everything that follows is based on these three concepts.

A. Atom

Atoms are the tiniest components of one body that may chemically combine with other atoms (Figure 1.3). They are the basic building blocks of all substances, whether solid, liquid, or gaseous. Its positively charged nucleus, made composed of protons and neutrons, is surrounded by a cloud of negatively charged electrons that orbit it in shells or energy levels [5].

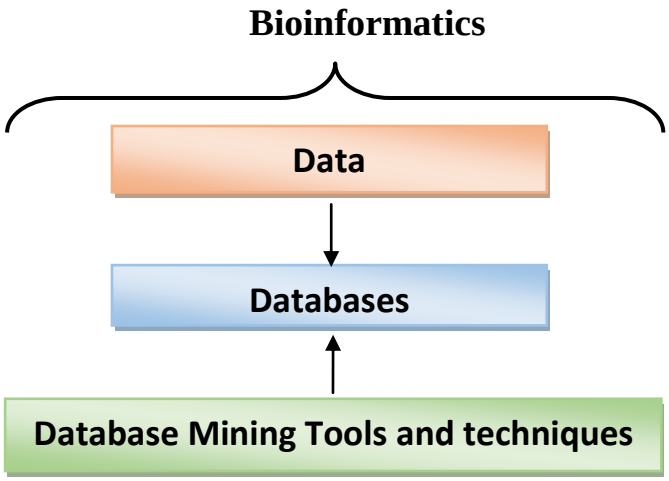


Figure 1.2. Components of bioinformatics.

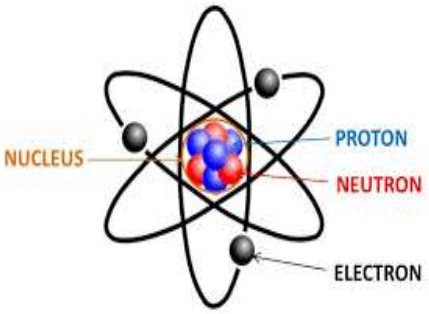


Figure 1.3. Components of atom [6].

B. Molecule

Molecules possess the chemical and physical characteristics of matter and are comprised of at least two atoms that are bonded together. These atoms may be identical or different from one another [7]. For instance, in the caffeine molecule (C8H10N4O2) illustrated in Figure 1.4, the atoms present include carbon (C), hydrogen (H), nitrogen (N), oxygen (O), and a methyl radical (CH3). It is important to note that molecules represent the smallest component of matter that retains the properties of the material it originates from.

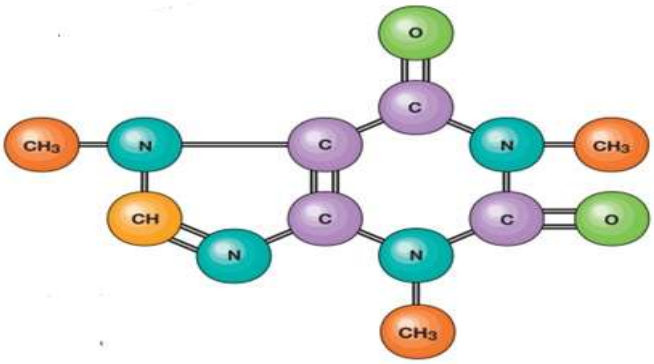


Figure 1.4. Caffeine molecule [8].

There are four major primary types of bonding: ionic, covalent, metallic, and Hydrogen bond

• **Ionic bond:** As illustrated in Figure 1.5, Ionic bonds between two or more atoms are created when one or more electrons are transferred between them [9].

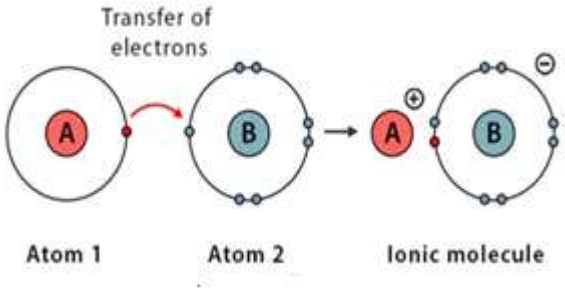


Figure 1.5. Ionic bond [10].

• **Covalent bond:** The increased probability of finding electrons between two atoms as a result of electron sharing is known as a covalent bond (Figure 1.6), [9].

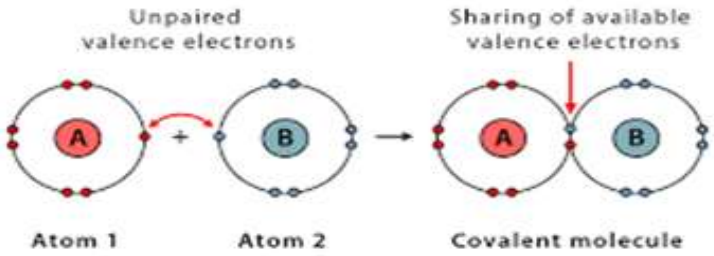


Figure 1.6. Covalent bond [11]

• **Metallic bond:** The force that holds the atoms in a metallic substance together is known as the metallic bond (Figure 1.7), [9].

• **Hydrogen bond:** An electronegative atom from another molecule, such as nitrogen, oxygen, or fluorine, interacts with a hydrogen atom to form a hydrogen bond, which requires the hydrogen to build a covalent bond with an additional electronegative atom. These bonds may exist inside the same molecule or between different molecules [13]. Figure 1.8 represents an example of hydrogen bond.

C. Macromolecules

Molecules with at least several tens of atoms are called macromolecules or polymers. In many biological processes, macromolecules are crucial building blocks of living things. These large molecules are composed of smaller building blocks, or monomers, that are linked together by covalent bonds to form long chains. There are four main types of macromolecules found in biological systems: proteins, nucleic acids, carbohydrates, and lipids. Nucleotides are the building

blocks of nucleic acids, while sugars, amino acids, glycerol, and fatty acids are the components of carbohydrates, proteins, and lipids respectively. Each of these classes of macromolecules has unique functions and properties that are critical for the survival and function of living organisms [14].

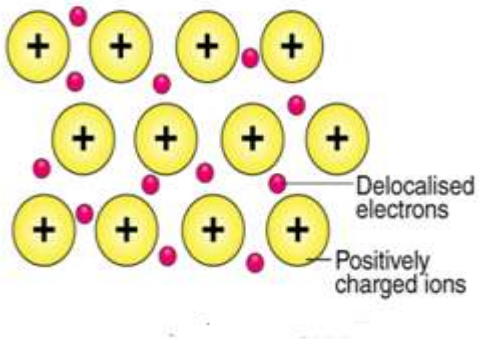


Figure 1.7. Metallic bonding in sodium [12].

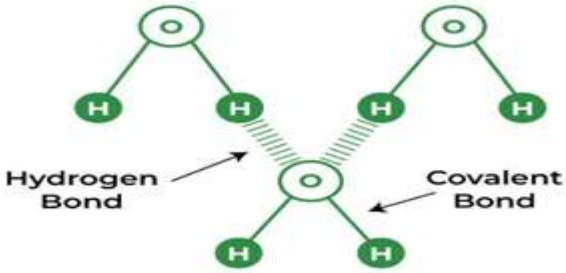


Figure 1.8.Hydrogen bonding in water (H2O) [13].

1. 3. 1. Data in bioinformatics

The types of data used in the field of bioinformatics are numerous, the most important of which are the following:

1.3.1.1. Nucleic acid sequences

A. DNA sequences

A biological macromolecule known as deoxyribonucleic acid, or DNA, is found in almost all cells as well as many viruses (Figure 1.9). All of the genetic data, also referred to as the genome, or all of the information required for the creation and growth of a living being, is contained in the DNA. Proteins can be produced thanks to DNA [15].

The double strands of DNA combine to form a double helix. Bacteria fall into the prokaryote category, which has DNA in a circular shape, and eukaryotes, which contain DNA in fragment form. In other words, it manifests as chromosomes. Three components make up the DNA nucleotide:

- A group of phosphate atoms.
- Deoxyribose, a 5-carbon sugar (pentose).
- One of the following nitrogenous bases: cytosine (C), thymine (T), adenine (A), or guanine (G).

Thus, a polynucleotide is formed by the sequential binding of 4 different types of nucleotides.

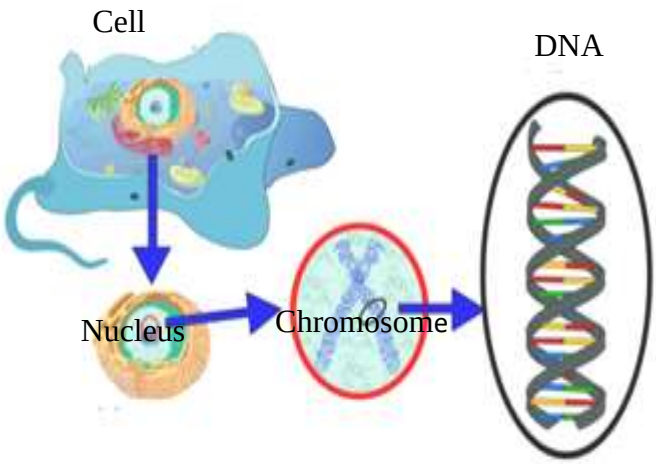


Figure 1.9. DNA sequence [16].

The pentose-phosphate backbone, which makes up the outside margins of the helix, is one of the two strands that make up DNA (Figure 1.10). The nitrogenous bases found inside are linked two by two in a complementary manner, meaning that adenine can only bind to thymine and cytosine can only attach to guanine.

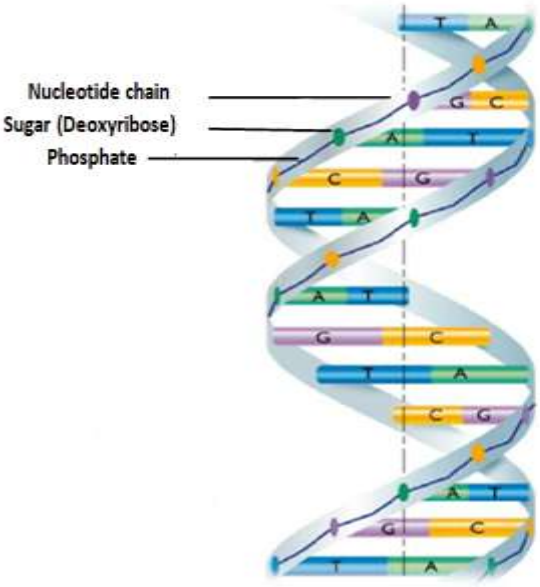


Figure1.10. DNA double helix [17].

The fundamental property of DNA is replication where DNA is able to copy itself faithfully. DNA splits into two strands, with each separated strand serving as a template to make a complementary strand. The result: two new DNA molecules, each with an old and a new strand (Figure 1.11).

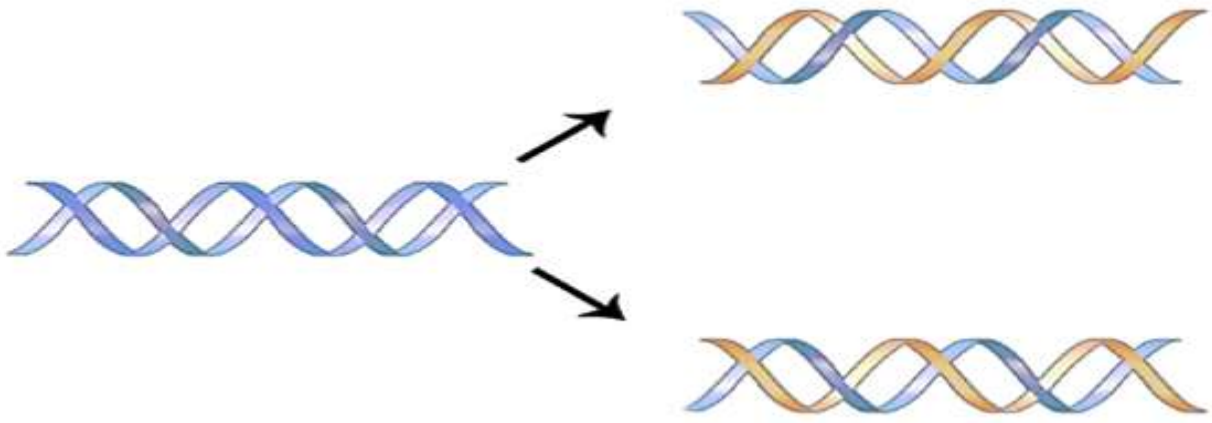


Figure 1.11. DNA replication [18].

B. Expressed sequence tags (ESTs)

Expressed sequence tags (ESTs) are short DNA sequences that are generated from complementary DNA (cDNA) fragments (Figure1.12). ESTs represent a partial sequence of a transcribed gene and are often used to identify and characterize genes in different organisms. By comparing EST sequences to known genes, researchers can identify new genes, discover alternative splicing variants, and study gene expression patterns in different tissues and under different conditions [19]. EST sequencing has been widely used in many areas of research, including functional genomics, gene discovery, and comparative genomics. Furthermore, EST databases have been created for many organisms, including humans, mice, and plants, and these databases are valuable resources for researchers to study gene expression and function.

C. RNA sequences

The nucleic acid ribonucleic acid (RNA) is essential for the control and expression of genes (Figure 1.13). There are three different types of RNA: messenger RNA (mRNA), transfer RNA (tRNA), and ribosomal RNA (rRNA), each of which plays a specific role in the expression of genes. mRNA transports the genetic information from DNA to the ribosome, where it is translated into a protein. By means of tRNA, amino acids are delivered to the ribosome where they are connected to the growing protein chain. An element that facilitates the formation of peptide bonds between amino acids is the ribosome's rRNA [21]. Long non-coding RNAs (lncRNAs) and tiny non-coding RNAs (miRNAs), such as microRNAs, are two examples of non-coding functions of RNA. RNA is a crucial part of many biological processes and serves a key function in the control of gene expression.

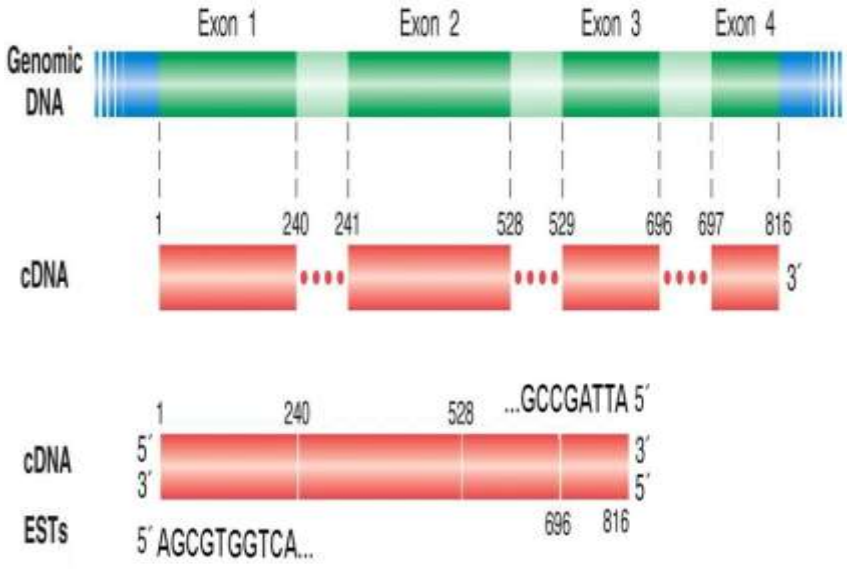


Figure 1.12. Expressed sequence tags [20].

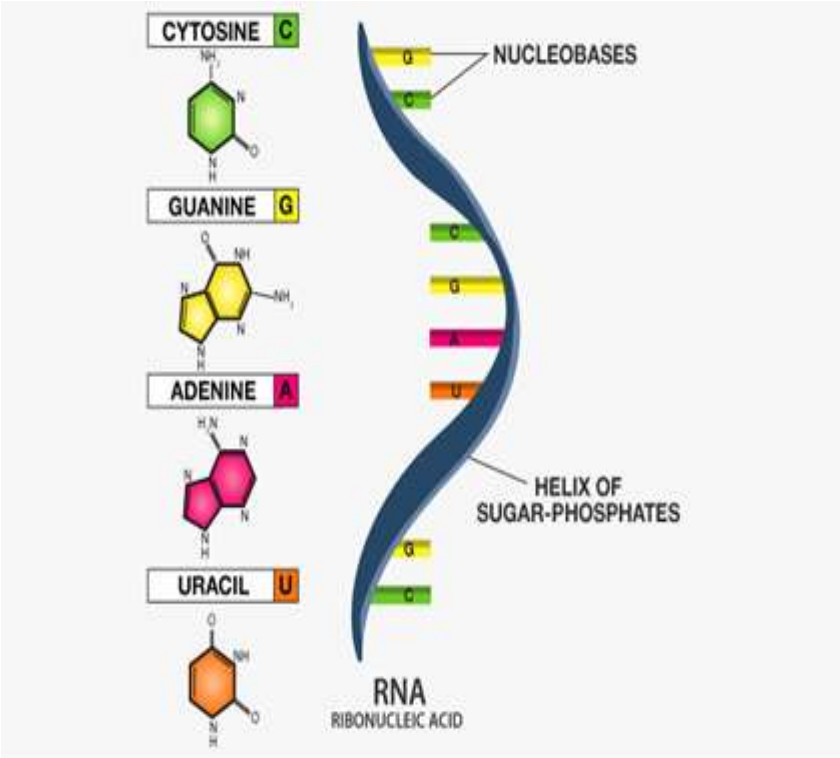


Figure 1.13. ribonucleic acid [22]

The two forms of nucleic acids, DNA and RNA, both play crucial roles in the storage and expression of genetic information. The following are some of the main distinctions between DNA and RNA [23]

• **Sugar molecules**: The sugar molecule contained in DNA is called deoxyribose, whereas ribose is found in RNA. The distinction is the presence or absence of an oxygen atom on the 2' carbon of the sugar ring.

• **Nitrogenous bases**: Nitrogenous bases are found in the nucleotides that make up DNA and RNA. The four nitrogenous bases that can be found in DNA are adenine (A), cytosine (C), guanine (G), and thymine (T). A, C, and G are the initial three bases in both DNA and RNA, however uracil (U) is used in place of thymine.

• **Structure**: RNA typically consists of a single strand, whereas DNA is a double-stranded helix. The complementary base pairs (A-T and C-G) in DNA establish hydrogen bonds that hold the two strands together, whereas RNA folds back on itself to produce different secondary structures.

• **Purpose**: During cell division, DNA is copied in order to transmit genetic information to daughter cells. DNA is responsible for storing genetic information. Numerous biological functions, such as protein synthesis, gene control, and the catalysis of biochemical reactions, depend on RNA.

• **Stability**: Due to the additional oxygen atom in RNA's ribose sugar, which makes RNA more prone to breakdown, DNA is often more stable than RNA.

• **Location**: DNA is normally found in the nucleus of eukaryotic cells, whereas RNA is present in both the cytoplasm and nucleus of cells. One such kind of RNA is mRNA, which carries genetic material from the nucleus to the cytoplasm for protein production.

1.3.1.2. Protein sequences

In humans, there are 23 pairs of chromosomes. The double helix, depending on the cell multiplication cycle, can compact very strongly to allow cell division and the production of 2 identical cells with the same DNA composition, or relax very strongly and thus allow transcription and therefore protein production. Peptide bonds connecting amino acid residues make form the macromolecules known as proteins. Living beings produce substances called biomolecules. As a result, the majority of them are organic substances. The protein is one of the primary biomolecules. Lipids, nucleic acids, and carbohydrates—particularly polysaccharides—make up the remaining three. Proteins are made of carbon, hydrogen, oxygen, sulfur, nitrogen, and phosphorus. [24].

A. Amino acids

Amino acids are the fundamental components of proteins, essential macromolecules found in all living things. They are organic compounds comprised of an organic component, an amino group (-NH2), a carboxyl group (-COOH), and a side chain specific to each amino acid (Figure 1.14). There are two types of amino acids: essential and non-essential, and there are 20 of them that are frequently utilized to construct proteins. A protein's amino acid sequence is defined by the nucleotide sequence in the relevant gene, and modifications to this sequence can have an impact on the protein's structure and functionality. Amino acids are classified based on their chemical

properties into polar, non-polar, acidic, and basic amino acids, which contribute to the overall structure and function of the protein, they are also important in many non-protein functions, such as serving as neurotransmitters, antioxidants, and precursors for other biomolecules , amino acids are also important in many non-protein functions, such as serving as neurotransmitters, antioxidants, and precursors for other biomolecules [25]. Deficiencies in amino acids can lead to a variety of health problems, including growth retardation, immune dysfunction, and neurological disorders [26]. Amino acids play a critical role in many biological processes, and their importance extends beyond their role as protein building blocks.

B. From DNA to protein

As we mentioned earlier, DNA consists of a double chain of bases, and the bonds between these bases are like a software code that Bill Gates described as more complex than any programs written by humans. The most important functions of these programs are the manufacture of proteins and the transfer of genetic traits; in addition to other functions that science has discovered some of and is still working to discover more. Protein manufacturing is one of the wonders of software in cells (Figure 1.15). The process of going from DNA to protein is known as gene expression, which involves several steps [28]:

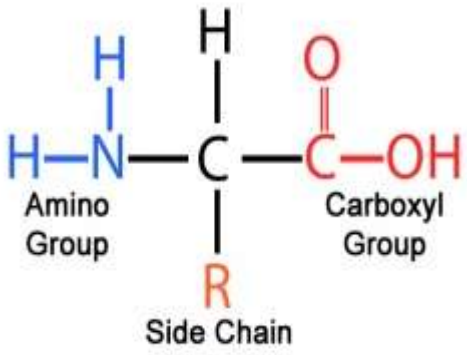


Figure 1.14. Configuration general of an amino acid [27].

• **Transcription**: The initial stage of gene expression, transcription involves copying a gene's DNA sequence into a messenger RNA (mRNA) molecule. The RNA polymerase enzyme reads the DNA sequence and creates a complementary RNA sequence in the cell's nucleus to do this.

• **RNA processing**: The mRNA molecule is then changed by splicing the exons together and deleting the introns, which are non-coding areas. After that, the mRNA molecule is moved from the nucleus into the cytoplasm.

• **Translation**: In the cytoplasm, ribosomes read the mRNA molecule and employ the data it contains to make a protein. The assembling of amino acids into a polypeptide chain in accordance with the arrangement of codons in the mRNA is the process known as translation.

• **Protein folding**: After the polypeptide chain is finished, it folds into the final three-dimensional shape, which is crucial to the protein's functionality.

• **Post-translational modifications**: The protein may additionally be altered after it has been translated, for as by the addition of sugar or lipid molecules, which may impact its stability or function.

C. Types of protein

Proteins are complex organic molecules that are essential for many biological functions. There are several types of proteins that play different roles in the body. Here are some of the major types of proteins [30]:

• **Enzymes:** Proteins that catalyze chemical reactions in the body are known as enzymes. They are essential for numerous biological functions and aid in accelerating metabolic activities.

• **Structural proteins:** These proteins give tissues and cells structure and support. Collagen, elastin, and keratin are a few illustrations of structural proteins.

• **Transport proteins:** These proteins aid in the movement of ions and chemicals via cell membranes. One transport protein that carries oxygen in the blood is hemoglobin.

• **Hormones:** Hormones are signaling molecules that are produced by the endocrine system. They regulate many bodily functions, such as growth and development, metabolism, and reproduction.

• **Antibodies:** Antibodies are proteins that help the immune system to identify and neutralize foreign substances, such as viruses and bacteria.

• **Contractile proteins:** These proteins are responsible for muscle contraction and movement. Examples include actin and myosin.

• **Storage proteins:** Some proteins are used to store nutrients, such as casein in milk, which provides amino acids for the growth and development of young animals.

• **Regulatory proteins:** These proteins help to control gene expression and other cellular processes. Examples include transcription factors, which bind to DNA and control the transcription of genes.

Proteins play a vital role in the functioning of cells and the body as a whole, and their diverse range of functions underscores their importance in maintaining health and well-being.

1.3.1.3. Protein structures

There are four stages of protein structure: primary, secondary, tertiary, and quaternary (Figure 1.16). A protein's structure dictates its function. With the aid of experimental methods, such as X-ray crystallography, NMR spectroscopy, and cryo-electron microscopy, protein structures can be ascertained. High-resolution pictures of the protein structure are produced by these methods, which can be utilized to comprehend protein function and create medications that specifically target particular proteins.

A. Primary structure

The linear arrangement of amino acids that make up the protein chain is referred to as the fundamental structure of a protein. The main structure, which determines the three-dimensional structure and function of the protein, is defined by the nucleotide sequence of the associated gene. The genetic code determines the amino acid sequence, with each amino acid being specified by a codon, which is a sequence of three nucleotides. Each protein has a distinct sequence of amino acids, and the main structure can be anywhere between a few and thousands of amino acids long [31]. The function of a protein can be significantly impacted by even minor alterations to its fundamental structure. Understanding the fundamental structure of proteins is crucial for comprehending their biological significance since it plays a significant role in determining how proteins operate

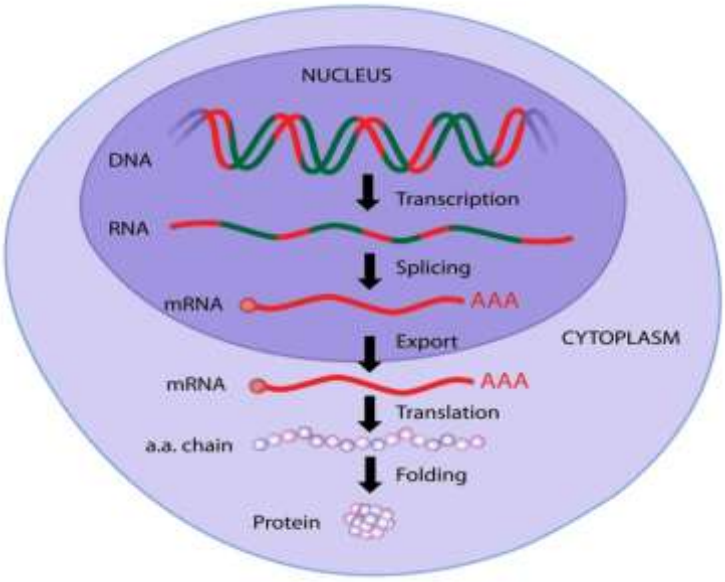


Figure 1.15. Protein synthesis [29].

B. Secondary structure

The local folding patterns of a protein that are kept stable by hydrogen bonds between the backbone atoms of amino acids are referred to as secondary structure. Alpha helices and beta sheets, which are both created by recurrent patterns of hydrogen bonding between the backbone atoms of nearby amino acids, are the two most prevalent types of secondary structure. Beta sheets are stretched structures with strands running in parallel or antiparallel directions, while alpha helices are coiled structures with a right-handed twist. These structures are crucial in defining the stability and overall three-dimensional form of the protein. The protein's amino acid sequence can be used to predict secondary structure, but experimental techniques like X-ray crystallography or NMR spectroscopy are required to determine the protein's precise structure [32]. For the purpose of generating medications that specifically target particular protein structures and for understanding how proteins operate biologically, it is essential to understand their secondary structure.

C. Tertiary structure

A protein's overall three-dimensional shape, which is produced by interactions between its side chains and the folding of its secondary structures, is referred to as its tertiary structure. Numerous non-covalent interactions, including hydrogen bonds, hydrophobic interactions, electrostatic interactions, and van der Waals forces, are responsible for the folding process and aid in stabilizing the finished structure. The tertiary structure, which governs the protein's surface characteristics, active site, and interaction partners, is crucial in establishing the biological activity of the protein [33]. The tertiary structure of a protein can be determined experimentally using methods like X-ray crystallography, NMR spectroscopy, and cryo-electron microscopy, and the structure can also be predicted computationally using the amino acid sequence of the protein. It's crucial to comprehend a protein's tertiary structure in order to comprehend how it works and to create medications that specifically target particular proteins.

D. Quaternary structure

Multiple protein subunits arranged in a larger protein complex are referred to as having quaternary structure. Non-covalent interactions like as hydrogen bonds, salt bridges, and hydrophobic interactions hold the individual subunits together, resulting in the formation of a stable complex with a specific three-dimensional shape. An essential component of a protein's function is its quaternary structure, which has an impact on the protein's stability, activity, and interactions with other molecules. Numerous proteins that carry out important biological tasks, including enzymes, ion channels, and antibodies, are made up of numerous subunits organized in a certain pattern [33]. Using methods like X-ray crystallography or cryo-electron microscopy, the quaternary structure can be identified experimentally. Understanding a protein's quaternary structure is crucial for comprehending its biological function and can assist in identifying possible targets for medications that can interfere with protein-protein interactions.

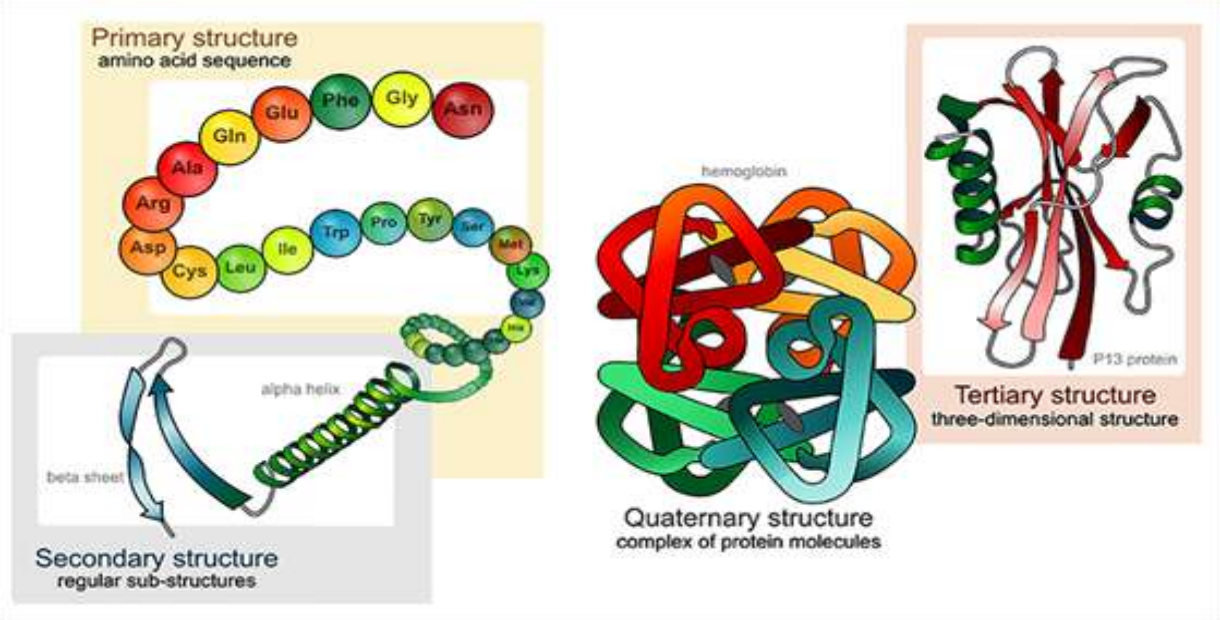


Figure 1.16. Protein structures [34].

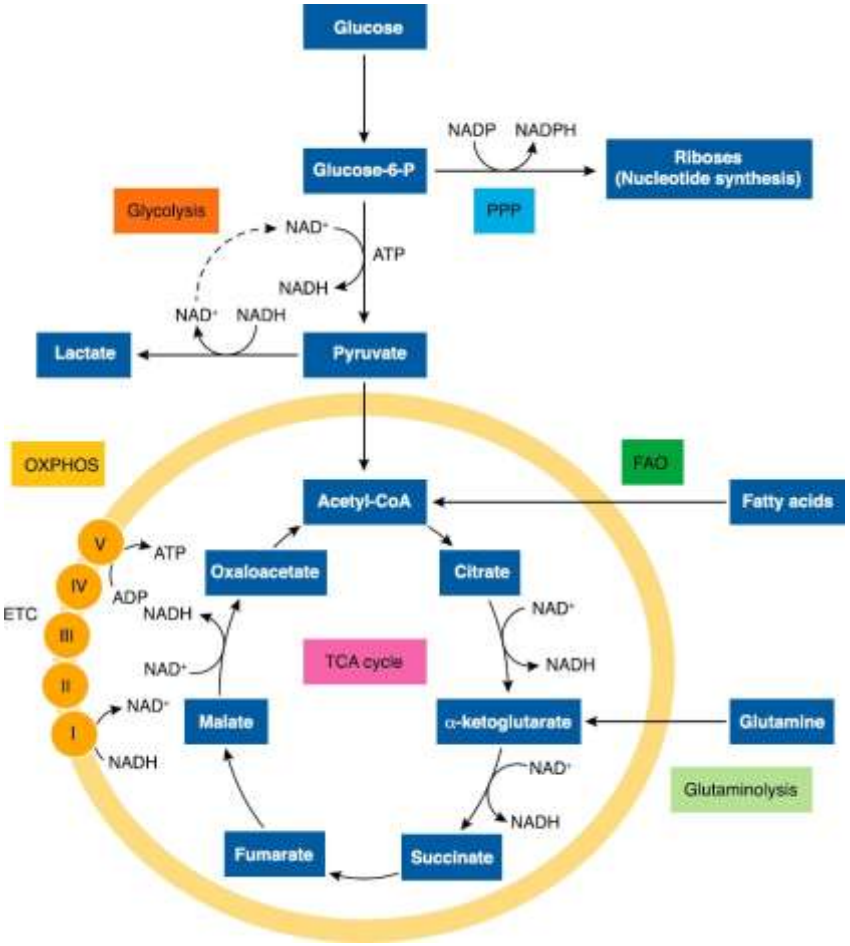


Figure 1.17. Metabolic pathways [37]

1.3.1.4. Metabolic pathways

Metabolic pathways are complex sequences of chemical reactions that occur within cells to convert nutrients into energy and other molecules that are necessary for cell function and survival (Figure 1.17). There are many different metabolic pathways, each of which is responsible for the synthesis or breakdown of specific molecules. These pathways are tightly regulated to ensure that the correct molecules are produced in the correct quantities, and that energy is efficiently extracted from nutrients [35]. Metabolic pathways are classified into two broad categories: catabolic pathways, which break down molecules to release energy, and anabolic pathways, which use energy to synthesize new molecules [36]. One example of a catabolic pathway is glycolysis, which breaks down glucose to produce energy in the form of ATP. An example of an anabolic pathway is the synthesis of nucleotides, which requires energy input from ATP. Metabolic pathways are highly interconnected, with the products of one pathway often serving as the starting materials for another pathway. The study of metabolic pathways is important for understanding basic cellular processes, as well as for developing new drugs and therapies to treat metabolic disorders.

1. 3. 2. Databases in bioinformatics

Databases play a crucial role in bioinformatics as they store, organize, and make available large amounts of biological data. Bioinformatics databases can be broadly classified into several categories based on their content:

1. 3. 2. 1. Sequence databases

Sequence databases are collections of genetic or protein sequences that have been obtained through experimental techniques or computational predictions. These databases are essential tools for researchers studying genomics, proteomics, and related fields. The most widely used sequence databases include:

• **GenBank**: It is the primary archive for nucleotide sequences managed by the National Center for Biotechnology Information of the National Institutes of Health. It is continually updated with new DNA sequences from various sources and has approximately 300 billion base pair sequences.

• **UniProt** :a comprehensive database of protein sequences and functional information, is managed by the European Bioinformatics Institute, the Swiss Institute of Bioinformatics, and the Protein Information Resource. It contains more than 200 million protein sequences from various organisms.

• **Protein Data Bank:** PDB is a database of complex assemblies, proteins, and nucleic acids in three dimensions. It has around 170,000 structures and is maintained by the Worldwide Protein Data Bank.

• **RefSeq:** Maintained by the NCBI, RefSeq is a curated library of reference sequences for genes, transcripts, and proteins. It offers a uniform and consistent collection of sequences for use in clinical testing and genetic research.

• **Ensembl:** itl is a genome annotation database that offers details on the organization, regulation, and function of genes for a variety of organisms. It is kept up by the Wellcome Trust Sanger Institute and the European Molecular Biology Laboratory's European Bioinformatics Institute (EMBL-EBI).

These databases are regularly updated with new sequences and annotations, and they are searchable and accessible through various tools and web interfaces, making them valuable resources for researchers in the life sciences.

1. 3. 2. 2. Structural databases

Data about the three-dimensional (3D) structures of biological molecules, such as proteins, nucleic acids, and other biomolecules, is kept in databases referred to as structural databases. Experimental methods like X-ray crystallography, NMR spectroscopy, and electron microscopy, among others, are used to get the structural data. These databases offer researchers a variety of data, such as atomic coordinates, bonding details, and other chemical characteristics.

• **Protein Data Bank (PDB**) is one of the most well-known structural databases.

• **The Electron Microscopy Data Bank**: is a database that houses the three-dimensional (3D) structures of biological macromolecules that have been discovered by electron microscopy.

• **The Worldwide Protein Data Bank:** a joint project of the US Protein Data Bank (PDB), the European Protein Data Bank (PDBe), and the Japanese Protein Data Bank (PDBj).

• **The Nucleic Acid Database:** is a database that contains the three-dimensional structures of RNA, DNA, and other nucleic acids.

For researchers in the domains of biochemistry, biophysics, and molecular biology, structural databases are a vital resource. These databases are frequently updated with new structural data as it becomes available. Among many other uses, they enable the investigation of molecular structure and function and serve as a basis for medication development.

1. 3. 2. 3. Enzyme databases

Enzyme databases are specialized databases that store information on enzymes, including their biochemical and structural properties, sequences, pathways, and interactions. These databases are important resources for researchers who are studying enzymes and their roles in biological processes. Some of the popular enzyme databases include:

• **BRENDA:** a comprehensive enzyme database that includes information on enzyme nomenclature, reaction mechanisms, substrates, inhibitors, and more.

• **KEGG**: a database that contains information on metabolic pathways and enzymes, as well as genome information for a variety of organisms.

• **MetaCyc** : a database that offers details on the enzymes and metabolic pathways for several organisms, including as bacteria, fungi, plants, and animals.

• **UniProt .**

• **ExPASy**: a proteomics server that includes several databases related to enzymes, including ENZYME, which is a database of enzyme nomenclature and classification.

These databases are valuable resources for researchers in fields such as biochemistry, molecular biology, and biotechnology. They are used for a variety of purposes, including predicting enzyme functions, designing new enzymes, and understanding metabolic pathways. Additionally, they provide a foundation for drug discovery and development, as enzymes are often targeted by pharmaceuticals to treat a variety of diseases.

1. 3. 2. 4. Micro-array databases

Microarray databases are specialized databases that store information related to microarray experiments, which are used to measure the expression of thousands of genes simultaneously. Microarrays are used in a variety of research fields, including genomics, proteomics, and bioinformatics, and have become an essential tool for studying gene expression, genetic variation, and disease mechanisms. Some of the most well-known microarray databases include:

•**Gene Expression Omnibus**: a freely accessible database of high-throughput experimental data from microarrays and other sources, managed by the National Center for Biotechnology Information.

• **ArrayExpress** : a public repository of microarray and other high-throughput experimental data, maintained by the European Bioinformatics Institute .

• **Stanford Microarray Database:** a database that provides access to microarray data and analysis tools, as well as gene expression data from other sources.

• **Human Protein Atlas**: a database that provides information on the expression and localization of proteins in human tissues, including microarray data.

• **Cancer Genome Atlas :** a database that provides access to genomic and microarray data from multiple types of cancer.

These databases allow researchers to share and analyze microarray data, providing a valuable resource for the scientific community. They provide a platform for data integration and analysis, allowing researchers to compare and visualize gene expression patterns across multiple experiments and samples. Microarray databases are also used to identify biomarkers for disease diagnosis and prognosis, as well as to identify potential drug targets and understand disease mechanisms.

1. 3. 2. 5. Clinical databases

A clinical database is a digital repository of patient health information and medical data collected during the course of clinical care. It can be used by healthcare providers, researchers, and other authorized parties to store, manage, and access information related to patients' medical conditions, treatments, outcomes, and other relevant details. Clinical databases can be used to support a wide range of clinical activities, including patient care, clinical research, quality improvement initiatives, and population health management. They can also help healthcare providers make more informed decisions, identify patterns and trends in patient data, and evaluate the effectiveness of different treatments and interventions. There are many examples of clinical databases used in healthcare. Here are a few:

• **National Cancer Database :** It is a national clinical database run by the Commission on Cancer and the American College of Surgeons. It collects information on cancer patients, including demographics, cancer type and stage, treatment information, and survival outcomes.

• **National Cardiovascular Data Registry:** It is a group of clinical databases managed by the American College of Cardiology. It collects data on patients with cardiovascular disease, including information on procedures, medications, and outcomes.

• **Medicare Claims Data:** The government-sponsored health insurance program for Americans 65 and older is called Medicare. It claims data includes information on medical services and procedures provided to Medicare beneficiaries, as well as prescription drug use.

• **Optum Clinformatics Data Mart:** Optum Clinformatics is a database of electronic health records (EHRs) and claims data from millions of patients in the United States. It includes information on medical conditions, procedures, medications, and outcomes.

• **National Inpatient Sample:** It is a database of hospital admissions in the country. It contains data about the patient's demographics, diagnoses, treatments, and results.

They may also be used to support specific medical specialties or subspecialties, such as oncology, cardiology, or pediatrics. Clinical databases are subject to strict privacy and security regulations, as they contain sensitive patient information that must be protected from unauthorized access or disclosure. Compliance with regulations such as the Health Insurance Portability and Accountability Act is essential to ensure the privacy and security of patient data.

1. 3. 2. 6. Pathway databases

Pathway databases are collections of information about biological pathways, including the interactions between genes, proteins, and other molecules that contribute to specific cellular processes. These databases can be used to identify and analyze relationships between biological molecules and pathways, as well as to develop new therapeutic approaches for diseases. Here are a few examples of pathway databases:

• **KEGG Pathway Database:** The Kyoto Encyclopedia of Genes and Genomes (KEGG) is a comprehensive database of pathways, genes, and proteins for various organisms. The KEGG pathway database includes information on metabolic pathways, signaling pathways, and genetic information processing pathways.

• **Reactome:** is a database of human pathways and processes, including information on signaling, metabolism, and gene regulation. It contains information on over 10,000 reactions, 1,000 pathways, and 3,000 proteins.

• **WikiPathways:** WikiPathways is an open-access database of biological pathways maintained by the scientific community. It includes pathways for a wide range of organisms and provides tools for creating and editing pathway diagrams.

• **BioCyc:** BioCyc is a collection of pathway/genome databases for thousands of organisms. It includes information on metabolic pathways, regulatory pathways, and gene regulation.

• **Panther Pathway Database:** The Panther (Protein ANalysis THrough Evolutionary Relationships) database includes pathways for various organisms, including humans, mice, and zebrafish. It includes pathways for biological processes such as cell signaling, metabolism, and immune response.

These databases can be used by researchers to study biological pathways and develop new therapeutic approaches for diseases. They can also be used by clinicians to better understand disease pathways and identify potential drug targets.

1. 3. 2. 7. Chemical databases

Information about chemical substances and their characteristics is compiled in chemical databases. Scientists, researchers, and engineers often utilize them to get data and information about various compounds and their qualities. Chemical databases come in a wide variety of forms, each one intended to fulfill a particular function. Chemical databases, for instance, include:

• **PubChem:** The National Institutes of Health manages this public database of chemical substances. Over 100 million chemical compounds' chemical composition, characteristics, and biological functions are covered in detail.

• **ChemSpider:** The Royal Society of Chemistry maintains this free database of chemical structures. It includes details on more than 80 million chemical structures, such as spectra, characteristics, and references to additional resources.

• **ChEMBL:** The European Bioinformatics Institute maintains this database of bioactive substances. It includes details on the targets, tests, and therapeutic applications of more than 2 million chemicals' biological activity.

• **Reaxys:** Elsevier manages this subscription-based database. More than 240 million chemical compounds are covered, along with details on their reactions, synthesis techniques, and physical and chemical properties.

• **The American Chemical Society Registry** is a database of chemical compounds. It provides details on more than 150 million chemical compounds, including information on their names, structures, and characteristics.

Chemists, pharmacists, and other researchers who study chemical substances need to have access to chemical databases. They offer a wealth of knowledge that may be applied to create new medications, create new materials, and address a variety of chemical issues.

1. 3. 3. Database mining tools (Analysis tools) and techniques

Bioinformatics tools are software applications designed to perform specific tasks, such as sequence alignment, genome assembly, or protein structure prediction. Bioinformatics techniques are broader approaches or methodologies used to solve specific problems in bioinformatics, such as gene expression analysis or phylogenetic analysis. Both tools and techniques are essential in bioinformatics research, as they enable researchers to process and analyze complex biological data and gain insights into the underlying biological mechanisms.

1. 3. 3. 1. Tools

The use of appropriate search engines and analysis tools is necessary for the use of various databases. Database mining is the process of making use of data, and these technologies go by the same name. The following are some analysis tools:

• **BLAST** (Basic Local Alignment Search Tool): This is a widely used tool for sequence alignment and comparison. It can be used to search for similar sequences in nucleotide or protein databases and can help identify homologous genes, infer evolutionary relationships, and predict protein functions.

• **ClustalW**: This tool is used for multiple sequence alignment, which is a fundamental step in comparative genomics and phylogenetic analysis. ClustalW can align nucleotide or protein sequences and generate a phylogenetic tree to visualize evolutionary relationships.

• **HMMER** (Hidden Markov Model based protein sequence analysis): This tool is used for protein sequence analysis, particularly for identifying remote homologs of a protein of interest. It uses hidden Markov models to search for similar protein sequences and identify conserved protein domains.

• **MEME** (Multiple Em for Motif Elicitation): This tool is used to discover motifs, or short conserved sequences, in a set of DNA or protein sequences. MEME uses the Expectation-Maximization algorithm to find conserved motifs and can be useful in identifying regulatory elements in DNA sequences.

• **DAVID** (Database for Annotation, Visualization and Integrated Discovery): This tool is used for gene functional analysis and visualization. It can be used to identify enriched functional annotations, pathways, and gene ontology terms in a set of genes, and visualize the results in a network or heatmap format.

• **ENTREZ:** is a database search and retrieval system created by the US National Center for Biotechnology Information, a division of the National Institutes of Health. It is a comprehensive search engine that allows users to access and search many different biomedical databases, including PubMed, GenBank, and Protein Data Bank.

• **DNAPLOT:** is a tool created by the UK-based European Bioinformatics Institute. Users of the web-based program can see and examine DNA sequence data. DNAPLOT provides various features, including the ability to visualize DNA and protein sequences, compare sequences to identify similarities and differences, and analyze sequence features such as gene locations and motifs.

• **BRITE** (Biomolecular Relations In Information Transmission and Expression) : is a hierarchical classification system designed to provide a comprehensive and consistent annotation of biomolecular functions and relationships across different organisms and bimolecular types. It is based on a tree-like hierarchy of functional categories that are organized based on their structural and functional similarities.

• **Taxonomy Browser**: is a tool created and kept up by the National Center for Biotechnology Information, which is based in the United States. It is an online database that allows users to

explore and navigate the hierarchical classification of living organisms, based on their taxonomic relationships

These are just a few examples of the many database mining tools available in bioinformatics. Depending on the specific research question and type of data being analyzed, other tools may also be useful.

1. 3. 3. 2. Techniques

In bioinformatics, a variety of computational methods are employed. Among them are:

• **Data mining**: Using computational techniques, data mining is the process of identifying patterns and relationships in huge databases. Data mining is a technique used in bioinformatics to evaluate biological data from multiple sources, including genomic, proteomic, and metabolomic data, and to find trends, connections, and new hypothese for additional research.

• **Statistical techniques Biological data**, such as gene expression profiles, protein interactions, and genetic variation, are analyzed and interpreted using statistical techniques. To evaluate the importance of observed differences, infer correlations between variables, and create models to describe biological events, statistical techniques are applied.

• **Network analysis:** Network analysis is the study of the composition and characteristics of large-scale networks, such as gene regulatory, metabolic, and protein-protein interaction networks. Network analysis is used in bioinformatics to pinpoint crucial nodes, clusters, and routes involved in biological processes and to forecast novel relationships and functionalities.

• **Data visualization**: To make it easier to analyze and interpret data and information, data visualization represents it graphically. Data visualization techniques are used in bioinformatics to study and present complicated biological information, such as genetic information, protein structures, and metabolic pathways, in a clear and understandable manner.

• **Sequence Alignment**: Comparing biological sequences like DNA and protein sequences uses the computational method of sequence alignment. Inferring evolutionary links and locating homologous sequences are both done with it.

1. 4. Bioinformatics applications in life sciences and technologies

Bioinformatics has a wide range of applications in life sciences and technologies, some of which are listed below:

1.4.1. Genomics

By offering methods for evaluating massive amounts of genomic data, bioinformatics has significantly advanced the discipline of genomics. One of the key uses of bioinformatics in genomics is the sequencing and analysis of the entire genome. Single nucleotide polymorphisms (SNPs) and structural differences, which may play a role in the development of illnesses, can be found using bioinformatics methods. For individuals with certain disorders, this information can be used to develop precision medicine methods and targeted medicines. Additionally, functional annotation of genes using bioinformatics can shed light on the biological processes and pathways

involved in disease processes. As a result, novel therapeutic targets may be created and disease mechanisms may be better understood [38]. In conclusion, bioinformatics has evolved into a crucial tool in genomics research, allowing for the detection of genetic variants, the creation of tailored therapies, and the identification of novel therapeutic targets.

1.4.2. Proteomics

Proteomics relies heavily on the subject of bioinformatics, which offers computer tools and techniques for analyzing the massive volumes of data produced by this area of study. The application of bioinformatics for protein identification, characterisation, sequencing, and structure prediction has significantly increased in recent years. New algorithms and software tools for peptide identification, protein inference, de novo sequencing, sequence alignment, homology modeling, molecular docking, protein-protein interaction prediction, network analysis, and post-translational modification analysis have been developed to make this possible. These techniques have made it possible for scientists to examine protein structures, functions, and interactions in more detail than ever before, opening up a variety of new biological perspectives, such as improved illness detection and medication development [39]. Bioinformatics has been used, for instance, to pinpoint protein biomarkers for numerous illnesses, including cancer, and to forecast how medications would affect protein structures. Overall, bioinformatics provides the computational and is a crucial enabling tool for proteomics.

1.4.3. Drug discovery

With the ability to quickly and effectively analyze vast amounts of biological data, bioinformatics has emerged as a key tool in the drug discovery process. Bioinformatics can uncover prospective pharmacological targets, forecast the effectiveness and toxicity of treatment candidates, and offer insights into the molecular mechanisms underlying disease. Finding new therapeutic targets is one way that bioinformatics is used in the drug discovery process. Bioinformatics can discover important biological processes involved in disease progression and find new targets for drug development by studying genomic, proteomic, and metabolomic data. By foreseeing how potential drugs would interact with certain molecular targets and evaluating their potential adverse effects, bioinformatics can also be utilized to enhance the efficacy and safety of medication candidates. Overall, incorporating bioinformatics into the process of finding novel medications has the potential to speed up drug development and enhance patient outcomes.

1.4.4. Personalized medicine

Bioinformatics has been instrumental in the development of personalized medicine by allowing for the analysis of large-scale biological data to develop individualized treatment plans. By analyzing an individual's genomic and clinical data, bioinformatics can identify genetic variations that can influence a person's response to certain treatments. One example of the application of bioinformatics in personalized medicine is the use of pharmacogenomics, which involves studying the relationship between an individual's genes and their response to drugs. Bioinformatics tools can analyze a patient's genomic data to identify genetic variants that may affect the metabolism or

efficacy of certain drugs, allowing for the development of tailored treatment plans that maximize effectiveness and minimize adverse effects [40]. In addition, bioinformatics can be used to analyze clinical data, such as medical imaging and electronic health records, to provide insights into disease progression and identify the best treatment options for a particular patient. Overall, bioinformatics has become a key tool in personalized medicine, enabling healthcare providers to create custom treatment programs that are suited to each patient's particular genetic profile and physical features.

1.4.5. Agricultural biotechnology

Bioinformatics plays an important role in agricultural biotechnology by providing tools and techniques for analyzing and interpreting vast amounts of genomic and proteomic data, thereby accelerating the development of new crop varieties and improving agricultural productivity. One of the key applications of bioinformatics in agricultural biotechnology is in crop improvement through genomics-assisted breeding. By analyzing the DNA sequences of different crop varieties, researchers can identify genes associated with desirable traits such as disease resistance, drought tolerance, and yield potential, and use this information to develop new varieties with improved traits [41].

1.4.6. Forensic science

For the study of DNA evidence, bioinformatics has grown in significance as a technique in forensic science. DNA profiling, which analyzes short tandem repeat markers to produce a distinctive genetic profile for a person, is one of the major uses of bioinformatics in forensic research. Comparing DNA profiles from various sources, including as crime scenes, suspects, and victims, using bioinformatics methods allows for the identification of matches and the exclusion of non-matches. Bioinformatics is also used to analyze complicated DNA mixes, which can be challenging to interpret using conventional techniques. Bioinformatics can aid to identify the contributors to the mixture and provide evidence for forensic investigations by utilizing statistical models and algorithms to evaluate DNA mixes [42].

1.4.7. Environmental science

Environmental science utilizes bioinformatics in a variety of ways, using it to evaluate vast amounts of data produced by environmental sample data. Analyzing metagenomic data is one of the major uses of bioinformatics in environmental science. The study of genetic material that has been extracted directly from environmental materials, such as soil, water, and air, is known as metagenomics [43]. The analysis of metagenomic data using bioinformatics techniques enables scientists to recognize and categorize the microorganisms present in a given environmental sample and to look into their functional roles in that environment. Environmental science also uses bioinformatics for the analysis of data related to air and water quality. For the study and interpretation of intricate environmental data, bioinformatics is a vital tool.

1.5. Challenges and limitations

While bioinformatics has many advantages, it also faces several challenges and limitations. One of the biggest challenges of bioinformatics is the quality of biological data. Biological data can be highly variable, with errors and noise introduced at every step of the experimental and data analysis processes. For example, there may be errors in DNA sequencing, errors in gene annotations, or biases in sample collection. These issues can introduce noise and errors into the data, making it difficult to draw reliable conclusions from bioinformatic analyses. Another challenge in bioinformatics is the complexity of biological systems. Biological systems are inherently complex, with many interacting components and feedback loops. This complexity can make it difficult to develop accurate and predictive models of biological systems. In addition, there may be many unknowns about how different components of biological systems interact, which can make it difficult to accurately model these interactions. Bioinformatic analyses are often based on predictions and correlations, and may not be backed up by extensive experimental validation. This can lead to false positives and over-interpretation of results. Biological data is often collected from many different sources, and integrating this data to create a comprehensive picture of biological systems can be challenging. Integration can also be hindered by differences in data formats, standards, and annotations across different databases and data sources. In addition, computing resources can be a limiting factor for bioinformatics research. Bioinformatic analyses often require significant computing resources, both in terms of processing power and data storage. This can be a challenge for researchers without access to high-performance computing resources. Finally, as bioinformatics becomes increasingly used in personalized medicine and genetic testing, issues around ethics and privacy become more important. It is important for bioinformaticians to be aware of these issues and to follow appropriate ethical guidelines. This includes ensuring that patient data is handled securely and confidentially and that bioinformatic analyses are used in a responsible and ethical manner [44, 45, 46].

While bioinformatics has many advantages and has made significant contributions to our understanding of biological systems, it also faces many challenges and limitations. Addressing these challenges will be important in order to ensure that bioinformatics continues to be a powerful tool for understanding biological systems and developing new treatments and therapies.

1. 6. Conclusion

The field of bioinformatics has revolutionized the way we study biological systems and has become an integral part of modern biological research. The increasing amount of biological data generated from various sources requires sophisticated tools and algorithms to extract meaningful insights. Bioinformatics has enabled the efficient organization, analysis, and interpretation of biological data, providing researchers with new opportunities for discovery and advancing our understanding of life processes. As the field continues to develop and evolve, the future of bioinformatics looks promising, with the potential to revolutionize medicine, biotechnology, and other fields that rely on a deeper understanding of biological systems. A group of bioinformaticians announced their findings in June 2021 after using artificial intelligence to anticipate the architecture of more than 350,000 proteins produced by the human body. One of the most significant in recent years, this discovery has made it possible to design new treatments to cure diseases and take a giant step ahead in our understanding of the human body.

Chapter 2

Data Mining and Classification Models

Chapter 2: Data Mining and Classification Models

Data mining and classification techniques are of paramount importance in various domains, as they enable the extraction of valuable insights and patterns from large and complex datasets. These techniques facilitate the organization, categorization, and interpretation of data, empowering researchers to make informed decisions and uncover hidden relationships that can drive advancements in scientific research and decision-making processes.

In this chapter, we delve into the fundamental principles and methodologies encompassing data mining and classification models. It is organized as follows: In the second section, we talk about data mining concept followed by data mining techniques including supervised and unsupervised learning in the third section. Section four described techniques used in data preprocessing for mining. In the fifth section, we present the most important classification models.

2.1. Introduction

Data mining is a rapidly growing field that is focused on discovering hidden patterns and insights in large and complex datasets. This interdisciplinary field brings together techniques and methods from various domains, including statistics, machine learning, computer science, and more. The primary objective of data mining is to extract knowledge from data that can be utilized to enhance decision-making and solve real-world problems. In recent years, the growth of data mining has been driven by the explosion of data in various fields such as social media, scientific research, and the internet. Data mining techniques enable the extraction of valuable insights from large datasets that would otherwise be difficult or impossible to discover.

Classification is a critical step in data mining, involving the categorization or labeling of fresh observations based on their characteristics or properties. Various industries, including marketing, healthcare, and finance, utilize classification models generated from previous data to make precise predictions about unobserved data. The creation of effective classification models involves stages such as data preprocessing, feature selection, and model selection. Data must be transformed, cleaned, and scaled to prepare it for modeling. Feature selection plays a crucial role in enhancing the effectiveness and accuracy of classification models by identifying insightful and relevant features from the dataset. Model selection entails choosing the appropriate categorization model that best addresses the data and the specific problem at hand. Given the exponential growth of data in domains like social media, the internet, and scientific research, data mining and classification have gained increasing importance.

In this chapter, we will explore the fundamentals of data mining and classification models, including various techniques and algorithms used for data preprocessing. We will also discuss the advantages and limitations of different classification models.

2.2. Concept of data mining

Data mining has been a concept that predates the digital era, encompassing the utilization of data for uncovering knowledge. For centuries, there have been manual formulas for statistical modeling and regression analysis that laid the foundation for this idea. In the 1930s, Alan Turing's introduction of the universal computing machine introduced the era of electromechanical computers. This development sparked the proliferation of digital information, which has continued to grow exponentially up until the present day. Significant

progress has been made since then, with data now permeating every aspect of business and everyday life [47].

Since the 1990s, the definition of data mining has evolved. One of the seminal definitions was presented by Fayyad, Piatetsky-Shapiro, and Smyth in their book [48]. They defined data mining as the process of examining huge datasets to find relevant connections, patterns, and trends by poring over vast volumes of data that are kept in repositories. This involves sifting through great volumes of information kept in databases, data warehouses, and various other information sources, where the data may be incomplete, noisy, or inconsistent. This definition highlights the importance of discovering "interesting knowledge" and the challenges that arise when dealing with large, complex datasets. It also recognizes that data may be incomplete, noisy, or inconsistent, and that there may be implicit relationships and unknown patterns to be discovered. In order to extract this knowledge from the data, several statistical and mathematical methods as well as pattern recognition technologies are used.

Briefly, we can define data mining as the process that uses statistical and computational methods to explore large datasets and discover patterns and insights. Its main goal is to extract valuable information from data and present it in an understandable format for further analysis. By uncovering hidden patterns and trends in data, data mining can help organizations make better decisions and drive business growth. For example, data mining can be used to identify customer buying patterns to inform product development and marketing strategies, as well as optimize business operations by detecting inefficiencies and areas for improvement.

2.3. Data mining techniques

Data mining techniques are a group of statistical models and algorithms used to glean new knowledge from huge and complicated databases, which can subsequently be leveraged for decision-making. Based on the learning approach, data mining techniques can be divided into two primary categories: supervised learning and unsupervised learning.

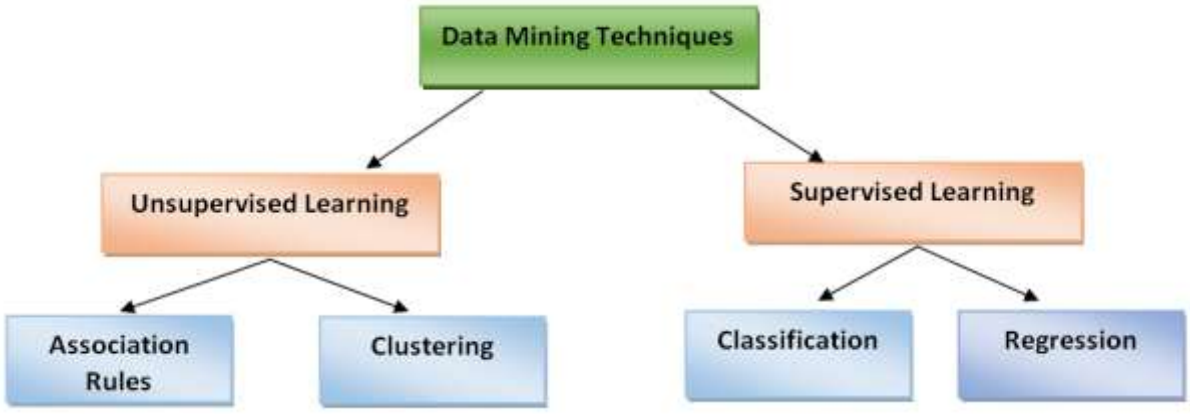


Figure 2.1. Data mining techniques.

2.3.1. Supervised learning:

In the process of supervised learning, a model is trained using labeled data and a known target variable. The end goal is to build a predictive model that, using the patterns discovered from

the labeled training data, can accurately predict the target variable for new or unobserved data [49]. The dataset must be divided into the training set and the testing set, where the latter evaluates the model's performance, and the former is used to train the model. The testing set often consists of the residual data after the training set, which typically comprises a sizeable amount of the data (about 70–80%). To acquire accurate results, it is crucial to make sure that the dataset is diverse and contains examples that are typical of the population [50]. The model is trained by feeding labeled data into the algorithm and iteratively modifying the model's parameters to close the gap between expected and actual output [51]. Using the testing set, the model is tested after it has been trained, and its accuracy is evaluated by contrasting the anticipated output with the actual output for each example in the testing set. The evaluation metrics used for measuring model performance depend on the problem being solved. There are two main categories of supervised learning which are:

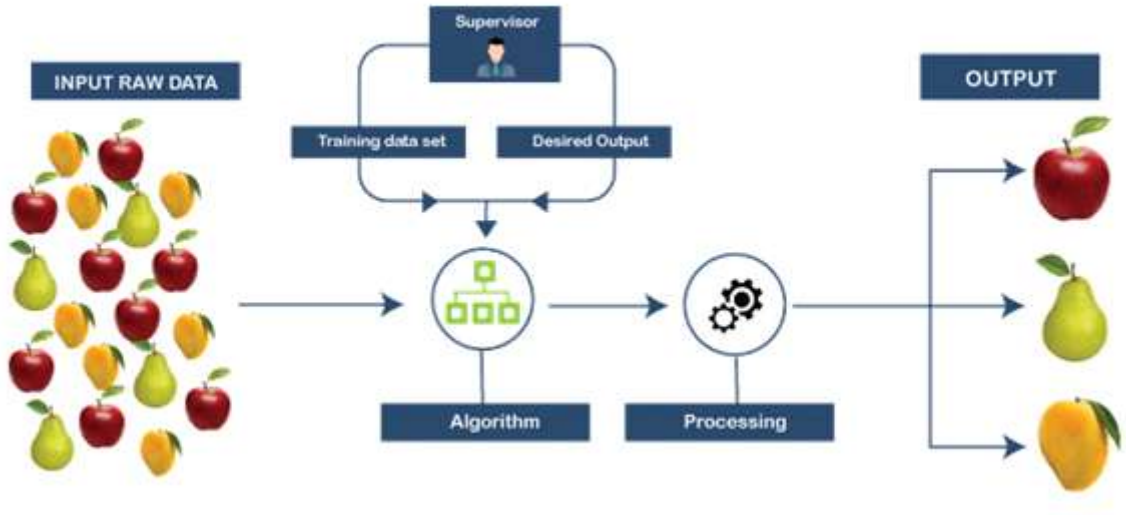


Figure 2.2. Supervised learning [52].

2.3.1.1. Classification

Classification is a crucial task that involves grouping input data into predetermined categories or labels based on their specific features or attributes. This process typically requires the training of a machine learning (ML) model on a labeled dataset, where each data point is associated with a known class or label. The model learns to detect patterns or relationships within the features of the labeled data, enabling it to make predictions on the labels of new, unlabeled data points [53, 54]. Various examples of classification problems exist, such as identifying spam emails based on their content and characteristics, predicting loan defaults based on credit history, and recognizing specific objects or animals within an image based on their visual features. There are different types of classification techniques that can be employed which are:

A. Binary classification

As seen in Figure 2.3(A), binary classification is a supervised ML technique used to predict a binary output variable based on a set of input features. The objective of binary classification is to predict a binary output variable that has only two possible values, typically expressed as 0 and 1, negative and positive, or false and true [55]. Binary classification is used in many different sectors, including predicting if an email is spam or not, detecting whether a transaction is fraudulent or not, assessing whether a person has a specific disease or not, and more [56].

B. Multi-class classification

Multi-class classification is a type of classification problem where each data point is assigned to one of several possible classes or labels [57], (Figure 2.3 (B)), (Figure 2.4 (A)). Multi-class classification problems are common in various applications such as object recognition, speech recognition, and text classification. For example, in text classification, a document can be assigned to one of several possible topics or categories, such as politics, sports, or entertainment. In speech recognition, an utterance can be classified into one of several possible words or phrases. In object recognition, an image can be classified into one of several possible objects, such as a car, a tree, or a person.

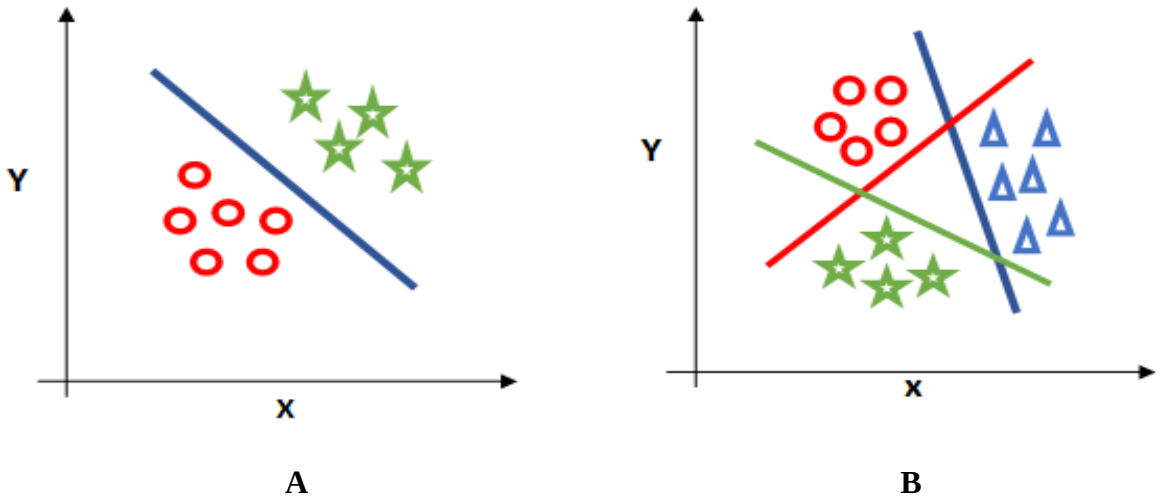


Figure 2.3. Binary and Multi-class classification.

C. Multi-label classification

The term "multi-label classification" describes a classification issue in which each data point has the potential to be assigned more than one label or class, as illustrated in Figure 2.4 (B). In simpler terms, the classifier's output is not limited to a single label; instead, it can assign a set of labels to the input. The classifier is trained to predict whether each combination of labels is present or absent [58, 59].

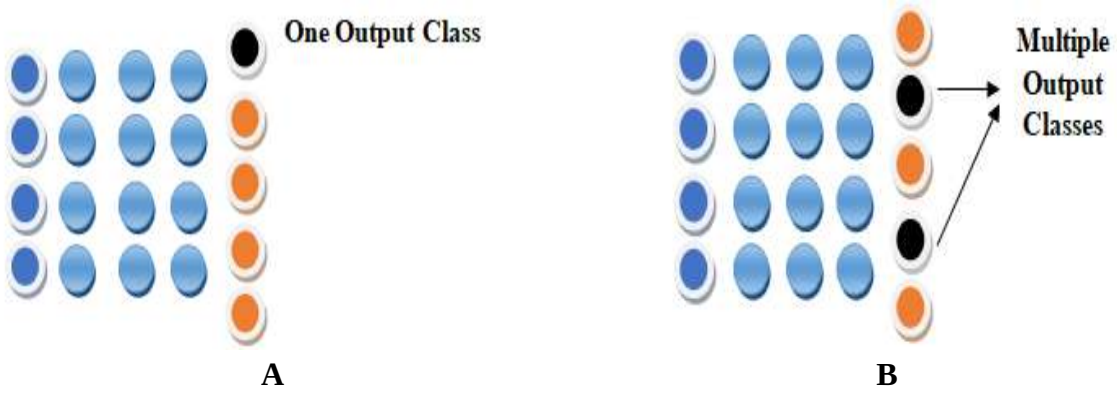


Figure 2.4. Multi-class and Multi-label classification [60].

For example, classifying an image of a person by their age, gender, and ethnicity. This approach essentially converts the multi-label classification task into a multi-class classification problem, but with a much larger number of classes. Multi-label classification can be a challenging problem because of the large number of possible label combinations, which can lead to sparsity in the data and difficulty in selecting appropriate evaluation metrics. However, it is a useful technique for many real-world applications where data points can belong to multiple categories or have multiple attributes.

D. Hierarchical classification

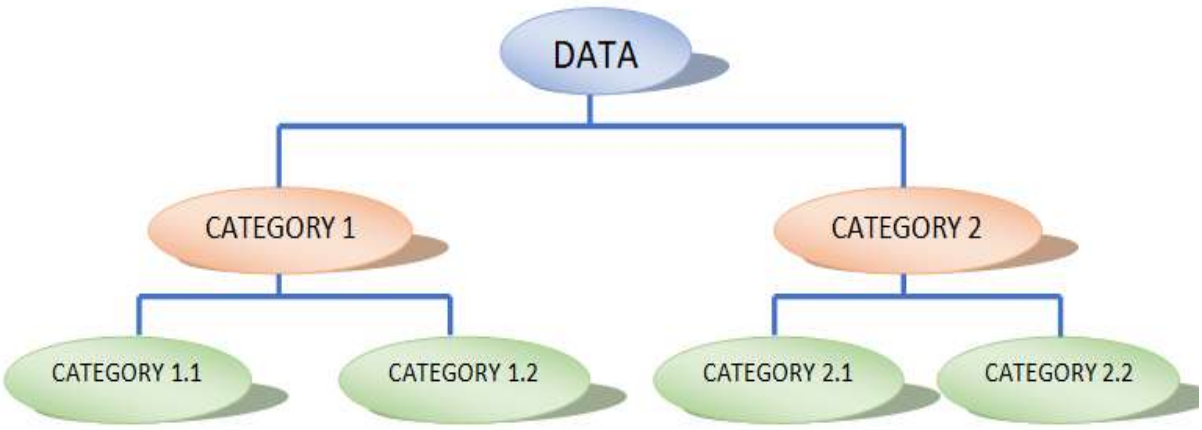


Figure 2.5. Hierarchical classification.

Hierarchical classification is a type of classification that is closely related to multi-label classification, but with an added structure. In hierarchical classification, the labels or classes are arranged in a hierarchical or tree-like fashion, where higher-level categories are more general and broad, whereas lower-level categories are more detailed [61],(as shown in Figure 2.5). For example, classifying animals into broad categories like mammals, birds, reptiles, etc., and then further classifying them based on specific characteristics. In a hierarchical classification problem, the goal is to assign each data point to the most specific class or label

in the hierarchy. This can be done by either top-down or bottom-up approaches. In the top-down approach, the algorithm starts with the most general category and recursively splits it into more specific categories until it reaches the most specific category that fits the data point. In the bottom-up approach, the algorithm starts with the most specific category and aggregates it into broader categories until it reaches the most general category that fits the data point.

E. Imbalanced classification

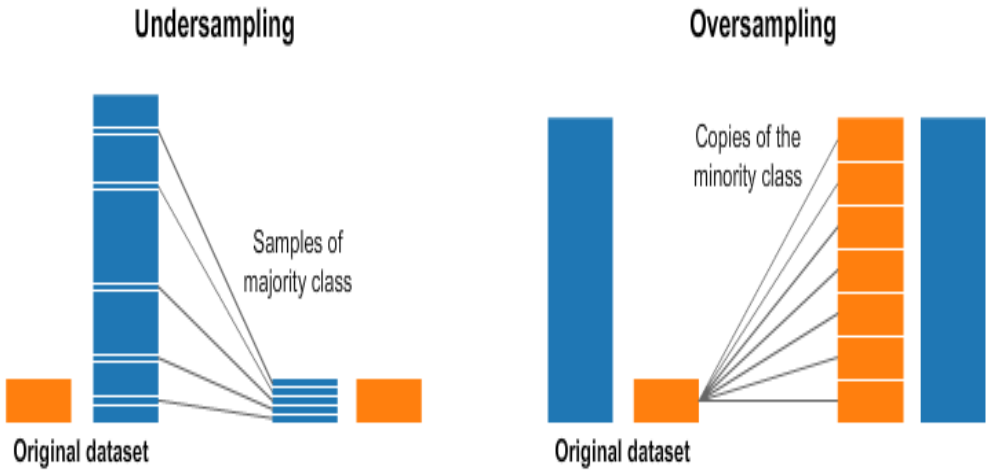


Figure 2.6. Imbalanced classification [62].

A classification issue known as imbalanced classification occurs when there is considerable variation in the number of instances among different classes or labels (Figure 2.6). Detecting uncommon medical diseases, as an illustration, using patient data. One class (referred to as the minority class) may have much fewer examples than the other classes (referred to as the majority class(es)) in an unbalanced classification problem. Because most classification algorithms have a tendency to favor the majority class, imbalanced classification can be a difficult problem in ML and lead to subpar performance when predicting the minority class. A classification model that consistently predicts the majority class, for instance, would achieve 99% accuracy but would entirely fail to identify the minority class if a dataset had 99% of its instances belonging to one class and only 1% to another. There are numerous strategies that may be utilized to address class imbalance in a dataset in order to handle the problem of imbalanced classification. By either oversampling the minority class or undersampling the majority class, resampling seeks to balance the number of samples in each class. By allocating various misclassification costs to various classes, cost-sensitive learning enables the model to take the data imbalance into account. In order to enhance the number of training examples accessible for the minority class, synthetic data generation also involves creating new examples for that class. To enhance the performance of classification models on unbalanced datasets, these methods can be applied separately or in combination. By

preventing the minority class from being neglected or ignored during the classification process, these strategies can aid in improving the performance of classification models on imbalanced datasets [63, 64].

2.3.1.2. Regression

A fundamental job in supervised ML called regression is making predictions about a continuous output variable using a set of input features. This method is frequently used to examine the relationship between independent and dependent variables in statistical modeling and data mining. In a variety of disciplines, including finance, economics, and marketing, regression models are employed. Regression predicts a value within a continuous range, in contrast to binary classification, making it ideal for forecasting numerical outcomes like stock prices or home prices. The following list includes some of the most popular regression types:

A. Linear regression

It is a statistical technique used to establish a connection between one or more independent variables (often referred to as "X") and a continuous dependent variable (usually represented as "Y"), the objective is to determine the line that best captures the patterns exhibited by the observed data points [65], (as illustrated in Figure 2.7 (A)). Then, this line can be used to forecast new values of the independent variable(s). To anticipate and model numerous occurrences, the linear regression method is frequently utilized in a variety of fields, including engineering, economics, social sciences, and finance. But because it presumes a linear relationship between the variables, it might not be suitable for non-linear ones.

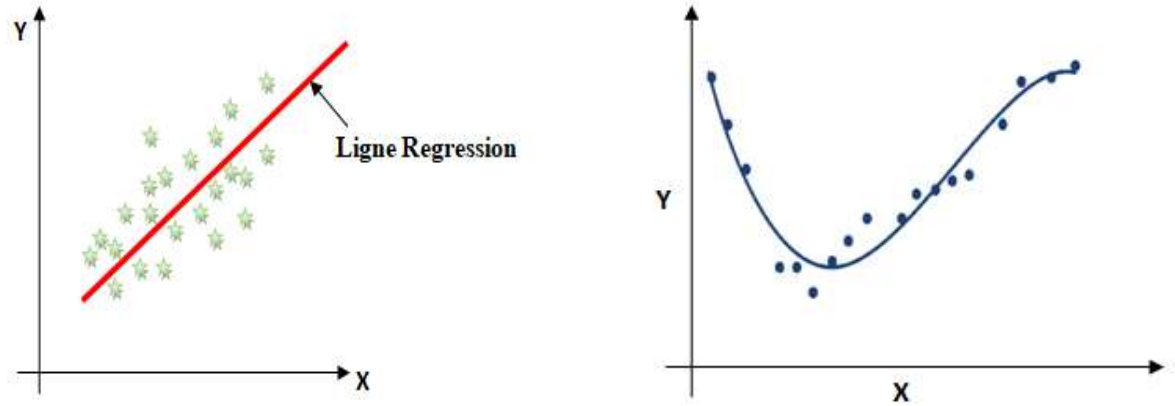


Figure 2.7. Linear and Polynomial regression.

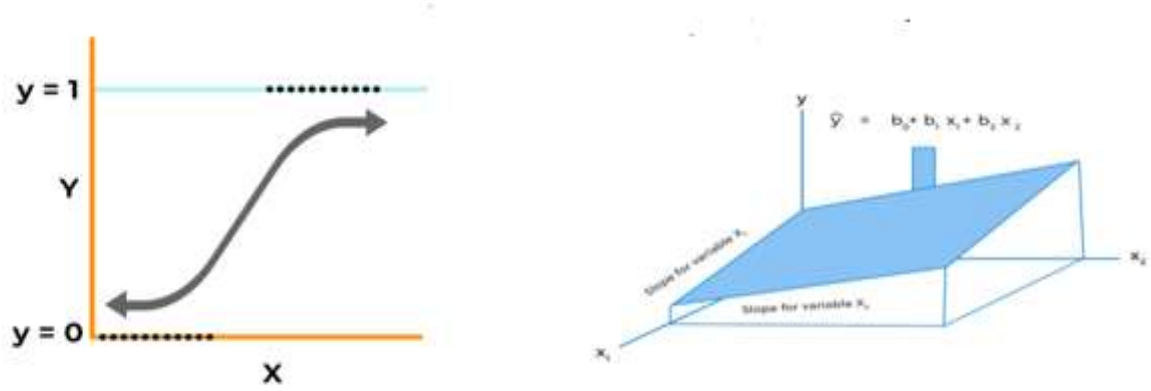
B. Polynomial regression

In polynomial regression, the relationship between the independent and dependent variables is modeled using a polynomial function of degree n (Figure 2.7(B)) .Polynomial regression can capture more complex nonlinear interactions than linear regression, which presumes a linear

relationship between the variables. Using optimization methods like least squares, polynomial regression aims to fit a polynomial equation to the data. The complexity of the relationship and the balance between precision and overfitting are taken into consideration when determining the polynomial's degree [66, 67]. Physicists, biologists, and economists frequently employ polynomial regression to simulate intricate interactions between variables.

C. Logistic regression

It is common practice in disciplines including health, biology, social sciences, and marketing to apply a statistical model known as logistic regression to estimate the likelihood of a binary outcome based on one or more independent factors (Figure 2.8 (A)). It is a kind of generalized linear model that applies a logistic function to translate every input value into a probability between 0 and 1. Contrary to linear regression, which forecasts continuous numerical values, logistic regression is created especially for issues involving binary categorization [68].



A **B**

Figure 2.8. Logistic and Multiple regressions [69-70].

D. Multiple regressions

Multiple regressions are a type of statistical model that can be used to represent the relationship between a dependent variable and two or more independent variables. The idea of simple linear regression, which examines the link between a single independent variable and a single dependent variable, is expanded by this method [71], (Figure 2.8(B)). We can shed light on the combined impact of several factors on the desired outcome by using multiple regressions. It is commonly used to comprehend and forecast complicated interactions among variables in a variety of domains, including the social sciences, economics, finance, and business.

E. Ridge regression

In cases of multicollinearity (high correlation) among the independent variables, ridge regression, a particular kind of linear regression, is employed (Figure 2.9 (A)). Finding the optimum equation that minimizes the sum of the squared errors while also penalizing high coefficient values is the aim of ridge regression. The least squares equation is amended by a

penalty term also referred to as the regularization parameter or the ridge parameter. The regression coefficients are reduced toward zero by the penalty term, which is a function of the squared magnitudes of the coefficients [72].

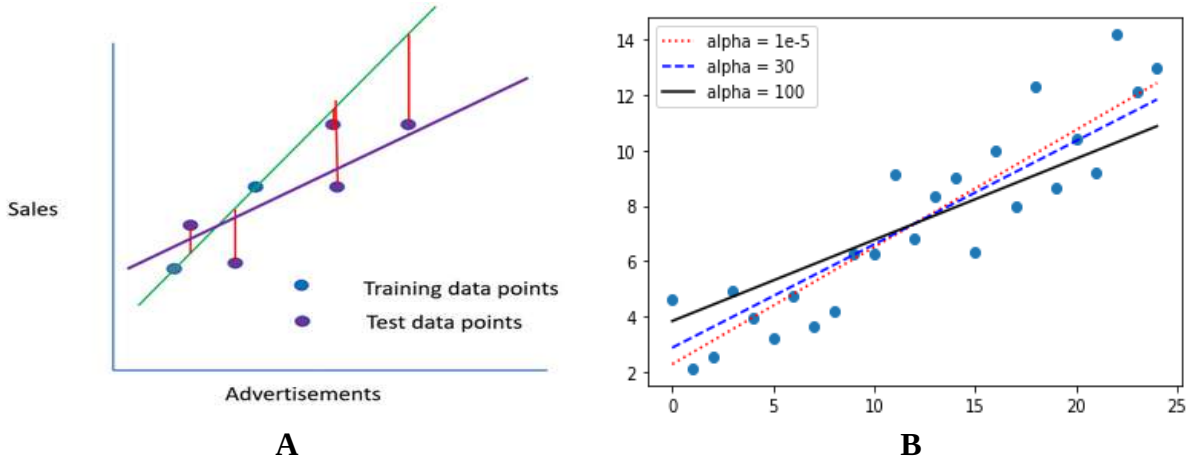


Figure 2.9. Lasso and Ridge regression [73-74].

F. Lasso regression

This type of regression is also used when there is multicollinearity among the independent variables (Figure 2.9 (B)). The goal is to find the best equation that minimizes the sum of the squared errors while also reducing the number of independent variables in the model [72].

2.3.2. Unsupervised learning

The purpose of unsupervised learning is to uncover patterns or structures in the data without being explicitly told what to look for (Figure 2.10). This is done by training a model on unlabeled data. Unsupervised learning does not have a specified goal variable that the model is attempting to predict, in contrast to supervised learning. Finding connections and patterns within the data itself is what this implies. When the data is unstructured or when the patterns or links within the data are unclear, unsupervised learning might be helpful. It can be used for things like association rules, anomaly detection, and clustering. As there is no specified objective variable to predict in unsupervised learning, the dataset is often not split into a training set and a testing set. Instead, the model is tested based on how effectively it recognizes patterns or structure in the data after being trained on the complete dataset. There are two main categories of unsupervised learning which are:

2.3.2.1. Association rules

In order to find intriguing connections between variables in massive datasets, association rules are a form of rule-based ML technique (Figure 2.11). They are commonly used in data mining, market basket analysis, and customer segmentation. Association rules are built on the concept of frequent item sets, which are sets of items that appear together in transactions more

frequently than a predefined threshold (usually a minimum support level). Once frequent itemsets have been identified, association rules can be generated by examining the relationships between items in the itemsets. The confidence and support in an association rule are measures of its strength. While confidence is the proportion of transactions that contain the subsequent item when the antecedent item is present, support is the proportion of transactions that have both items in the rule [76, 77]. Association rules can be useful for a variety of applications, such as product recommendations, cross-selling, and customer behavior analysis. By understanding the relationships between items in a dataset, businesses can make informed decisions about pricing, promotions, and inventory management. The most well-known algorithms for discovering association rules are:

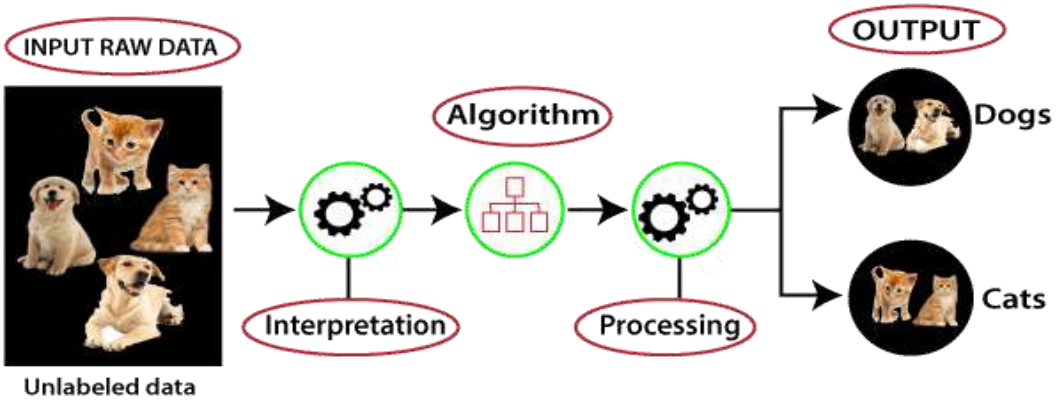


Figure 2.10. Unsupervised learning [75].

• **Apriori Algorithm**: The most used algorithm for mining association rules is this one. Up until there are no more frequent itemsets left to find, it generates frequent itemsets of increasing size. The algorithm employs a bottom-up methodology, where it starts with individual items, then generates all possible combinations of items with a minimum support threshold [76].

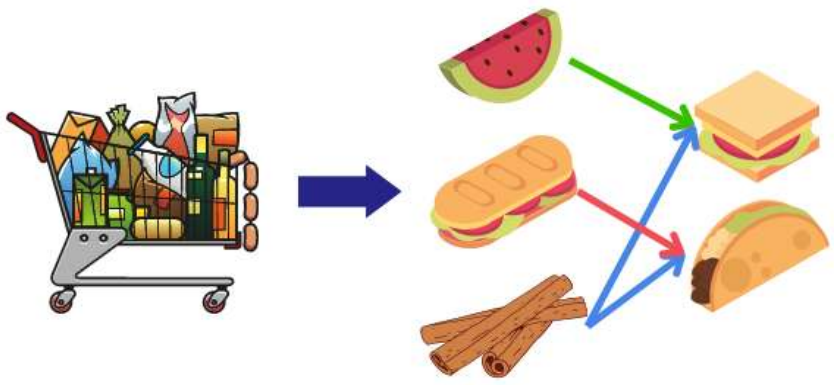


Figure 2.11. Association rules learning [79].

• **FP-Growth Algorithm**: This is another popular algorithm for association rule mining. Unlike Apriori, which generates candidate itemsets by joining pairs of frequent itemsets, the FP-growth technique creates an FP-tree, a tree-like data structure, to effectively condense the transactions and find common itemsets [76].

• **ECLAT Algorithm**: This technique uses a depth-first search approach to locate frequently used itemsets. Recursively searching through the dataset's subsets of items, it eliminates subsets that fall below the required level of support [78].

2.3.2.2. Clustering

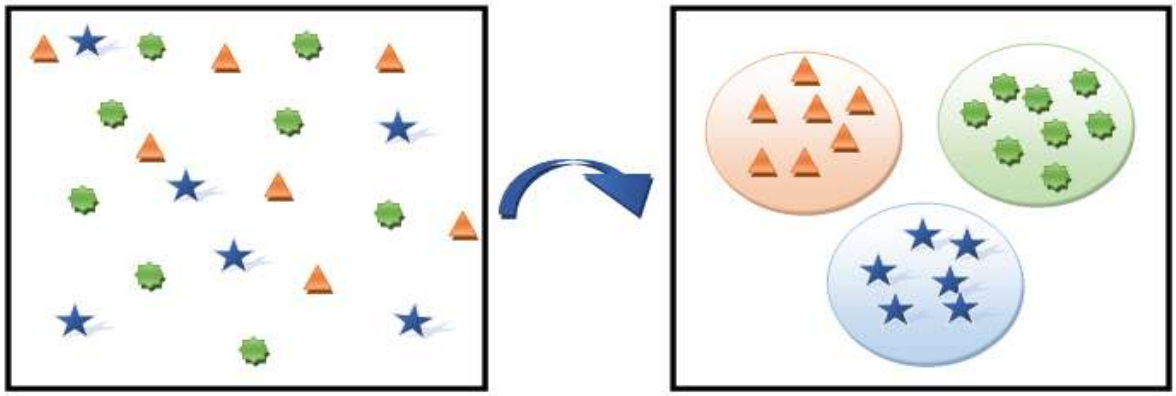


Figure 2.12. Clustering technique.

In ML and data analysis, the clustering technique is used to group together data points with similar characteristics. (Figure 2.12). It is an unsupervised learning technique where the algorithm identifies patterns in the data without being explicitly trained on labeled examples. The goal of this technique is to identify groups of data points that share common features or properties. The algorithm assigns each data point to a cluster based on its similarity to other points in that cluster while maximizing the dissimilarity between clusters [80]. Market segmentation, image processing, and natural language processing are just a few of the uses for clustering. In huge datasets, the clustering technique can be utilized to find patterns and relationships that would be difficult or impossible to find by manual examination. Additionally, it can be applied as a preliminary step in other ML techniques, such as classification and anomaly detection, to help improve their performance. Clustering algorithms come in a variety of forms, each with unique advantages and disadvantages. The following are a few of the most typical clustering methods:

• **K-means clustering:** One of the most used clustering algorithms is this one. The data is divided into k clusters, where k is a user-specified parameter, and this is how it functions: The method repeatedly assigns each data point to the centroid of the closest cluster. The centroids are then updated based on the updated cluster assignments, and the process continues iteratively until a predefined stopping criterion is met. Although K-means is quick and effective, the initial cluster centroids selected may have an impact on its performance [81].

• **Hierarchical clustering:** This algorithm produces nested clusters that resemble a tree, where each cluster is a subset of a bigger cluster. Hierarchical clustering can be divided into two basic categories: agglomerative and divisive. Each data point in agglomerative clustering is initially clustered separately, and the closest cluster pairings are then repeatedly combined to form a single cluster for all the data points. In the second category, the algorithm recursively splits clusters into smaller subclusters until the desired level of granularity is achieved [82].

• **Density-based clustering:** Based on regions of the data with a high density, this algorithm finds clusters, where points are close together and separated by areas of lower density. Density-based clustering algorithms are useful for data with irregular shapes and sizes, and can handle noise and outliers well [83].

• **Spectral clustering:** This method transforms the data into a lower-dimensional space using the similarity matrix's eigenvectors, and then employs a clustering algorithm in the newly created space. It is particularly effective for non-linearly separable data [84].

• **Subspace clustering:** This type of clustering is used when the data has multiple subspaces or dimensions. Subspace clustering algorithms identify clusters in subsets of the data dimensions, rather than in the entire dataset at once. This approach can be more effective than traditional clustering methods when the data has multiple subspaces [85].

• **Fuzzy clustering:** permits each data point to have a different degree of participation in different clusters. When there is data uncertainty or when a data point may belong to more than one cluster, this method can be helpful when there is overlap or mixed membership across clusters, or when data points may have ambiguous or uncertain cluster assignments [86].

2.4. Data preprocessing for mining

Data preprocessing aids in converting raw data into a format that can be analyzed, making it a crucial step in the data mining process. Data preprocessing's major objective is to make sure the data is accurate, consistent, and prepared for usage in the mining process. Here are some key steps involved in data preprocessing for mining:

2.4.1. Data cleaning

In the critical stage of data mining known as data cleaning, defects, inconsistencies, and inaccuracies in data are found and then corrected or eliminated. The accuracy, completeness, and dependability of the data are improved by data cleaning, which also raises the caliber and dependability of the conclusions and models that are drawn from the data. Here are some common techniques used in data cleaning:

• **Handling missing values**: Missing values can occur when data is not collected or recorded for some observations. Several techniques for handling missing values are used such as

imputation (filling in missing values based on the values of other observations), deletion (the elimination of observations with blank values), and flagging (identifying missing values as a separate category) [87].

• **Handling duplicates:** Duplicates can occur when the same data is recorded multiple times. Duplicates can be identified and removed using techniques such as sorting, deduplication, and record linkage [88].

• **Handling outliers:** Data points known as outliers differ markedly from the rest of the dataset or expected behavior. Outliers can be identified and handled using techniques such as clustering, data visualization, and statistical methods [89].

• **Handling inconsistencies and errors:** Inconsistencies and errors in the data can arise due to data entry errors, measurement errors, or inconsistencies in the data collection process. These can be identified and corrected using techniques such as data profiling, data cleaning rules, and manual inspection [87].

The accuracy and efficacy of the insights and models that are obtained from the data can be improved by using these methods and others to guarantee that the data is clean and ready for analysis.

2.4.2. Data integration

Data integration is the process of combining data from many sources to give the data a unified perspective. This is often necessary when dealing with large and complex datasets that are spread across multiple systems and formats. There are several techniques that can be used for data integration:

• **ETL (Extract, Transform, Load):** This method is frequently used to combine data from various sources into a single system. Data must be extracted from several sources, formatted to fit a similar pattern, and loaded into the intended system. ETL tools such as Talend, Informatica, and Microsoft SSIS can automate this process [90].

• **Data virtualization:** This technique involves creating a virtual layer over the data sources, allowing users to access and query the data as if it were in a single system [91]. Data virtualization tools such as Denodo and SAP HANA can provide a unified view of the data without physically moving or copying it.

• **Data warehousing:** This involves creating a centralized repository for data from multiple sources, which can then be accessed and analyzed using several tools [92]. Data warehouses such as Oracle, Teradata, and Microsoft SQL Server can provide a consolidated view of the data.

• **API integration:** This involves using application programming interfaces (APIs) to integrate data from different sources. APIs allow different systems to exchange data in real time, making it easier to keep data up-to-date and synchronized [93].

• **Manual integration:** This involves manually combining data from different sources, such as copying and pasting data between spreadsheets or databases. This technique can be time-consuming and error-prone but may be necessary in some cases.

2.4.3. Data transformation

The process of transforming raw data into a more useful format for analysis is known as data transformation [46]. As it can significantly affect the precision and utility of the final analysis, it is an important step in the data preparation process. Data transformation is frequently required because ML algorithms frequently need input data in a particular format or range. Additionally, new characteristics can be extracted from the existing data to enhance the functionality of ML models. Therefore, for efficient data preparation and analysis, understanding data transformation techniques is crucial. There are several techniques for data transformation, including:

• **Feature scaling:** This technique scales the values of a variable to a specific range, such as between -1 and 1 or between 0 and 10. Feature scaling is often used to avoid the impact of outliers on the analysis. Two main techniques of feature scaling:

1. **Normalization:** With this method, a variable's values are scaled between 0 and 1, making the variable's values comparable to each other. Normalization is often used when the range of values in a variable varies widely [94].
2. **Standardization:** This technique transforms the values of a variable to have a mean of 0 and a standard deviation of 1, making it easier to compare the variable to other variables [94].

• **One-hot encoding:** This technique is used to transform categorical variables into numerical variables [95]. It creates new binary variables for each category, a value of 1 indicating that the observation corresponds to that category, and a value of 0 indicating otherwise.

• **Discretization**: is the procedure for changing continuous data into a discrete form. In other words, it involves dividing a continuous variable into a finite number of categories or intervals [96]. This is often done in order to simplify data analysis or modeling, and to make the data more manageable or easier to understand. Discretization is commonly used in various fields such as statistics, signal processing, and ML. For example, in ML, continuous variables may be discretized in order to create a categorical feature that can be used in a model. In signal processing, continuous signals may be discretized into samples for further processing.

2.4.4. Data reduction

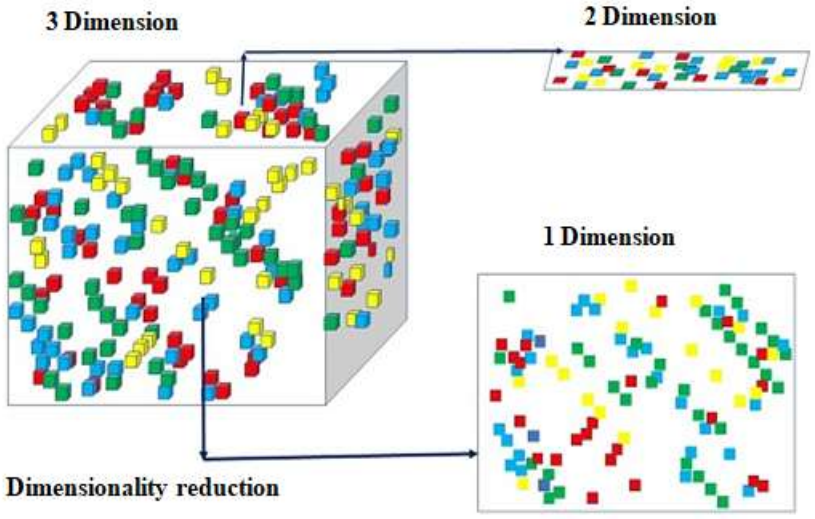


Figure 2.13. Data reduction [96].

Data reduction refers to the process of reducing the amount of data in a dataset without losing important information. This is often necessary when dealing with large datasets that are too computationally expensive to analyze in their entirety. There are several techniques for data reduction including :

• **Sampling:** This technique includes choosing a portion of the primary dataset to be analyzed. There are different sampling techniques, such as cluster, stratified, and random sampling.

• **Dimensionality reduction:** This technique entails keeping the majority of the original data while lowering the number of variables in the dataset (Figure 2.13). There are two main techniques for dimensionality reduction:

1. **Feature extraction:** This method enhances the performance of models by generating additional features from the existing data. It may entail methods like principal component analysis, which lowers the number of data dimensions while maintaining the majority of the original data [97].
2. **Feature selection:** Using this method, a subset of the original dataset's variables is chosen based on how crucial they are to the investigation .Filter methods, wrapper methods, and embedded methods are approaches for feature selection [97].

• **Data compression:** Data compression is a method for reducing the amount or storage needs of data while maintaining the information that is most important. The data is compressed using several techniques such as wavelet compression, fractal compression [98].

These techniques for data reduction can help to make large datasets more manageable while retaining most of the original information. Understanding these techniques is important for effective data analysis, especially when dealing with big data.

2.5. Fundamental problems in supervised learning models

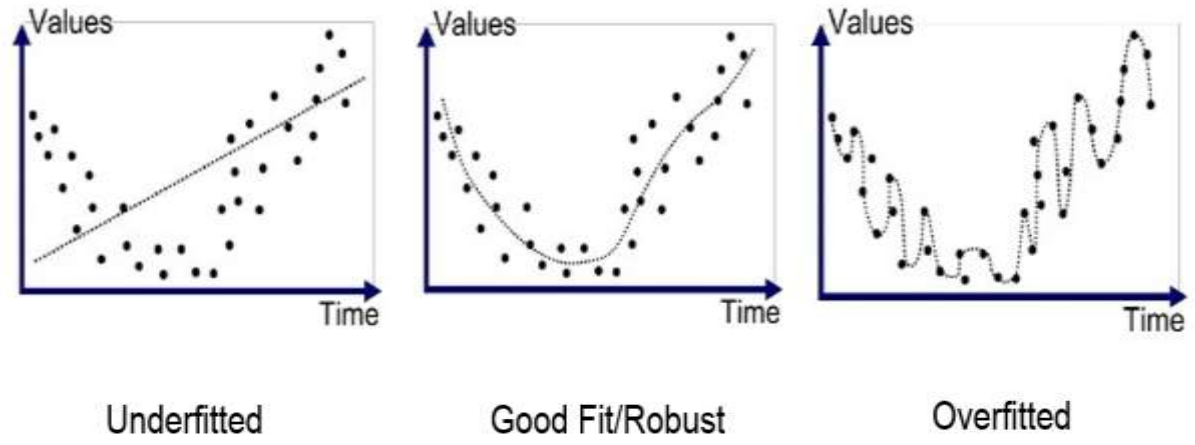


Figure 2.14. underfitting and overfitting [278].

2.5.1. Overfitting: It happens when a model gets overly complicated and learns the training data too well, which leads to poor generalization to new, unforeseen data (Figure 2.14). It is characterized by excessively low training error but high testing error [100].
To prevent overfitting problem, we can increase the training dataset size, use cross-validation, apply regularization techniques like L1 or L2 regularization, perform feature selection, consider early stopping, use simpler models, employ ensemble methods, and regularly evaluate the model's performance on unseen data. These strategies help prevent the model from memorizing noise or random variations in the training data, improving its ability to generalize and make accurate predictions on new data.

2.5.2. Underfitting: This occurs when a model is too simple to detect the fundamental patterns in the data. leading to high bias and high training and testing error (Figure 2.14). It fails to learn the training data and performs poorly on both training and testing data [100].
To prevent underfitting in supervised learning, increase model complexity, add relevant features, reduce regularization, increase training iterations, adjust hyperparameters, gather diverse data, and consider ensemble methods. These steps help address underfitting and improve the performance of supervised learning models.

2.5.3. Model Interpretability: Model interpretability refers to the ability to understand and explain how a model makes predictions or decisions, while it is an important challenge, many complex models lack it. Model interpretability involves gaining insights into the underlying factors or features that the model considers important in its decision-making process. Interpretable models provide clear and intuitive explanations, allowing humans to understand and trust the reasoning behind the model's outputs. Interpretability is important in various domains, such as healthcare, finance, and law, where the transparency of decisions and the ability to explain them are crucial for ethical, legal, and practical reasons.

2.6. Classification models

Classification models come in a wide variety of forms, such as decision trees, random forests, logistic regression, support vector machines, and neural networks. The best model to choose will depend on the particular issue at hand as well as the peculiarities of the data because each of these models has strengths and drawbacks of its own. Once trained, classification models can be used to forecast fresh data labels based on that data's characteristics. They are therefore suitable for a variety of applications.

2.6.1. Logistic Regression

A common classification approach used to estimate the likelihood that an input belongs to a particular class is called logistic regression (LR). It functions by applying a logistic function to the input data, which converts the features into a probability between 0 and 1. It is a straightforward algorithm with good performance that can handle both linear and non-linear input data. It is frequently utilized in a variety of fields, including marketing, healthcare, and credit scoring. The LR algorithm seeks to identify the logistic function coefficients that best suit the training data. In order to reduce the discrepancy between the anticipated probabilities and the actual labels, the algorithm modifies the coefficients during training. By calculating the likelihood of each class and choosing the class with the highest probability, the logistic regression model may be used to categorize new inputs after being trained [101,102,103].

A. LR algorithm

The LR algorithm takes as input a set of m training examples {(x1, y1), (x2, y2), ..., (xm, ym)}, where xi represents the input features of the i^{th} example, and yi represents the corresponding output label, which is either 0 or 1. In addition, the algorithm requires the specification of the learning rate α and the maximum number of iterations. The output of the algorithm is a set of learned parameters θ. The steps involved in the training process of LR:

1. **Initialize the parameters:** We start by initializing the weights and biases for the LR model.
2. **Define the cost function:** The cost function evaluates how well the LR model performs. It is usually described as the training data's negative log-likelihood. The reduction of the cost function is the aim of model training.
3. **Optimize the parameters:** To reduce the cost function and determine the ideal weights and biases, we employ an optimization approach like gradient descent. The cost function's gradients with respect to the weights and biases are calculated throughout each iteration of the optimization method, and the parameters are updated as necessary.
4. **Repeat until convergence:** The optimization procedure is iterated through a maximum number of times or until the cost function converges to a minimum.
5. **Evaluate the model:** After the LR model has been trained, we assess how well it performs using a different test dataset.

6. **Tune hyperparameters:** To enhance the model's performance on the test dataset, we can tweak hyperparameters like the learning rate and regularization strength.

B. Advantages and limitations of LR

The simplicity of LR is one of its key benefits. Even for those without substantial statistical background, it is rather simple to comprehend and apply. Furthermore, LR offers interpretable coefficients that enable us to comprehend how the independent factors and the dependent variable are related. Additionally, LR is a parametric model, which enables predictions to be made for new data that is similar to the data used to train the model. LR, however, also has significant drawbacks. Its assumption of a linear relationship between the independent factors and the log probabilities of the dependent variable is one of its key drawbacks. In reality, this may not always be the case, and LR may not produce reliable predictions when the connection is not linear. Furthermore, LR makes the assumption that the observations are independent, which may not necessarily be true in real-world situations. Last but not least, LR can only model binary outcomes and might not be appropriate for more complicated outcomes involving several categories.

2.6.2. Decision Trees

Popular classification and regression analysis algorithms include Decision Tree (DT). The algorithm builds a tree-like model of decisions and potential outcomes. According to the results of each decision, the tree branches out onto multiple paths (Figure 2.15). Each decision in the tree is based on the value of a certain trait. The idea behind decision trees is to split the data recursively based on the values of the input features until a particular stopping criterion is satisfied. Finding the feature and value that divides the data into distinct classes or accurately forecasts the target variable is the objective. The algorithm chooses the characteristic and value that best separates the data at each node of the tree and produces two or more branches. Each branch corresponds to a portion of the data that has been divided according to the feature value. A stopping condition, such as when the depth of the tree reaches a specific point or the number of samples in a node drops below a predetermined threshold, may serve as a stopping point for the process [104-107].

A. DT algorithm

The decision tree approach requires a set of m training examples, denoted as (x1, y1), (x2, y2),..., (xm, ym), where xi denotes the i^{th} example's input features and yi denotes the output label, which is a categorical value denoting the example's class. The algorithm also needs a stopping criterion to be specified, such as the maximum depth of the tree or the least amount of samples needed to split a node. A tree structure that may be used to categorize additional instances is the algorithm's output. Following is how the decision tree algorithm works:

1. Make the tree's root node.

2. Assign the majority class of the training instances to the current node if the stopping requirement is satisfied, such as the maximum depth of the tree being reached, or return if the minimum number of samples necessary to split a node is not fulfilled.
3. Decide which feature will best divide the data. The best feature can be chosen for this by calculating the information gain or Gini impurity for each feature.
4. Make a new internal node for the feature you've chosen.
5. Divide the training instances according to the chosen feature. A subset of the training instances with a specific value for the chosen feature are represented by each child node.
6. Repeat steps 2 through 5 on each child node until the halting requirement is satisfied.
7. Give the tree back.

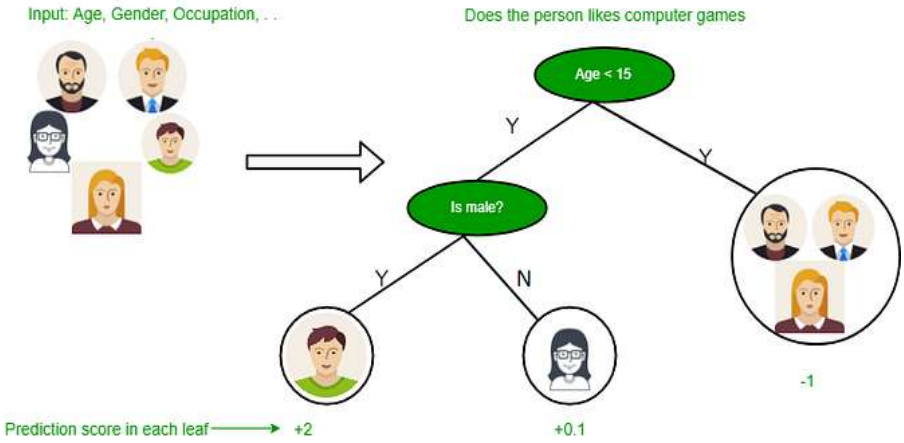


Figure 2.15. Decision Trees technique [108].

B. Advantages and limitations of DT

DTs have the benefit of being simple to comprehend, interpret, and visualize. The tree structure is a great option for multi-party decision-making procedures because non-technical stakeholders may understand it with ease. DTs can also work with numerical and categorical data, as well as missing values. The ability to employ DTs for feature selection is another benefit. The technique can determine which aspects of the data are most crucial, which can be useful for lowering the dataset's dimensionality and enhancing the effectiveness of other ML algorithms. DTs are resilient to outliers in the data and can handle non-linear correlations between variables. They are a suitable option for datasets with intricate interactions between the variables as a result. The fundamental drawback of DTs is their propensity for overfitting where a tree that is very complicated may fit the training data too closely and perform poorly on new, untainted data. By employing strategies like pruning, which entails removing branches from the tree that do not enhance its efficiency, this issue can be reduced. DTs can be biased toward variables that have a lot of different categories, which is another drawback, the accuracy of the model may be impacted if this results in an unequal weighting of the variables. Trees that make decisions can be sensitive to even little data changes; as a result, it

may be challenging to evaluate the model if the data marginally alter the tree's entire structure.

2.6.3. Random Forest

A group of decision trees called random forests (RF) are trained on various subsets of data (Figure 2.16). They have been proven to be quite efficient in practice and can be utilized for both classification and regression issues. Building several decision trees, each trained on a distinct subset of data and a random subset of characteristics, is how random forests function. The final prediction is determined by taking a majority vote or averaging the predictions of all the trees. This technique reduces overfitting and increases accuracy [109,110].

A. RF algorithm

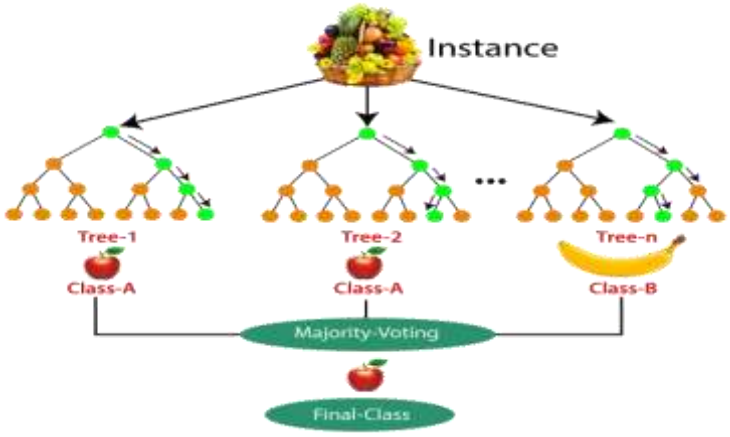


Figure 2.16. Random Forest technique [111].

A series of m training examples ((x1, y1), (x2, y2),..., (xm, ym)) is used as input by the random forest algorithm, where xi denotes the input features of the i^{th} example and yi denotes the matching output label. The algorithm also needs the number of trees to be produced, the size of the random subset of features to take into account at each split, and a criterion for splitting that will select which attribute is appropriate to split on at each node. The algorithm produces a random forest F with a certain number of trees as its result. The following is the random forest's training algorithm:

1. Specify how many trees there are in the forest, their maximum depth, and the amount of attributes that should be taken into account when determining the appropriate split at each node.
2. For every forest tree:
    a. Create a bootstrap sample at random from the training data that is the same size as the original data.

   b. Construct a decision tree using the bootstrap sample, selecting the split based on the feature with the highest information gain or Gini impurity at each node of the tree.
   c. Extend the tree until it reaches its maximum depth or until it has fewer samples in each leaf node than a predetermined amount.
3. Return random forest consisting of all decision trees.

The random forest averages all of the decision trees' forecasts throughout prediction to get a final prediction.

B. Advantages and limitations of RF

Their resistance to noise and outliers is one of their key advantages. RFs can lessen the influence of individual trees that can be overfitting the data by mixing several decision trees. This can result in forecasts that are more accurate, which is another benefit of RFs. Using numerous decision trees helps improve the algorithm's accuracy, which is especially useful for complex datasets with many variables. Another advantage of RFs is that they can provide information about feature importance, the data can be better understood with this information and identify important variables for future analysis. Additionally, RFs can handle large datasets with high dimensionality without overfitting or increasing computation time significantly, this scalability is important for applications where data is growing rapidly and needs to be analyzed quickly. Despite their advantages, RFs have some limitations to consider. One limitation is the difficulty in interpreting the results. Because RFs use multiple decision trees, it can be hard to understand how individual trees contribute to the final prediction. Overfitting can also be an issue with RFs, although they typically exhibit lower overfitting risk than individual decision trees. However, overfitting is still possible if the number of trees is too high or the depth of the trees is too deep. Another potential limitation of RFs is their computational expense. RFs can be computationally expensive, especially when there are many trees or when the dataset is very large. When one class is significantly more numerous than the other in the data, they could also do poorly. The performance of RFs can be improved by tuning a number of hyperparameters. This can take some time and calls for some algorithmic expertise.

2.6.4. Support Vector Machines

A common model for classification issues is the support vector machine (SVM). They look for the hyperplane that divides the data into different classifications the most effectively. SVM identifies the border that maximizes the separation between the nearest points of various classes. The data points nearest to the decision boundary are used to define these support vectors, which serve as the hyperplane (Figure 2.17). A linear hyperplane can better separate the data in a higher-dimensional space created by SVM using a kernel function. The fundamental idea of SVM is to identify the hyperplane, also known as the hinge loss that maximizes the margin and minimizes the error. SVM divides the data into two classes

because it is a binary classifier. One-vs-one or one-vs-all strategies can be used to extend it to tackle multi-class classification issues [112-114].

A. SVM algorithm

The kernel function K is used to compute the inner product of feature vectors in a high-dimensional space, and the regularization parameter regulates the trade-off between maximizing the margin and minimizing the training error. The SVM algorithm requires as input a dataset D containing m instances (x1, y1), (x2, y2),..., (xm, ym), where each instance consists of a set of input features xi and an output label yi. The algorithm produces a hyperplane H in the feature space that, to the greatest extent possible, divides examples of various classes. The SVM training algorithm's steps are outlined in the following order:

1. Choose a kernel function and kernel parameters.
2. Initialize the model parameters (weights and biases) to small random values.
3. For a predetermined number of iterations or until convergence is reached:
   a. Randomly shuffle the training data.
   b. For each training example:
      i. Apply the current model parameters to the input features to calculate the predicted class.
      ii. Compute the loss between the predicted class and the true class label.
      iii. Update the model parameters to minimize the loss using gradient descent or another optimization algorithm.
4. Return the trained SVM model with the learned parameters.

B. Advantages and limitations of SVM

SVM has several advantages and limitations. One of the significant advantages of SVM is its effectiveness in handling high-dimensional data such as text and image data. SVM converts the data into a higher-dimensional space using kernel functions, making it more effective in separating the data. SVM is also memory efficient, as it uses only a subset of training points to define the hyperplane, making it more efficient than other algorithms. Moreover, SVM is robust to outliers and noisy data, as it maximizes the margin between the closest points of different classes. SVM is versatile as it can handle both linear and non-linear data by choosing appropriate kernel functions. It also performs well on small to medium-sized datasets with a few hundred to a few thousand data points. However, SVM has several limitations. One of the major limitations is that SVM performance is heavily dependent on the choice of kernel function, which can result in poor performance if selected inappropriately. SVM can be computationally intensive, especially when using non-linear kernel functions, making it time-consuming for large datasets. Additionally, SVM does not provide probability estimates directly, which can be problematic in certain applications. Additionally, for SVM to function at its best, a number of hyperparameters must be set, which can be difficult and requires some understanding of the algorithm. Lastly, SVM is primarily a binary classification algorithm and

may require extensions for multi-class classification problems, which can be complex and time-consuming.

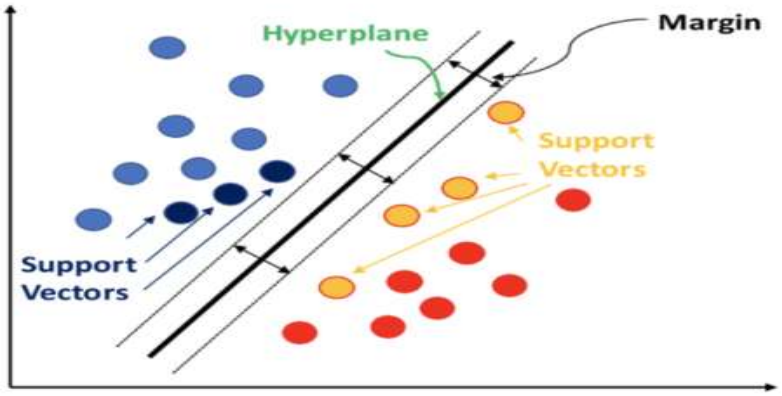


Figure 2.17. Support Vector Machine technique [115] .

2.6.5. Naive Bayes

For text categorization issues, the straightforward probabilistic model known as Naive Bayes (NB) is frequently utilized. The NB method bases its estimation of the likelihood of an event happening on the likelihood of related events. The algorithm is referred to as "naive" since it makes the assumption that each attribute exists independently of the others. In other words, it is predicated on the idea that the existence of one feature has no bearing on the existence of any other feature. This supposition speeds up the procedure and makes the calculating of probability simpler. The approach computes the conditional probability for each feature given the class and the prior probability for each class. The posterior probability of each class given the input information is then estimated by combining these probabilities. The predicted class is the one with the highest posterior probability. For both classification and regression issues, NB can be employed. The algorithm determines the likelihood of each class given the input features and selects the class with the highest likelihood as the projected class for classification. With respect to regression, the technique calculates the output variable's conditional probability distribution in light of the input features [116-118].

A. NB algorithm

The NB algorithm requires a collection of m training instances, denoted as (x1, y1), (x2, y2),..., (xm, ym), where xi denotes the i^{th} example's input features and yi denotes an output label. The algorithm also needs the input features F = f1, f2,..., fn to be specified. A classifier that can predict the output label of new instances is the output of the algorithm. Following is how the NB algorithm works:

1. Compute the prior probability of each class by dividing the number of instances belonging to that class by the total number of instances.

2. For each feature and each class, compute the conditional probability of that feature given that class by counting the number of instances belonging to that class that have that feature divided by the total number of instances belonging to that class.
3. Use the computed prior and conditional probabilities to compute the posterior probability of each class given the features of a new instance using Bayes' theorem. Specifically, multiply the prior probability of each class by the product of the conditional probabilities of the features given that class, and normalize the resulting probabilities to sum to 1.
4. Assign the new instance to the class with the greatest posterior probability.

B. Advantages and limitations of NB

One of the significant advantages of NB is its simplicity, making it easy to implement and understand, and suitable for quick and efficient classification tasks. NB also has a fast training and prediction speed, making it ideal for real-time or large-scale applications. It is scalable to large datasets, making it a suitable choice for high-dimensional problems. Moreover, NB requires relatively low memory, making it suitable for applications with limited memory resources. It is also less prone to overfitting compared to other complex algorithms. However, NB also has several limitations. One of the major limitations is the strong feature independence assumption, which can lead to suboptimal results when the features are highly correlated. NB is also a simple algorithm that cannot capture complex relationships between features, limiting its expressiveness. It cannot handle missing data and requires the removal or imputation of missing values. The size of the features affects NB, which may require feature scaling or normalization. Lastly, NB may not perform well on classes with very few instances, as it may not accurately estimate their probabilities, leading to poor performance on rare classes.

2.6.6. K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a straightforward technique that works well for classification and regression issues. Based on the features of its k-nearest neighbors in the feature space, it predicts the output variable (Figure 2.18). The underlying premise of KNN is that related objects tend to cluster together. In other words, it determines, depending on distance, the k-nearest neighbors to a given data point and utilizes them to predict. The procedure locates the k-nearest neighbors to the input data point and assigns it to the class that appears most frequently among the k-neighbors in order to use KNN for classification. Different distance metrics, such as Euclidean, Manhattan, or cosine distance, are used to determine the separation between the data points. A hyperparameter that the user must select is the value of k. The algorithm locates the KNN and computes the average or weighted average of their target values as the predicted value for the input data point when using KNN for regression. KNN is a lazy algorithm, meaning that it does not require training on the entire dataset but rather stores the data points to be used for prediction during runtime. This makes the algorithm suitable for large datasets but can result in slower prediction times for large k values [119,120].

A. KNN Algorithm

KNN algorithm requires a dataset D containing m instances (x1, y1), (x2, y2),..., (xm, ym) where each instance consists of a set of input features xi and an output label yi, a positive integer k representing the number of nearest neighbors to take into account for each new instance, and a distance metric function dist(x, xi) that determines the distance between two instances x and xi. A classifier that can forecast the output label of new instances is the algorithm's output. The steps of the KNN algorithm are as follows:

1. For a new instance x with input features x1, x2, ..., xn, compute the distance between x and each training instance xi:
   a. Calculate the distance metric dist(x, xi) between x and xi.
   b. Associate each distance with its corresponding output label yi.
2. Choose the k closest neighbors after sorting the distances in ascending order.
3. Compute the output label of the new instance x as follows:
   Select the most common label among the k nearest neighbors.
4. Return the predicted output label.

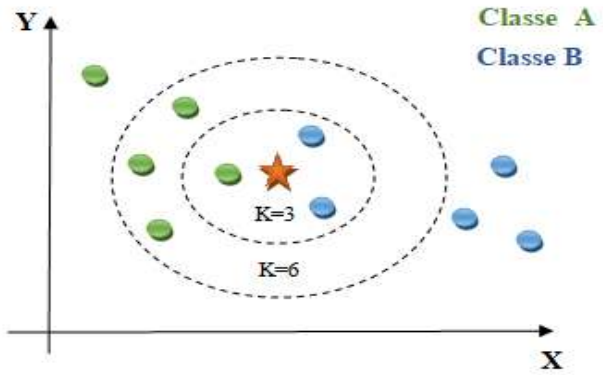


Figure 2.18. KNN technique.

B. Advantages and limitations of KNN

KNN is a simple yet powerful ML algorithm that has several advantages and limitations. One of KNN's key benefits is its simplicity, which makes it simple to use and comprehend. KNN can handle both linear and non-linear data and can work well for both small and large datasets. It also provides interpretable results, making it easy to understand how a prediction was made. KNN does not require training on the entire dataset, making it suitable for large datasets, and it can handle classification problems with multiple classes. However, KNN also has several limitations. One of the major limitations is the sensitivity to the choice of distance metric, and different distance metrics may produce different results. When there are many characteristics or dimensions, KNN may experience the curse of dimensionality, leading to sparsity and difficulty in finding meaningful neighbors. It is necessary to choose an adequate value for k, as choosing an incorrect value can lead to poor performance. As the value of k

increases, the prediction time of KNN can increase significantly, leading to slow prediction time for large k. Lastly, KNN can be biased towards the majority class in imbalanced datasets, making it challenging to handle imbalanced data.

2.6.7. Gradient Boosting

Gradient boosting (GB) is an ensemble technique for combining various weak models (often decision trees) into a strong model (Figure 2.19). It operates by training fresh models to attempt to fix the flaws in the older models. GB is a technique used to strengthen a weak model by adding successively stronger models that can fix the flaws in the weaker models. An easy model, such as a decision tree, is first fitted to the training data by the algorithm. The mistakes of this initial model are then determined, and a new model is trained using its residuals. The new model is included in the ensemble, and the procedure is repeated up to the point where performance is adequate or the allotted number of iterations has been used. GB uses a gradient descent approach to train each new model to reduce the errors of the prior models. By determining the direction and size of the steepest drop in the error function, the gradient descent technique enables the new model to modify its parameters in a way that minimizes errors the greatest. The ensemble's forecasts are combined to get the final prediction, typically by averaging or using a weighted average [121-123].

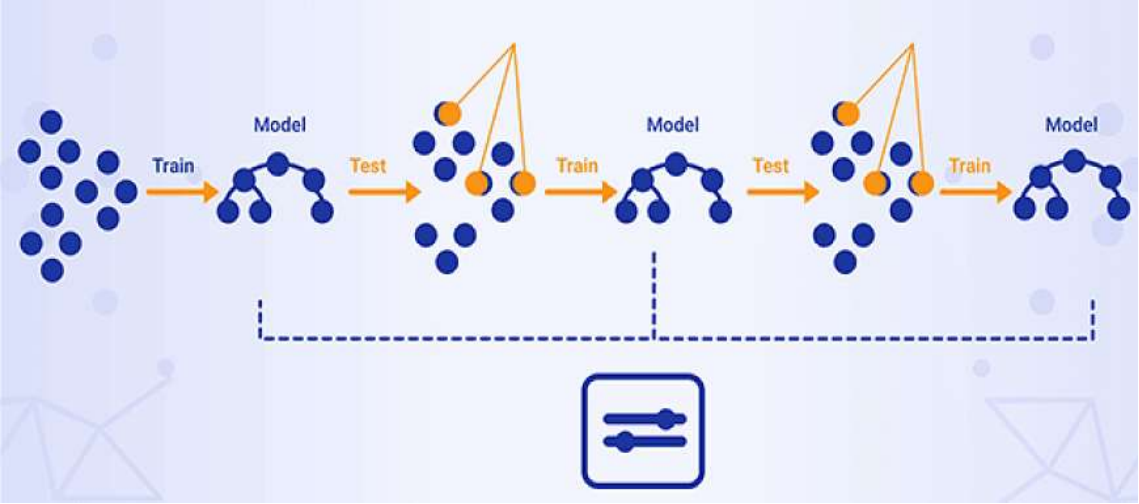


Figure 2.19. Gradient Boosting technique [124].

A. GB Algorithm

GB Algorithm uses gradient descent to optimize the objective function and minimize the errors of the model. Each new model is trained to approximate the negative gradient of the objective function with respect to the predictions of the previous models, which is a measure of the direction and magnitude of the error that needs to be corrected. The gradient descent updates the weights of the new model by minimizing the objective function along the direction of steepest descent. GB algorithm works as follows:

1. Define the loss function to optimize, such as mean squared error for regression or binary cross-entropy for classification.
2. Define the base estimator, which can be any ML model that can make predictions, such as decision trees or linear models.
3. Initialize the ensemble by fitting the base estimator to the data and computing the negative gradient of the loss function with respect to the predictions of the base estimator. These negative gradients are the residuals between the true labels and the predictions of the base estimator.
4. Fit a new base estimator to the negative gradients computed in the previous step. The goal is to predict the negative gradients with the new base estimator and add these predictions to the current ensemble.
5. Update the ensemble by adding the predictions of the new base estimator multiplied by a learning rate, which controls the contribution of the new estimator to the ensemble.
6. Repeat steps 4 and 5 until a stopping requirement, such as a maximum number of estimators or a minimal increase in the validation score, is satisfied.
7. Pass the new instance through each estimator in the ensemble and add their guesses together to produce a prediction for it.

B. Advantages and limitations of GB

GB has both advantages and limitations. One of its primary advantages is its ability to achieve high predictive accuracy on a wide range of datasets and outperform other algorithms. It is also tolerant of outliers and noisy data and can handle a variety of data formats, including continuous, categorical, and ordinal variables. Additionally, GB can do feature selection to determine the most crucial characteristics in the dataset and capture complex and non-linear correlations between input features and output variables. GB does have its limitations, though. For large datasets or intricate models, the approach may be computationally expensive. Additionally, it is prone to overfitting, particularly if the model is overly complicated or the hyperparameters are not calibrated properly. It can take a lot of time and careful consideration to tune hyperparameters like learning rate, tree count, and depth. It can also be tricky to understand the final model created by GB, making it difficult to explain how the algorithm arrived at its predictions. Finally, GB can be sensitive to noisy data, which can result in poor performance if the noise is not appropriately handled.

2.6.8. Artificial Neural Networks

In both classification and regression issues, neural networks constitute a versatile and effective class of models. The input information and output predictions are processed by layers of neurons that make up these systems. Artificial neural networks come in a variety of shapes and sizes, but the following are the ones that are most frequently used for classification:

2.6.8.1. Multilayer Perceptron

In ML and pattern recognition applications, Multilayer Perceptrons (MLPs) are a common type of neural network (Figure 2.20). The foundation of the MLP is the idea of a feedforward neural network, which refers to a neural network in which information only goes in one direction, from the input layer to the output layer, without any feedback connections. There are several node layers that make up the MLP, including an input layer, one or more hidden layers, and an output layer. Every node in the input layer stands in for a feature of the input data, and every node in the output layer stands for a class or value of the output data. Using a series of nonlinear transformations, the nodes in the hidden layers take the input data and extract higher-level characteristics that can be utilized to produce a prediction or classify the data. The MLP's weights and biases are modified during training in order to reduce the discrepancy between the expected and actual output. Backpropagation, a method for accomplishing this, involves computing the gradient of the loss function with respect to the weights and biases and updating them using gradient descent or a similar optimization algorithm [125-127].

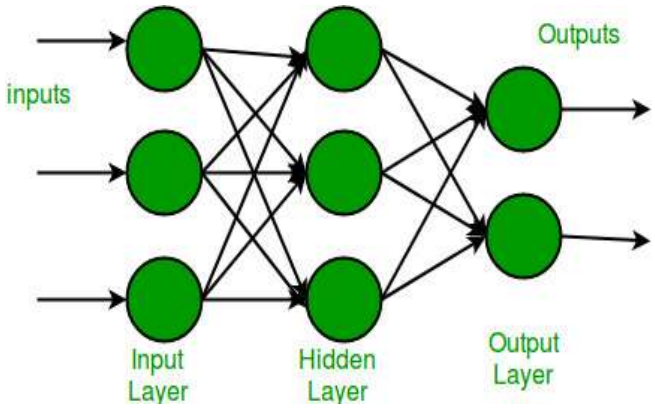


Figure 2.20. MLP architecture [128].

A. MLP Training process

Backpropagation is a form of supervised learning technique that is used to train the MLP. The set of m training examples ((x1, y1), (x2, y2),..., (xm, ym)) is the input for the backpropagation algorithm. Here, xi denotes the input features of the i^{th} example and yi denotes the matching output label. The method additionally needs a learning rate to be specified. A set of learnt weights and biases from the method are produced as its output, and these weights and biases are then applied to additional input cases to generate the network's output. Following are the steps in the backpropagation algorithm:

1. **Initialize the network**: Each neuron in the MLP is given random weights and biases at startup.

2. **Forward propagation**: The network receives the input data and uses its weights and biases to determine each neuron's output. The input for the next layer is the output of the previous layer.
3. **Calculate the error**: A loss function is utilized to calculate the difference between the output that was expected and the output that was actually produced.
4. **Backpropagation**: Using the chain rule of calculus, the error is spread backwards through the network. The gradient of the loss function with respect to the weights and biases in each layer is computed using the error.
5. **Update the weights and biases**: An optimization approach is used to update the weights and biases in each layer. A portion of the gradient is subtracted from the existing weights and biases to calculate the new weights and biases. The learning rate hyperparameter regulates the fraction.
6. **Repeat steps 2 through 5**: Steps 2 through 5 are performed until the error on a validation set stops reducing or for a predetermined amount of iterations.
7. **Make use of the learned MLP**: After the MLP has been trained, you can use it to forecast the results of fresh input data by putting it through the network and calculating the results of the top layer.

B. Advantages and limitations of MLP

MLP is a potent ML model that can understand intricate nonlinear relationships between inputs and outputs, generalize effectively to new data, and manage high-dimensional data. In addition, it may be parallelized for quick processing and can learn features in a hierarchy if the input data distribution or task requirements change. When the model is too complicated or the dataset is too short, the MLP is prone to overfitting, and training it can be computationally expensive for big and deep networks. An MLP's performance is dependent on how the weights are initialized, and because it's a black box, it might be challenging to interpret. The MLP also does not explicitly reflect the temporal connections between the input and output data, which makes it unsuitable for processing sequential data, such as time series or text data.

2.6.8.2. Deep Neural Networks

Deep neural networks (DNNs) are widely used to describe MLPs with plenty of hidden layers. (Figure 2.21). "Deep" refers to the network's depth, which is the sum of the layers between the input and output layers. The number of hidden layers in a DNN can vary, but typically networks with more than three hidden layers are considered deep. The main difference between Deep Neural Networks (DNNs) and Multilayer Perceptrons (MLPs) is: MLPs usually have only one or two hidden layers, while DNNs have many more layers, often ranging from 3 to more than 100. The additional hidden layers in DNNs allow them to model very complex nonlinear relationships between the input and output. Due to its success in a variety of applications, such as image identification, natural language processing, and speech recognition, deep neural networks have grown in popularity in recent years. These applications include their capacity to understand complicated characteristics and patterns in

data. In summary, the DNN expands on the MLP's basic idea by adding extra layers that let the network recognize more intricate and abstract elements in the input data [125,129].

A. Training process of DNNs

The algorithms used for training DNNs and MLPs are similar in many ways, but there are some key differences. Both types of networks use the backpropagation algorithm to adjust the weights and biases of the network during training. However, there are some differences in the implementation of the backpropagation algorithm for DNNs and MLPs. One of the key differences is that DNNs are typically much larger and deeper than MLPs, with more parameters to be optimized. As a result, training a DNN can be much more computationally expensive and requires specialized optimization techniques, such as batch normalization or dropout, to prevent overfitting. In addition, DNNs often use more sophisticated activation functions, such as rectified linear units (ReLU), which can help to mitigate the vanishing gradient problem that can occur with deep networks. So while the basic principles and algorithms used in training DNNs and MLPs are similar, the implementation details and optimization techniques used can differ significantly due to the increased complexity and depth of DNNs.

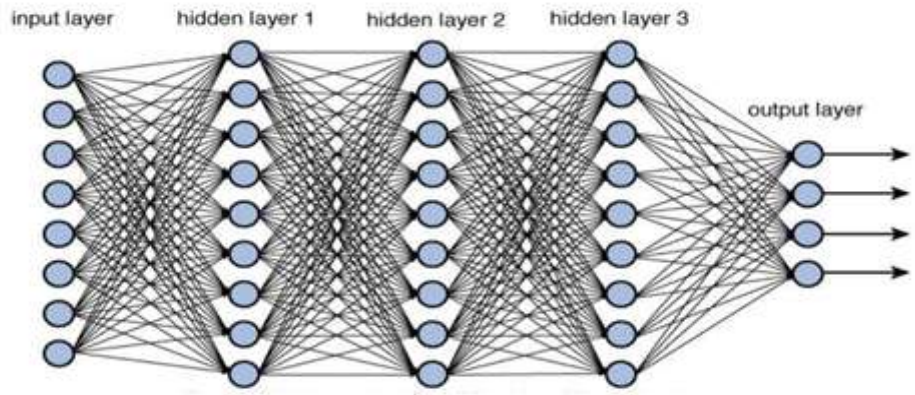


Figure 2.21. Deep Neural Network [279].

B. Advantages and limitations of DNNs

DNNs have the benefit of being able to learn more complicated and abstract features from high-dimensional data since they are built on MLP principles. They don't require as much feature engineering because they can automatically learn features that are pertinent to the current task. DNNs are able to generalize well to new data and can scale up to handle very huge datasets. On a variety of tasks, such as speech recognition, image classification, and natural language processing, they have attained state-of-the-art performance. DNNs, as opposed to MLPs, are more equipped to handle complex and structured data, such as images and sequences, because they can learn hierarchical representations of the data. DNNs are prone to overfitting when the model is too complicated or the dataset is too short, and training them can be computationally expensive, especially for big and deep networks. DNNs are also

more challenging to comprehend than MLPs since it is more difficult to grasp how they make predictions due to the more opaque internal workings of DNNs.

2.6.8.3. Convolution Neural Networks

Convolution Neural Networks (CNNs) are a particular class of neural networks that are frequently employed for image and video recognition applications. To extract features from the input photos, it employs convolutional layers. A CNN's main principle is to employ a number of convolutional layers, each of which applies a particular set of filters to the input data to extract features pertinent to the task at hand (Figure 2.22). Through a training procedure that involves modifying the neural network's weights to reduce the discrepancy between the expected output and the actual output, these filters are learned. The spatial dimensionality of the feature maps is decreased and overfitting is prevented when data is downscaled through pooling layers as it moves through the network. One or more fully connected layers of the CNN combine the results of the earlier levels to produce the final output, which is a prediction. The weights of the fully connected layers are changed during training in order to reduce the discrepancy between the expected output and the actual output [130-133].

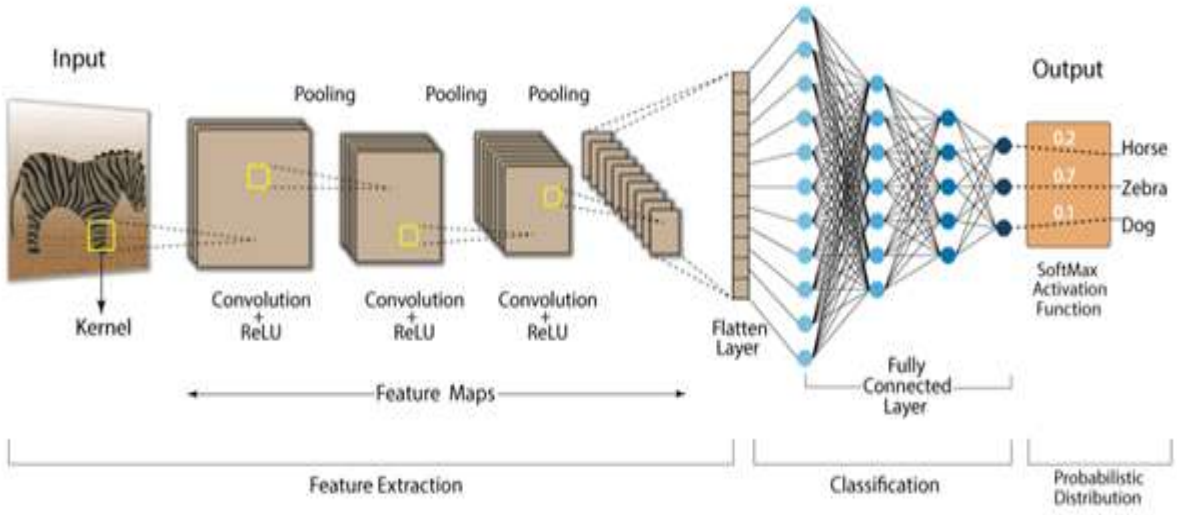


Figure 2.22. Convolution Neural Networks [134].

A. Training process of CNN

After Define the CNN's architecture, mentioning the number of convolutional layers, the number of filters in each layer, the size of the filters, the convolution's stride, the activation function for each layer, the pooling function, the window size, the number of fully connected layers, the number of neurons in each layer, and the activation function for the fully connected layers. Additionally, select the output layer's loss function; CNN is trained using a backpropagation algorithm. The approach uses a dataset D with m instances (x1, y1), (x2, y2),..., (xm, ym) as input. Each instance is made up of a collection of input features (xi) and

an output label (yi). To anticipate the output label of new instances, the CNN is trained. The following is how the backpropagation algorithm works:

1. **Initialize the network:** The convolutional and pooling layers, as well as every neuron in the network, are initialized with random weights and biases for the CNN.
2. **Forward propagation**: The input data is sent into the network, and each neuron's output is calculated using the network's weights and biases. A collection of feature maps are produced for each layer in a CNN by applying a set of filters to the input data.
3. **Calculate the error**: A loss function is utilized to calculate the difference between the output that was expected and the output that was actually produced.
4. **Backpropagation**: Using the chain rule of calculus, the error is spread backwards through the network. This is done in a CNN by calculating the gradient of the loss function in relation to the weights and biases of each layer as well as the gradient in relation to the input data.
5. **Update the weights and biases**: An optimization approach like stochastic gradient descent is used to update the weights and biases in each layer. A portion of the gradient is subtracted from the existing weights and biases to calculate the new weights and biases. The learning rate hyperparameter regulates the fraction.
6. **Repeat steps 2 through 5**: Steps 2 through 5 are performed until the error on a validation set stops reducing or for a predetermined amount of iterations.
7. **Employ the trained CNN**: After the CNN has been trained, it can be used to forecast the results of new input data by submitting it to the network and calculating the output.

B. Advantages and limitations of CNN

In comparison to other ML models for image processing, CNN has a number of advantages, such as the capacity to handle spatial invariances and distortions in the input and the ability to automatically build hierarchical representations of visual data. On a variety of image-related tasks, such as picture classification, object recognition, and segmentation, CNNs have reached state-of-the-art performance. The training of CNNs can be computationally expensive, particularly for big and deep networks, and a lot of labeled data is necessary for them to perform well. CNNs are less interpretable than some other ML models, which make it challenging to comprehend how the model generates its predictions. Additionally, CNNs might be vulnerable to overfitting if the model is too complicated or the dataset is too short.

2.6.8.4. Recurrent Neural Networks

A particular class of neural network called recurrent neural networks (RNN) is frequently employed for sequential data processing in categorization applications like natural language processing. Information is maintained across successive time steps by using a feedback loop (Figure 2.23). RNNs have a loop in which the results of a one-time step are sent back into the network as the input for the following time step, in contrast to feedforward neural networks, which process input data in a single run through the network. As a result, RNNs can keep a

hidden state that records data for the whole series that has been seen up to this point and utilizes that data to forecast what will come next in the sequence. The fundamental unit of an RNN is a straightforward neuron with a self-recurrent connection, but more sophisticated architectures, like Long Short-Term Memory and Gated Recurrent Unit networks, have been created to address the vanishing gradient problem that can arise during training of conventional RNNs [135-137].

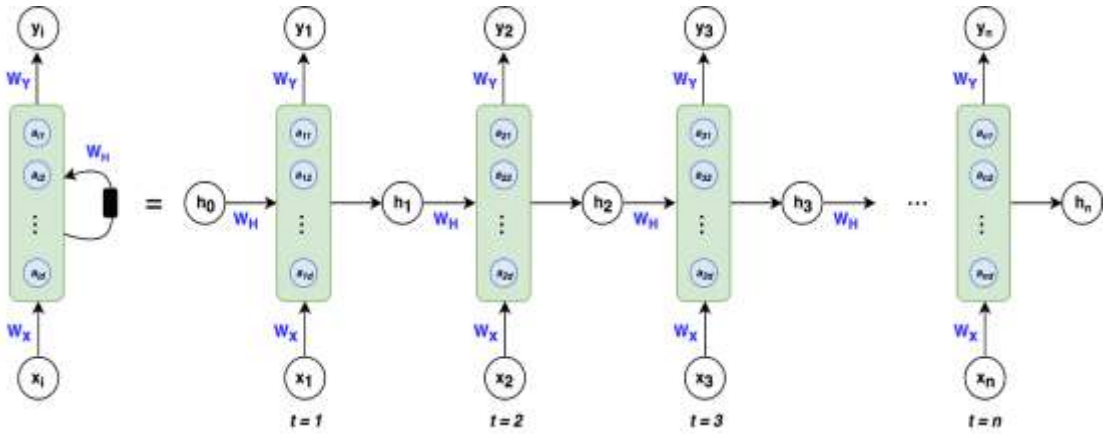


Figure 2.23. Recurrent Neural Networks [138].

A. Training process of RNN

The RNN's architecture must first be defined, including the number of layers, the number of neurons in each layer, and the function used to activate each layer. Additionally, select the output layer's loss function. The following is how a basic RNN is trained:

1. **Create the network's starting state**: The RNN is created with random weights and biases for each neuron.
2. **Forward propagation**: The input sequence is fed into the network one element at a time, and each neuron's output is computed using its weights and biases. With the output from one time step serving as the input for the following, this procedure is repeated for each time step.
3. **Calculate the error**: A loss function is utilized to calculate the error at each time step by comparing the predicted output to the actual output.
4. **Backpropagation through time:** The calculus chain rule is used to propagate the error backwards over time. Each time step's weights and biases are taken into account when computing the gradient of the loss function using the error.
5. **Update the weights and biases**: Each time step's weights and biases are adjusted via an optimization process. By deducting a portion of the gradient from the existing weights and biases, the new weights and biases are calculated. The hyperparameter for learning rate controls the fraction.
6. **Repeat steps 2 through 5**: Steps 2 through 5 are performed a predetermined number of times or up until the error on a validation set stops dropping.

7. **Employ the trained RNN:** After being trained, the RNN can be used to forecast the results of new input sequences by submitting them to the network and calculating the results of the last time step.

B. Advantages and limitations of RNN

For tasks requiring sequential data processing, RNNs have a number of benefits over conventional ML models. In order to produce predictions based on the context of the entire sequence, they can handle variable-length input sequences and maintain a hidden state that captures data about the entire sequence observed thus far. The performance of RNNs on many of these tasks, including time-series prediction, speech recognition, and natural language processing, is state-of-the-art. RNNs can, however, be expensive to train computationally, particularly for big and complicated datasets. The vanishing gradient problem, which can happen when gradients are too small during backpropagation and make it challenging to train long-term dependencies, might also affect them. More sophisticated RNN architectures, like Gated Recurrent Unit and Long Short-Term Memory networks, have been created to solve this issue. Finally, RNNs may be more difficult to interpret than other ML models, making it challenging to comprehend how the model predicts the future.

2.6.8.4. Long Short-Term Memory

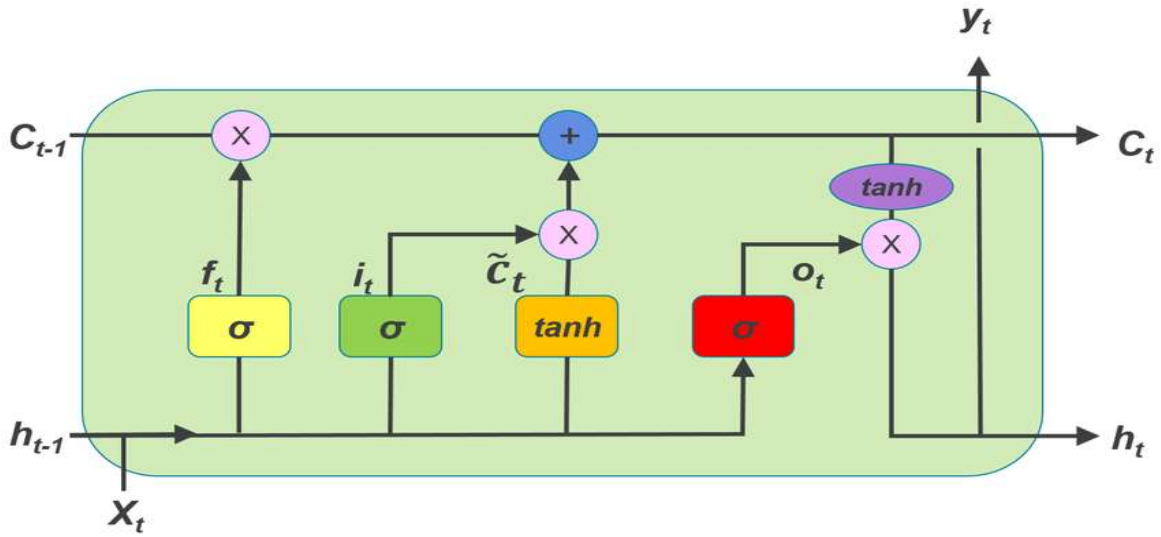


Figure 2.24. Long Short-Term Memory [139].

Long-term dependencies are handled by LSTMs, a particular kind of RNN. It uses memory cells to store information over numerous time steps (Figure 2.24). The idea behind Long Short-Term Memory (LSTM) is to solve the issue of disappearing gradients in conventional RNNs by employing a more intricate cell structure that enables the network to choose store or forget information over time. A forget gate, an output gate, and three input gates make up the LSTM cell's three primary parts. The output gate determines how much of the cell state is used to create predictions, while the input gate determines how much old information is kept

in the cell state and the forget gate determines how much new information is added to it. These gates work as nonlinear functions that take the current input, the prior hidden state, and a bias term as inputs and output a value between 0 and 1 that represents how much information is allowed through. A hidden state that captures long-term dependencies in the input sequence can be maintained by LSTMs throughout time by selectively preserving or forgetting information. Using this knowledge, LSTMs can use the input sequence to predict outcomes accurately. The performance of LSTMs on many of these tasks has been demonstrated to be state-of-the-art, and they are frequently utilized in tasks involving speech recognition, natural language processing, and other types of sequence modeling [140-143].

A. Training process of LSTM

The architecture of the LSTM, including the quantity of LSTM cells, the dimensions of the input and output layers, and the activation functions for each layer, are defined as the initial stage. LSTM training procedure:

1. **Set up the LSTM network**: The weights and biases for each gate and memory cell are randomly assigned when the LSTM network is first set up.
2. **Forward propagation**: The input sequence is fed into the LSTM network, and the hidden state and output are calculated at each time step using the weights and biases. The full sequence is completed by repeating this method.
3. **Determine the error**: A loss function is used to calculate the discrepancy between the projected output and the actual output and to calculate the error.
4. **Backpropagation via time**: Using the chain rule of calculus, the error is propagated backwards through time. The gradient of the loss function with respect to the weights and biases in each gate and memory cell is calculated for each time step using the error.
5. **Update the weights and biases**: An optimization technique is used to update the weights and biases in each gate and memory cell. A portion of the gradient is subtracted from the existing weights and biases to calculate the new weights and biases. The learning rate hyperparameter regulates the fraction.
6. **Repeat steps 2 through 5**: Steps 2 through 5 are performed until the error on a validation set stops reducing or for a predetermined amount of iterations.
7. **Employed the trained LSTM** By feeding new input sequences through the network and computing the output and hidden state for each time step after the LSTM has been trained, it is possible to use it to predict the output of new input sequences.

B. Advantages and limitations of LSTM

When it comes to sequence modeling tasks, LSTM networks have a number of advantages over traditional RNNs. For tasks like natural language processing and speech recognition, where the meaning of a sentence can depend on words that appear far apart from one another, LSTMs can handle long-term dependencies in input sequences. LSTMs are simpler to train

and more successful at modeling complicated sequences because they are resistant to the vanishing gradient problem that can arise during RNN training. LSTMs are particularly suited to modeling sequences where some information is more essential than other information because they may selectively store or forget information over time. A number of topologies, including encoder-decoder models, can employ LSTMs to solve more difficult sequence modeling problems. However, training LSTMs can be computationally costly, particularly for big datasets with several parameters. If they are not sufficiently regularized, they may also be vulnerable to overfitting. Finally, it can be tricky to comprehend how the network generates its predictions because LSTMs can be challenging to interpret.

2.6.8.5. Deep Belief Network

Deep Belief Network (DBN) is an artificial neural network that learns a compressed representation of input data using multiple layers of Restricted Boltzmann Machines (RBMs) (Figure 2.25). Each RBM layer of a DBN is trained in an unsupervised manner to reconstruct the input data. RBMs are generative models that encode the input data into binary values using hidden units. RBMs modify the weights of the network using the Contrastive Divergence method to boost the likelihood that the input data can be reconstructed. Because RBMs learn in a hierarchical manner, each layer can learn at a different degree of abstraction—lower layers can learn straightforward characteristics like edges and corners, while higher layers can learn more intricate features like forms and objects. DBNs can be improved by utilizing supervised learning for classification problems after unsupervised pre-training. With RBMs modeling the probability distribution of input data and each RBM layer learning to represent input data at different degrees of abstraction, the basic idea behind DBN is to learn a hierarchy of representations of input data [144-146].

A. Training process for DBN

Pre-training and fine-tuning steps are included in the training process for DBN:

1. **Pretrain each layer as a RBM**: The weights and biases of each layer are first set at random, and contrastive divergence is used to conduct a layer-by-layer unsupervised training procedure. Every layer receives training to rebuild the input data and identify key features. Each layer's output is then fed into the following layer as its input.
2. **Use backpropagation to fine-tune the network**: After all layers are pretrained, the network as a whole is adjusted using labeled data and backpropagation. To reduce the difference between the expected and actual results, all layers' weights and biases are updated simultaneously.
3. **Carry out steps 1-2** repeatedly over a number of epochs until the network reaches a desirable level of performance.
4. **Use the taught Deep Belief Network**: The trained Deep Belief Network can be applied to a variety of tasks, including feature extraction, regression, and classification.

While preventing the issue of vanishing gradients that can arise in deep neural networks during backpropagation, the pre-training step enables the network to acquire meaningful characteristics from the data. The fine-tuning procedure enhances the network's capacity to categorize fresh data.

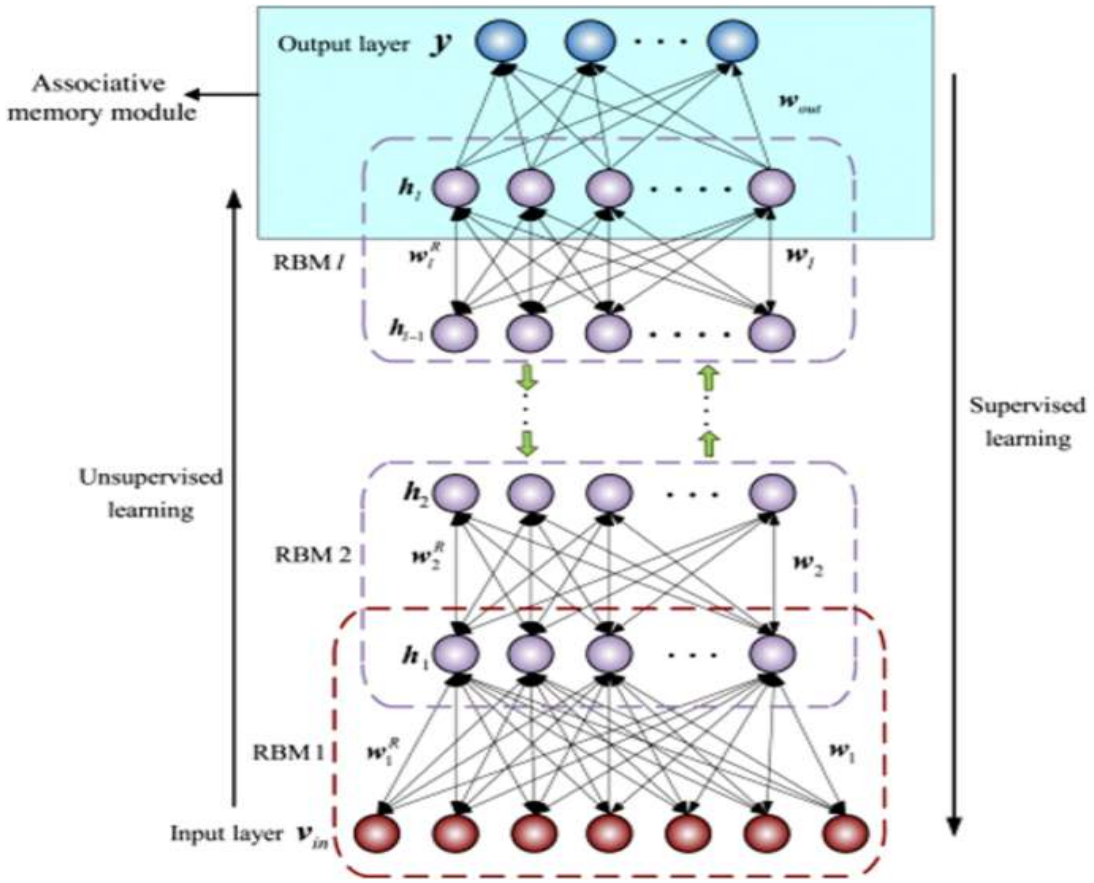


Figure 2.25. Deep Belief Network [147].

B. Advantages and limitation of DBNs

DBNs have several advantages over traditional neural networks. First, DBNs can learn hierarchical representations of input data, allowing them to extract and learn more complex features than traditional networks. This makes them particularly useful for tasks such as image and speech recognition. Second, the unsupervised pre-training of DBNs allows them to learn from unlabeled data, which can be advantageous when labeled data is scarce or expensive to obtain. Third, DBNs can be fine-tuned using supervised learning to perform classification tasks, which can improve their accuracy. However, DBNs also have some limitations. One major limitation is their computational complexity, which makes them difficult to train and optimize for large datasets. Additionally, DBNs require careful tuning of hyperparameters, such as the number of hidden layers and the learning rate, which can be time-consuming and computationally expensive. Finally, the interpretability of the learned features in DBNs can be challenging, as the representations learned by each layer may be highly abstract and difficult to interpret. Overall, DBNs are a powerful tool for learning

hierarchical representations of data, but their computational complexity and interpretability challenges should be considered when applying them to real-world problems.

2.6.9. Other classification models

Gated recurrent units (GRUs): GRU networks are a subclass of RNNs that are useful for classifying sequences. They can manage long-term dependencies in sequential data better thanks to a gating mechanism that enables them to selectively update their hidden state.

Ensemble Methods: A classification algorithm of this type combines the outcomes of various models in order to increase overall accuracy. This can be accomplished using strategies like bagging, boosting, or stacking.

Elastic Net: Combining both L1 and L2 regularization, elastic net is a ridge and lasso regression hybrid. In some instances, this might bring about a balance between the two approaches and improve performance.

Generative Adversarial Networks: are subclasses of neural networks that can be used to create artificial data that can be utilized to train classifiers. The discriminator network aims to discern between actual and bogus data, while the generator network attempts to make realistic data.

Variational Autoencoders : are a different class of neural network that can be trained on simulated data to perform classification tasks.

Autoregressive Models: Autoregressive models are a form of sequence classification method that models the conditional probability of each element in a sequence given its previous elements. They operate by learning a latent representation of the data that can be utilized to produce new samples. Recurrent neural networks and Markov models are two examples of techniques that can be used for this.

2. 7. Conclusion

Data mining and classification models are powerful tools that enable organizations to make informed decisions and gain valuable insights from large and complex datasets. By using various data mining techniques and algorithms, such as clustering, decision trees, and neural networks, businesses can identify patterns, relationships, and trends within their data, which can then be used to improve operational efficiency, customer satisfaction, and profitability. Moreover, classification models help organizations to classify data into different categories based on predefined criteria, enabling them to make accurate predictions and informed decisions. This approach is particularly useful in several fields where identifying and predicting trends and patterns are crucial to success. However, it is important to note that data mining and classification models require careful planning, preparation, and interpretation to ensure accurate results. Data quality, preprocessing, and feature selection are all critical

factors that can affect the accuracy and reliability of the models. In summary, data mining and classification models offer significant advantages for businesses and organizations that need to analyze large and complex datasets. By leveraging these tools effectively, businesses can make informed decisions that drive growth and success.

Chapter 3

Virtual Screening for Activity Prediction in Drug Discovery

Chapter 3: Virtual Screening for Activity Prediction in Drug Discovery

Virtual screening plays a crucial role in drug discovery by efficiently evaluating a large number of compounds, reducing the need for costly and time-consuming experimental screenings. It aids in the identification of potential drug candidates, saving resources and enabling researchers to focus on the most promising molecules for further development.

This chapter offers a comprehensive overview of the virtual screening process, its possible applications, and its potential as a valuable tool for expediting the drug discovery process. The chapter is organized as follows: The second section presents a general vision of drug discovery including the concept of interaction between the drug and the target, the traditional process of drug development, and its challenges and limitations. In the third and fourth sections, we present the concept and goals of virtual screening respectively. Section five explains the categories of virtual screening. We discuss the kinds of molecular descriptors in section six. Section seven presents a summary of the most important works using machine learning in virtual screening for activity prediction followed by most of its challenges and limitations in the last section.

3.1. Introduction

Throughout human history, diseases have been a major threat to health and wellbeing. Scientists and medical researchers have been working to find effective treatments and cures for diseases for centuries. In the past, traditional drugs were often derived from natural sources, such as plants, animals, and minerals. However, the discovery of synthetic chemistry in the 19th century paved the way for the development of modern drug. The process of discovering new drugs is complex and time-consuming, involving several stages, including target identification, hit discovery, lead optimization, and clinical trials. During target identification, scientists identify a specific molecular target, such as a protein or enzyme that is involved in the disease process. Hit discovery involves the identification of compounds that bind to the target and have the desired biological activity. Lead optimization involves modifying the hit compounds to improve their efficacy, selectivity, and safety. Finally, clinical trials involve testing the lead compounds in humans to evaluate their safety and efficacy. However, the traditional methods of drug discovery have several challenges, including the high cost and time required to synthesize and test large numbers of compounds experimentally. To overcome these challenges, computational methods, including virtual screening, have emerged as a powerful tool in drug discovery. Virtual screening involves the use of computer-based algorithms to identify compounds that are most likely to have the desired biological activity. One of its main advantages is its ability to rapidly screen large numbers of compounds in a cost-effective manner. It can also be used to identify compounds with novel mechanisms of action and to optimize lead compounds to improve their efficacy and safety.

The history of drug is a story of human ingenuity and determination to overcome diseases. The development of modern drug has led to significant improvements in health and wellbeing. However, the traditional methods of drug discovery have several challenges, and the advent of computational methods, including virtual screening, has provided new

opportunities to accelerate the drug discovery process and identify new treatments for diseases.

3.2. General vision of drug discovery

3.2.1. Drug- Target

When we become sick, the usual course of action is to take medication to aid in our recovery. However, have we ever pondered the workings behind these drugs and how they combat or eliminate illness? The answer lies in the mechanism of drug-target interaction, which explains how these drugs function.

Drug is small molecule that binds to different macromolecules (proteins, nucleic acids, receptors, enzymes, hormones, ion channels) within the human body, triggering a positive biological response. The specific macromolecule, also known as a bimolecular, whose function and activity are altered by a particular drug, is referred to as the drug target in the field of pharmacy. Drug-target binding usually involves non-covalent interactions, including hydrogen bonding, electrostatic interactions, and hydrophobic interactions.

The interaction between a drug and its target can be influenced by multiple factors, such as the chemical composition of the drug, the binding location on the target protein, and the signaling pathways triggered by the target.

Figure 3.1 represents an example of a drug –target interaction which involves three main steps. First, the molecule (drug) binds to the target protein at the active site. Second, the drug-target complex activates or inhibits the activity of the target protein, depending on the mechanism of action of the drug. Finally, the drug and its metabolites are eliminated from the body through various routes, such as renal excretion, biliary excretion, or metabolism by the gut microbiota [148].

The specificity of the drug-target interaction is an important consideration in drug development, as drugs that interact with multiple targets can lead to off-target effects and unwanted side effects. Therefore, the ideal drug should have high affinity and specificity for its target, with minimal interaction with other proteins in the body.

3.2.2. Drug development process

The process of drug development involves a series of steps that can take up to a decade or more to complete. It typically starts with preclinical research and finish with clinical trials and approval with drug regulatory agencies (Figure 3.2).

3.2.2.1. Target selection

Identifying specific molecules or proteins within a biological system that can be targeted to treat a particular disease or condition is crucial in the drug development process. The process typically involves an integration of computational and experimental methods, including gene expression analysis, functional genomics, and high-throughput screening (HTS) assays, to identify potential targets for drug intervention [149]. Once a target has been identified, it must

be validated to ensure that the disease pathogenesis is greatly influenced by its critical role and that its inhibition will lead to therapeutic benefits. The most promising targets are those that have a clear link to the disease and are amenable to modulation through small molecules, biologics, or other modalities. Additionally, it is important to consider factors such as safety, pharmacokinetics, and commercial viability when selecting a target for drug development [150]. Ultimately, the success of a drug development program depends heavily on the quality of target selection, and therefore, it is a critical area of focus for researchers and pharmaceutical companies alike.

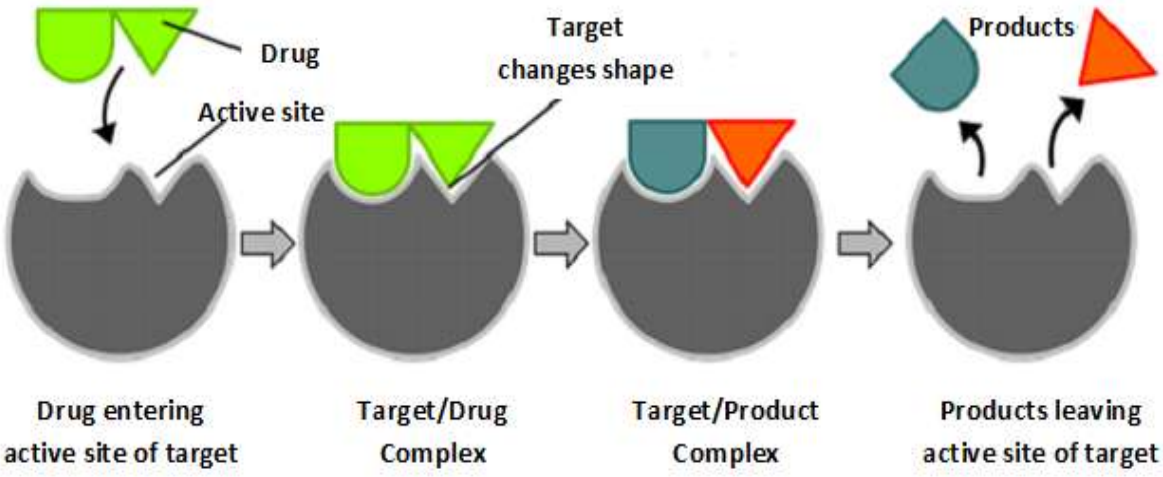


Figure 3.1. Drug-Target interaction [148].

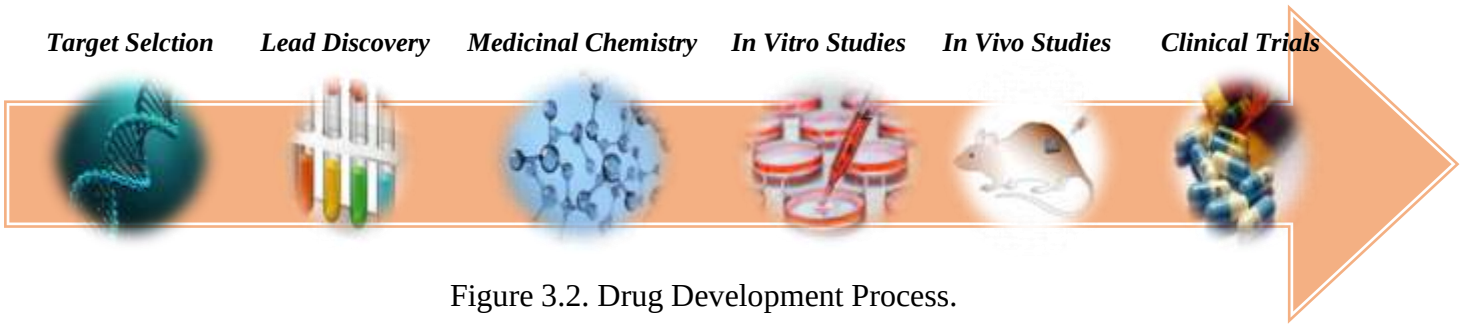


Figure 3.2. Drug Development Process.

3.2.2.2. Lead discovery

Lead discovery is the process of identifying and optimizing potential drug candidates that can modulate a specific target, which was selected in the previous step of drug development. This step involves the screening of large chemical libraries, natural product extracts, or computer-generated compounds using a variety of in vitro and in vivo assays, to identify lead compounds that show activity against the target of interest. To expedite the identification of

potential drug candidates, HTS is frequently implemented to assess large libraries of compounds against a particular target. Other methods, such as virtual screening and fragment-based drug design, are also utilized in order to discover potential lead compounds. [151]. After Identifying lead compound, it is subjected to a series of optimization steps to enhance its pharmacological properties. The optimization process includes the refinement of the compound's potency, selectivity, pharmacokinetics, and safety by performing various activities like analog synthesis and testing, pharmacophore optimization, and computational modeling to predict binding affinity with the target. [152]. The primary aim of lead discovery is to find one or more lead compounds that show promising drug-like properties, such as efficacy, safety, and target selectivity, and can be further developed into clinical candidates.

3.2.2.3. Medicinal chemistry

Medicinal chemistry is a critical step in the drug development process that involves the design and synthesis of compounds that have the potential to become drug candidates. In this step, chemists use their knowledge of the target protein structure and the lead compound identified in the lead discovery step to design and synthesize new molecules that can improve the efficacy, selectivity, pharmacokinetics, and safety of the lead compound [153]. Medicinal chemists employ a variety of strategies to optimize lead compounds, including structure-activity relationship (SAR) analysis, molecular modeling, and synthetic chemistry. SAR analysis involves the synthesis and testing of a series of compounds that have structural variations from the lead compound to determine the effects of these changes on the compound's biological activity. Molecular modeling is used to predict the binding interactions between the compound and the target protein, while synthetic chemistry is used to optimize the compound's properties, such as its solubility, stability, and bioavailability [154]. The goal of medicinal chemistry is to identify one or more lead compounds that show favorable drug-like properties, such as potency, selectivity, pharmacokinetics, and safety, and can be further developed into clinical candidates.

3.2.2.4. In vitro studies

One crucial stage of the drug development process involves conducting in vitro studies, which entails testing prospective drug candidates in a laboratory environment through various methods such as cell-based or biochemical assays. These studies provide valuable information about the compound's biological activity, selectivity, and mechanism of action, as well as its potential toxicity and pharmacokinetics [155]. In vitro studies typically involve the use of isolated cells or tissues, which are exposed to the drug candidate at varying concentrations to determine its effects on cellular function. Biochemical assays are also used to study the binding of the drug candidate to its target protein, as well as to measure the compound's enzymatic activity and selectivity. In vitro studies valuable information regarding the effectiveness and toxicity of the drug candidate can be obtained, and can guide further optimization of the compound's properties. It is crucial to acknowledge that the ability of in vitro studies to predict the effects of a drug candidate in vivo is limited. Therefore, further

testing in animal models and clinical trials is imperative to evaluate the drug candidate's safety and efficacy.

3.2.2.5. In vivo studies

The drug development process includes in vivo studies that are essential to assess the pharmacokinetics, efficacy, and safety of potential drug candidates, by testing them on live animals. In vivo studies are typically conducted in multiple animal species and involve the administration of the drug candidate via various routes, such as oral, intravenous, or subcutaneous, to determine its pharmacokinetic profile and potential toxicity. Animal models that are chosen for in vivo studies should mimic the human disease state as closely as possible and provide relevant data on the effectiveness and safety of the potential drug compound. In vivo studies also provide valuable information about the drug candidate's metabolism, distribution, and excretion, which are important factors in determining its potential clinical use. In addition, in vivo studies can help to identify potential toxicities and adverse effects of the drug candidate, which can guide further optimization of the compound's properties [155]. The findings obtained from in vivo studies play a significant role in guiding the design and execution of clinical trials, which represent the subsequent phase in the drug development pipeline.

3.2.2.6. Clinical trials

The last and most significant step in the drug development process, clinical trials involve testing potential medication candidates for safety and efficacy in humans. These trials are normally carried out in stages, each of which addresses a different aspect of the drug's effectiveness and safety. A small number of healthy volunteers participate in phase 1 trials to evaluate the safety, tolerability, and pharmacokinetics of medication candidates. Phase 2 trials assess the drug candidate's safety and efficacy in a bigger, more regulated sample of patients with the target condition. Phase 3 studies validate the efficacy and safety of the medication candidate in a larger patient population. Phase 4 trials are lastly carried out following drug approval to assess long-term safety and efficacy [156]. Clinical studies must have regulatory agency permission and are subject to stringent regulation by organizations like the US Food and Drug Administration (FDA). Clinical trial outcomes determine whether a medicine is approved for use in commerce and provide prescribing information [155].

3.2.3. Challenges and limitations of traditional process of the development of drug

The traditional process of drug development faces several challenges and limitations, including:

• **High failure rate**: The success rate of drug development is low, with many compounds failing to make it past preclinical testing or clinical trials. This is due to a variety of factors, including poor pharmacokinetics, toxicities, lack of efficacy, and unexpected side effects.

• **Time-consuming and expensive**: The drug development process can take several years or even decades, with costs ranging from millions to billions of dollars. This makes it challenging for smaller companies or academic researchers to bring new drugs to market.

• **Animal models may not accurately reflect human physiology**: Animal models are often used to assess the safety and efficacy of drugs before clinical trials. However, these models may not accurately reflect the complexities of human physiology, which can lead to misleading results.

• **Limited access to patients**: Clinical trials require access to large numbers of patients with the relevant disease or condition, which can be challenging to obtain. This is particularly true for rare diseases or conditions that affect specific populations.

The traditional process of drug development faces several challenges and limitations that can make it difficult to bring new drugs to market. New approaches and technologies are needed to improve the efficiency and success rate of drug development and to overcome these challenges.

3.3. Virtual screening concept

A popular computer technique in drug discovery is virtual screening, which identifies prospective therapeutic candidates from enormous libraries of compounds. Through the use of computer algorithms and molecular modeling methods, a series of chemicals are tested against a target protein or receptor in virtual screening to determine their affinity for the target. This approach can significantly reduce the time and cost required for drug discovery, as it allows researchers to focus on a smaller subset of compounds that have a higher likelihood of being effective in laboratory or clinical trials. [157, 158].

3.4. Goals of virtual screening

The primary goals of VS in drug discovery are:

• **Identify potential drug candidates**: To find possible drug candidates that can interact with a target protein or biological target and modify its function, huge chemical libraries are subjected to virtual screening. Finding tiny compounds with high binding affinities to the target protein is the aim in order to further improve them for medication development.

• **Prioritize compounds for further development:** Virtual screening can help prioritize compounds for further development by identifying the most promising drug candidates with high activity and favorable pharmacokinetic properties. This can save time and resources by focusing efforts on the most promising compounds.

• **Reduce the cost and time of drug discovery:** By lowering the time and expense needed for experimental screening of huge chemical libraries, virtual screening might hasten the drug discovery process. Faster and more effective drug discovery may result from this, thereby helping patients.

• **Improve the success rate of drug discovery:** By finding potential drug candidates with high activity and advantageous pharmacokinetic features, virtual screening can raise the

possibility of producing safe and effective medications, which can increase the success rate of drug discovery.

The goals of virtual screening in drug discovery are to identify and prioritize potential drug candidates, reduce the cost and time of drug discovery, and improve the success rate of drug discovery.

3.5. Virtual screening categories

There are several categories of virtual screening (VS) techniques (Figure 3.3), including:

3.5.1. Ligand-based VS

There are several types of ligand-based VS methods used in drug discovery, including:

3.5.1.1. Pharmacophore-based VS

This strategy entails building a pharmacophore model, which is a collection of characteristics that depicts the structural and chemical prerequisites for a molecule to attach to the target. These characteristics could include aromatic rings, functional groups, and donors and acceptors for hydrogen bonds. To find compounds that fit the model's characteristics, vast databases of molecules are virtually screened against the pharmacophore model. The most promising candidates are then chosen for further development after the detected hits have been further improved using a variety of computational and experimental techniques. This method's primary drawbacks are its reliance on the pharmacophore model's correctness and the requirement for experimental assays for validation. Pharmacophore-based VS has been effectively used in several studies for drug development, including the finding of potential inhibitors for a wide range of targets such kinases, G-protein coupled receptors, and enzymes [159, 160].

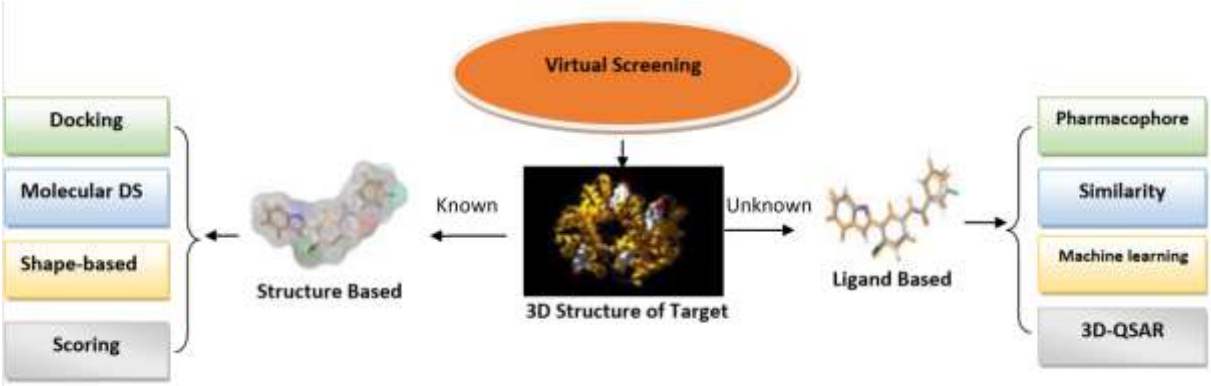


Figure 3.3. VS categories.

3.5.1.2. Similarity-based VS

Computational methods such as similarity-based VS are commonly used in drug discovery to identify potential drug candidates. This approach involves comparing the molecular structures or properties of known active compounds with large databases of molecules to identify hits with similar features. Molecular descriptors such as fingerprints, physicochemical properties,

or 3D pharmacophore features are commonly used to calculate similarity scores. Promising hits are further refined through computational and experimental techniques to select the most suitable candidates for further development. Although this method has been successfully applied in various studies to identify inhibitors for different targets such as kinases, nuclear receptors, and proteases, it has limitations, including dependence on the quality and availability of reference compounds and potential for false positives [161].

3.5.1.3. ML-based VS

ML-based VS is a computer method used in the drug development process that employs algorithms to anticipate the activity or characteristics of compounds based on available information. It's a method for finding new hits in huge chemical databases by building prediction models out of a set of known active and inactive compounds. The models can be trained on many molecular descriptors, including 3D structures, physicochemical properties, and molecular fingerprints. Once trained, they can be used to rank compounds based on their expected activity and predict the activity of new compounds. This method's main benefit is its capacity for handling huge datasets and predicting chemical activity in the absence of experimental data. However, it is constrained by the requirement for good training data and the risk of overfitting. VS based on ML has been used to find drugs for a variety of targets, including kinases, G-protein coupled receptors, and ion channels [162-164].

3.5.1.4. 3D-QSAR

A computational method called 3D-QSAR (Three-Dimensional Quantitative Structure-Activity Relationship) is used in drug discovery to link a molecule's biological activity to its three-dimensional structure. Using this technique, a predictive model is built that connects a molecule's physicochemical and steric characteristics to its activity. A set of molecules with known activities were utilized to create the model, which can be used to forecast the activity of new molecules. The 3D-QSAR model is frequently built on a number of molecular alignments that allow for the comparison of molecules' electrostatic, steric, and hydrophobic properties. This method's capacity to link a molecule's 3D structure to its activity allows for insights into the molecular connections that fuel activity, which is one of its key benefits. However, 3D-QSAR is reliant on alignment precision and necessitates superior structural data. The identification of possible inhibitors for numerous targets, including kinases, G-protein coupled receptors, and enzymes, has been accomplished using 3D-QSAR in several successful investigations in drug development [165].

3.5.2. Structure-based VS

3.5.2.1. Docking-based screening

Docking-based VS is a computational method used in drug development to uncover potential therapeutic candidates by simulating the interaction between target proteins and small compounds. Utilizing molecular docking software, which determines a tiny molecule's binding affinity and mode based on their respective 3D structures, allows for the achievement of this goal. With the help of this method, massive databases of small compounds are

screened against the target protein to find potential hits that might create stable interactions with the protein. The most promising candidates for further development are then chosen from the refined set of detected hits using a variety of computational and experimental methodologies. The main advantage of this approach is that it can forecast the binding mode and affinities of small molecules, which can direct the development of lead compound optimization. The main difficulties with this method are its reliance on the precision of the protein structure and the requirement for experimental validation. In drug development, docking-based VS has been effectively used in a number of investigations, including the finding of potential inhibitors for a wide range of targets including kinases, proteases, and receptors [166, 167].

3.5.2.2. Molecular dynamics simulations

Molecular dynamics (MD) simulations-based VS is a computational approach utilized in drug discovery for identifying potential drug candidates. It uses the results of DM simulations to predict the activity and binding affinity of compounds. This method employs molecular mechanics force fields and numerical integration algorithms to simulate the interactions between the target protein and the ligand over time. During the simulation, the system is subjected to specific initial conditions, such as pressure and temperature, and is allowed to evolve following classical mechanics laws. MD simulations-based VS can provide crucial information about the thermodynamics, energetics, and dynamics of protein-ligand complexes, as well as the stability and formation of such complexes. In drug discovery, this approach can help optimize lead compounds, discover new binding sites, and predict the effects of mutations on protein-ligand interactions. The significant advantage of this method is its ability to capture the complex conformational changes and flexibility of the protein-ligand complex, which are essential for their function. However, the computational cost can be a major limitation, as MD simulations can require a significant amount of computing power and time. Various studies have employed MD simulations-based VS in drug discovery successfully, including the identification of potential inhibitors for various targets like receptors, ion channels, and enzymes [168].

3.5.2.3. Shape-based VS

The molecular size and form of tiny compounds are used in the computer process known as "shape-based VS" to find prospective therapeutic candidates. This approach compares small molecule structures from a database to a 3D model of the target location based on the molecules' shape and electrostatic characteristics. The commonly used ROCS algorithm measures the degree of similarity between the query molecule and a database of compounds using a shape-tanimoto score. Drugs that target different molecular targets, such as kinases, G protein-coupled receptors, and nuclear hormone receptors, have been discovered via shape-based VS. Despite being effective, this method's main drawbacks are that it depends on the precision of target site information and the caliber of molecular databases [169, 170].

3.5.2.4. Free energy calculations (Scoring)

In VS, free energy calculations, commonly referred to as scoring functions, are a popular strategy for predicting the binding affinities of small compounds to target proteins. These techniques are predicated on the idea that the difference between a ligand's free energy in solution and free energy when bound to a protein can be used to determine a ligand's binding affinity. Various techniques, including molecular mechanics force fields and molecular dynamics simulations, can be used to calculate free energy. However, the force field's quality and the simulation method employed have a significant impact on how accurate these methods are. AutoDock, Glide, and GOLD are just a few of the scoring functions that have been developed for VS; each has its own advantages and disadvantages. Free energy estimates have been effectively used in numerous studies to find new inhibitors for a variety of targets, including kinases, proteases, and G protein-coupled receptors [171, 172].

3.5.3. Hybrid VS

This method combines both Structure-based and Ligand-based VS approaches to improve the accuracy of VS. In hybrid VS, the chemical features of known ligands are used to guide the docking simulations, which can increase the chances of identifying novel compounds.

3.6. Molecular description for VS

The molecular description is a critical component of VS, as it involves the representation of both the ligand and protein structures in a format that can be used for computational analysis. This includes the types of bonds and other physical and chemical properties that are relevant to binding interactions. By employing accurate and efficient molecular descriptions, VS can rapidly screen millions of compounds and identify those with the highest potential for binding to the target protein. Molecular descriptors can be divided into three main categories according to the dimension of the descriptor 1D, 2D, or 3D. [173-177] are some references which help us to write the following part concerning the molecular descriptions:

3.6.1. 1D Descriptor

A 1D descriptor, also known as a scalar descriptor, is a type of molecular descriptor that captures a single numerical value for a molecule. These descriptors provide the important physical and chemical properties of a molecule that can impact its interactions with a target protein. Those descriptors are often used as a first step in VS to quickly compare and filter large sets of molecules based on their properties. They are typically calculated based on the molecular structure of a molecule and can be used to identify molecules with similar properties or to prioritize molecules for further analysis. There are several 1D molecular descriptors that are commonly used in drug discovery and VS, but some of the most important ones include:

• **Molecular weight:** The sum of the atomic weights of all atoms in a molecule is often used to estimate the size and complexity of a molecule.

• **LogP (lipophilicity):** A measure of a molecule's ability to dissolve in lipid (fat) environments, which is important for predicting a molecule's permeability through biological membranes.

• **Number of hydrogen bond donors:** The number of hydrogen atoms in a molecule that can form a hydrogen bond with another atom or molecule is important for predicting a molecule's ability to form interactions with biological targets.

• **Number of hydrogen bond acceptors:** The number of atoms in a molecule that can accept a hydrogen bond, which is also important for predicting a molecule's ability to form interactions with biological targets.

• **Topological polar surface area:** The surface area of a molecule that is occupied by polar atoms or groups is important for predicting a molecule's ability to interact with biological targets.

• **Number of rotatable bonds:** The number of bonds in a molecule that can rotate around their axis, which is important for predicting a molecule's flexibility and ability to adapt to different target conformations.

• **Number of aromatic rings:** The number of planar rings in a molecule with a delocalized electron system, which is important for predicting a molecule's ability to form pi-stacking interactions with biological targets.

• **Number of heavy atoms:** The number of atoms in a molecule that are not hydrogen, which is important for predicting a molecule's size and complexity.

• **Number of chiral centers:** The number of atoms in a molecule with four different substituents that can exist in two mirror-image forms (enantiomers), which is important for predicting a molecule's potential stereochemistry and pharmacological activity.

• **Number of atoms with positive/negative charges:** The number of atoms in a molecule with a net positive or negative charge, which is important for predicting a molecule's electrostatic interactions with biological targets.

3.6.2. 2D Descriptor

A 2D descriptor is a type of molecular descriptor that characterizes a molecule based on its 2D structure. These descriptors capture information about the connectivity of atoms in a molecule, such as the presence and types of chemical bonds between atoms, the arrangement of atoms in rings, and the presence of functional groups. 2D descriptors are often calculated by software algorithms that analyze the molecular structure of a compound, and they can be used to compare and classify molecules based on their structural features. The most common 2D descriptors are mentioned below:

• **SMILES (Simplified Molecular Input Line Entry System)**: This is a string representation of the molecular structure that uses a specific syntax to encode atom types, bond types, and connectivity. For example, SMILES format of caffeine ($C_8H_{10}N_4O_2$) is as follows: ‘CN1C=NC2=C1C(=O)N(C(=O)N2C)C’ .

• **InChI (International Chemical Identifier)** : This is another string representation of the molecular structure that encodes connectivity, stereochemistry, and other molecular features in a standardized way. InChI format of cafeine is as follows: ‘1S/C8H10N4O2/c1-10-4-9-6-5(10)7(13)12(3)8(14)11(6)2/h4H, 1-3H3’.

• **Fingerprints**: These are binary strings that encode the presence or absence of specific substructures or molecular properties. Some examples include Morgan fingerprints that encode the presence or absence of substructures in a molecule (caffeine =‘0100000010’), MACCS keys that also encode the presence or absence of predefined substructures in a molecule (caffeine = ‘0100000001’), and 2D pharmacophore fingerprint that are typically generated based on the 2D coordinates of a molecule's atoms … etc.

• **Image- based descriptors:** represent the 2D structure of a molecule as an image, typically a bitmapped image. The pixels in the image correspond to different regions of the molecule, such as individual atoms or functional groups, and the color or intensity of each pixel represents a specific property, such as the electrostatic potential or the presence of a certain chemical group.

• **Graph- based descriptors:** represent the molecular structure as a graph, where nodes in the graph represent atoms and edges represent chemical bonds between the atoms. Different properties of the molecule, such as the type of atom or the bond order, can be encoded as attributes of the nodes and edges. Examples of graph-based descriptors include the molecular graph descriptor and the extended connectivity fingerprint.

3.6.3.3D Descriptor

A 3D descriptor is a type of molecular descriptor that encodes information about the three-dimensional structure of a molecule. Unlike 1D or 2D descriptors, which provide information about the molecular formula or two-dimensional connectivity, respectively, 3D descriptors capture information about the spatial arrangement of atoms in a molecule. There are several types of 3D descriptors that can be used to represent the 3D structure of a molecule. Here are some examples:

• **Molecular shape descriptors**: These describe the overall shape and size of a molecule, and are useful for predicting properties such as solubility, bioavailability, and molecular docking. Examples of molecular shape descriptors include volume, surface area, and shape index. The molecular shape can be computed using methods such as molecular mechanics, quantum mechanics, or Monte Carlo simulations.

• **Electrostatic potential descriptors:** These describe the distribution of electrostatic charges on the surface of a molecule, which can affect its interactions with other molecules. Examples of electrostatic potential descriptors include electrostatic potential surface area, surface charge density, and partial charges. These descriptors can be computed using quantum mechanics or molecular mechanics methods.

• **Surface properties descriptors:** These describe the surface properties of a molecule, such as hydrophobicity, polarizability, and hydrogen bonding capacity. Examples of surface properties descriptors include solvent-accessible surface area, topological polar surface area, and contact surface area. These descriptors can be computed using molecular mechanics or quantum mechanics methods.

• **Quantum mechanical descriptors:** These are based on quantum mechanical calculations of the electronic structure of a molecule, such as the HOMO-LUMO energy gap or the dipole moment. These descriptors can be used to predict properties such as reactivity, stability, and spectroscopic properties.

3.7. VS applications in drug discovery

By using VS, drug discovery researchers can save time and resources by prioritizing the most promising compounds for further experimental testing, ultimately leading to more efficient and effective drug development processes. It involves the prediction of a compound's activity against a specific biological target, which is critical in determining its potential to become a drug candidate. VS has several applications in drug discovery including:

3.7.1. Activity prediction

Activity prediction in drug discovery is the process of estimating the biological activity of compounds based on their molecular characteristics. It plays a crucial role in identifying potential drug candidates and guiding the development of effective therapeutics. One of its primary objectives is the discovery or prediction of patterns that reveal the relationship between molecular features and compound activity. By analyzing large datasets of compounds with known activity values, researchers strive to uncover recurring patterns or trends that provide valuable insights into the essential molecular characteristics associated with activity. These patterns can include specific chemical groups, structural motifs, or physicochemical properties that consistently correlate with high or low activity. By leveraging these discovered patterns, activity prediction models can be developed to estimate the activity of novel compounds, facilitating the efficient selection of promising candidates for further experimental evaluation. The objective of discovering or predicting patterns in activity prediction is to enhance the efficiency and success of drug discovery efforts by identifying compounds with desired biological activity and guiding the design of optimized therapeutics. Predicting compound activity is a fundamental and multifaceted aspect of drug discovery. It encompasses a diverse array of techniques aimed at estimating the activity of compounds. These predictions span various areas, such as binding affinity, inhibition, and ADMET characteristics.

A. Binding affinity prediction

Binding affinity refers to the extent of interaction between a small chemical and a target protein. It is essential for drug discovery to identify compounds that bind to the target protein with a high degree of affinity because this is typically associated with greater biological activity. Computational methods are used to estimate binding affinity by simulating the interaction between the small molecule and the target protein. One of the most widely used computational tools is molecular docking, which predicts the binding conformation and binding energy of a small molecule in the binding site of a target protein. Other computational methods, such as molecular dynamics simulations and free energy calculations, can also be used to predict binding affinity [178].

B. Inhibition prediction

A tiny molecule's power to inhibit is its capacity to stop a target protein's activity. Inhibition is frequently employed in drug discovery as a therapeutic approach to treat disorders linked to excessively active or dysfunctional proteins. By modelling the small molecule's binding and interaction with the target protein, computational approaches are utilized to predict inhibition. Molecular docking is a popular technique for anticipating inhibition, much like the prediction of binding affinity. Inhibition can also be predicted using other techniques, such as molecular dynamics simulations and QSAR analyses. [179].

C. ADMET prediction

The acronym ADMET stands for absorption, distribution, metabolism, excretion, and toxicity, all of which are crucial pharmacokinetic and safety characteristics of a drug candidate. To locate substances with good pharmacokinetic profiles and lower the possibility of toxicity problems, it is essential in drug discovery to anticipate ADMET features. By modeling the physicochemical and biological processes that control the absorption, distribution, metabolism, excretion, and toxicity of a substance, computational approaches are used to estimate ADMET parameters. The ADMET features of a substance can be predicted using ligand-based techniques, such as QSAR analysis and pharmacophore modeling, based on its chemical makeup. Based on the interaction between the chemical and the biological target or transporters, structure-based approaches, such as molecular docking and molecular dynamics simulations, can be utilized to estimate ADMET characteristics. [180].

3.7.2. Hit identification

Identifying possible lead compounds that could be turned into medications is a vital step in the drug development process known as hit identification. Using in vitro and in silico methods, a large number of compounds are screened in this step to identify those that have an action against a particular biological target. Hit identification is frequently an iterative process that may involve VS. Using either technique ligand-based and structure-based VS, it is possible to screen sizable compound libraries for hits that may be improved by lead optimization and preclinical testing to find the most promising drug candidates for further research [181].

3.7.3. Lead optimization

Lead optimization is a crucial step in the drug development process that tries to improve the potency, selectivity, pharmacokinetic characteristics, and safety of successful compounds. To find hit compound derivatives and analogs with desirable features, VS can be used. The scaffold can be changed or functional groups can be added to the hit compound's chemical structure to increase activity or selectivity. The activity, pharmacokinetics, and safety of the analogs and derivatives can then be evaluated utilizing a variety of in vitro and in vivo experiments. To ensure that the chemical can reach the target in enough concentrations and maintain those concentrations over time, lead optimization aims to balance the desired pharmacological qualities with drug-like features including solubility, stability, and

bioavailability. The features of analogs and derivatives, such as their binding affinity, potency, and selectivity, can be predicted by VS. This can help to direct their synthesis and evaluation and decrease the number of compounds that need to be manufactured and assessed in vitro and in vivo. To sum up, lead optimization is an important step in the drug discovery process that uses VS to find high-quality drug candidates by modifying and optimizing hit molecules [182].

3.7.4. Drug repurposing

Drug repurposing, also known as drug repositioning or drug reprofiling, is the act of coming up with new applications for medicines that have previously received clinical use approval or are at an advanced stage of clinical development. Given that many of these medications' safety and pharmacokinetic features are already known, this strategy may be an effective and affordable way to find new therapeutic uses for them. One method for screening a large number of compounds and finding medications that may have activity against new targets, thereby opening the door to new therapeutic applications, is VS. Based on the three-dimensional structures of the drug and the target, computational approaches are used to forecast the binding affinity of a drug to a particular target. Through VS, medicines with the potential to bind to a target with high affinity and selectivity are found, and these medications can subsequently undergo additional in vitro and in vivo testing [183]. The use of VS for drug repurposing has a number of advantages. First, it can be used to identify new therapeutic uses for medications that have failed in their initial indications due to efficacy or safety concerns. Second, it can be used to find novel drug combinations that might increase efficacy and have a synergistic effect. Finally, repurposing currently available medications rather than creating wholly new substances can speed up drug development and save costs. VS for medication repurposing do have certain drawbacks, though. Finding the right targets for a particular illness or condition is one of the biggest hurdles, and doing so requires a thorough understanding of the biology of the illness as well as the underlying mechanisms of action. In order to confirm that medications are safe and effective for their new application, additional safety and effectiveness studies may be necessary. A detailed understanding of disease biology and careful evaluation of safety and efficacy issues are both necessary for the success of the medication repurposing using VS methodology, which is a promising method for finding new therapeutic applications for existing pharmaceuticals .

3.8. Summary of most important works using ML in virtual Screening for activity prediction

Statistics says that the most common methods of ML used in VS are: SVM, RF, Neural Networks, GB, and NB because they have demonstrated good performance in predicting the activity of compounds and identifying potential drug candidates. In the fact, the choice of ML method for VS depends on several factors such as the nature of the dataset, the complexity of the problem, and the computational resources available. Different ML methods have different strengths and weaknesses, and the best method for a particular problem will depend on these factors. For example, if the dataset is high-dimensional and noisy, RFs or GB may be more

suitable than other methods because they can handle noisy data well and can work with a large number of input features. On the other hand, if the dataset is structured and has a clear separation between classes, SVM may be more appropriate, NB is the best in cases where the dataset is imbalanced or the features are sparse. Similarly, neural networks are a powerful method that can learn complex representations of the data, but it requires a large amount of data and computational resources. Therefore, it may be more suitable for large-scale VS projects that have access to significant computational resources. The less commonly used ML methods for VS and activity prediction are: DTs, KNN, LR for several reasons: DT have limitations in handling high-dimensional datasets and noisy data. In VS, the datasets can be quite complex, with a large number of input features and noise, which may limit the effectiveness of DTs. While KNN is simple and flexible, it is computationally expensive, and the choice of distance metric can have a significant impact on the performance of the algorithm. This can limit its effectiveness in large-scale VS projects. LRis also a simple and interpretable algorithm, but it has limitations in capturing non-linear relationships between features and target variables. In VS, there may be complex relationships between chemical features and target activity, which can limit the effectiveness of linear regression. Overall, the choice of ML method for VS and activity prediction should be based on careful consideration of the data and problem at hand, as well as the strengths and weaknesses of different methods. While some methods may be less commonly used, they may still have certain advantages in specific cases.

3.8.1. RF- based methods

Due to its effectiveness in handling missing data and ability to reveal the relative value of each parameter, RF is frequently employed in VS. A vast collection of chemical descriptors (features) are collected from the compounds and utilized to train an RF model in RF-based approaches for activity prediction. Based on the correlation between the molecular properties and the target activity, the model gains the ability to spot patterns in the data and anticipate outcomes. The technique has been used in numerous drug development efforts and has been proven to be successful in predicting the activity of compounds for a variety of protein targets. To increase the precision of the predictions, RF-based algorithms can potentially be integrated with other machine-learning strategies.

Svetnik et al. assessed the effectiveness of the RF algorithm as a classification and regression tool for chemical categorization and QSAR modeling in their study [184]. A broad collection of 61 datasets, including information on biological activity and molecular characteristics, was employed by the scientists. With an average classification accuracy of 84.6% and an average correlation coefficient of 0.76 for regression, the RF algorithm did well in both classification and regression tests. Additionally, the RF algorithm beat SVM and PLS as well as other classification and regression techniques. The study emphasized the RF algorithm's potential for use in drug discovery and development and showed how useful it is as a tool for chemical classification and QSAR modeling. In order to facilitate the discovery and development of new drugs, Song and Jiang's study [185] intended to create a better ensemble learning approach for inferring the relationship between chemicals and pathways. The KEGG database

contains chemical and pathway data for more than 8,000 substances and 314 pathways. The authors used this data to create an ensemble learning strategy that merged different classifiers to improve prediction accuracy. The method outperformed the NB and DT classification methods in terms of prediction accuracy, with a prediction accuracy of 95.36%. The study also discovered a number of substances that were strongly linked to particular pathways, revealing prospective medication targets and demonstrating the utility of ML techniques in drug development. Through the use of the RF algorithm, Singh et al.'s study [186] intended to create a QSAR-based model for differentiating between EGFR inhibitors and non-inhibitors. The study generated 3D molecular descriptors for each chemical using a dataset of 1,435 substances. The model's 87.35% accuracy, 86.61% sensitivity, and 87.99% specificity show that ML techniques can be used to predict the activity of chemicals. The research also revealed crucial molecular descriptors that may be used to distinguish between EGFR inhibitors and non-inhibitors, shedding light on the molecular characteristics of these drugs. In particular, the work shows how ML techniques can be used to anticipate whether certain chemicals will act as EGFR inhibitors or not. Mistry et al.'s goal in [187] was to create a computational toxicology vehicle prediction approach employing RF and DT models. The study generated 1777 2D molecular descriptors for each of the 509 substances in the dataset. The goal was to create a model that could precisely forecast the chances that various transporters would deliver a substance into cells. Both the RF and DT models worked admirably, with the accuracy of the RF model being better (83.8% vs. 80.5%) than that of the DT model. The study discovered crucial molecular descriptors that affected drug transport, including polarizability and electrical characteristics. Overall, the study showed the potential of computational toxicology's use of ML algorithms to forecast the transport of chemicals and emphasized the significance of creating precise prediction models in drug discovery and development. The goal of the study suggested by Shangjie et al. [188] was to undertake VS for COX-2 inhibitors using the RF algorithm and feature selection. In the study, 18,066 2D molecular descriptors were computed for each of the 3876 compounds with known COX-2 inhibitory activity. A variety of molecular descriptors were applied, including a novel set of pharmacophore-based descriptors, 2D fragment descriptors, and fingerprint-based descriptors. The study discovered that feature selection, with a precision of 92.6% and an area under the curve (AUC) of 0.989, could greatly increase the predictive model's accuracy. Additionally, a list of top-ranking chemicals with significant predicted COX-2 inhibitory action was found in the analysis, some of which had experimental validation. Overall, the work showed the promise of RF and feature selection in VS for drug development and emphasized the significance of using the right molecular descriptors for precise predictive modeling. In [189], the authors present a methodology for predicting the anticancer drug activity based on chemical attributes and minimum genetic information. The NCI60 cancer cell line panel provided information on pharmacological activity against 60 human cancer cell lines, which the researchers used. Gene expression profiles were employed as chemical descriptors, and molecular fingerprints and physicochemical attributes were used as genomic descriptors. On a portion of the data, RF models were trained, and on the remaining data, they were validated. AUC of 0.93 was attained by the top-performing model, which indicates good accuracy in

predicting medication action against cancer cells. This study reveals the possibility of predicting medication effectiveness using just limited genomic data and chemical qualities, which could assist in the creation of more efficient cancer treatments. The goal of [190] was to create a functional RF (FRF) model for predicting dose-response in drug screening investigations. The National Institutes of Health's Molecular Libraries Program, which contained data on the action of drugs against diverse biological targets, was utilised by the researchers. The data contained details on the drugs' chemical compositions and dose-response correlations. In order to predict the dose-response relationship of chemicals based on their chemical characteristics, researchers employed FRF modeling. The model's effectiveness was assessed using a variety of metrics, including mean squared error and coefficient of determination (R2). With an R2 of 0.80 for the test set, the results demonstrated that the FRF model outperformed other widely used models including LR and support vector regression. The study indicates the potential of FRF modeling for enhancing the precision of dose-response predictions in drug screening investigations, which may facilitate the creation of new medications for a variety of conditions. Using Kullback-Leibler divergence, Ahn et al. [191] have created an RF model for forecasting drug-target interactions. The ChEMBL database, which has data on the chemical structures of compounds as well as their bioactivity against diverse targets, was utilised by the researchers. The target proteins were represented using sequence-derived descriptors, and the data was preprocessed to produce molecular fingerprints as descriptors for the compounds. In order to predict the likelihood of drug-target interactions, the RF model was trained on the preprocessed data and evaluated using a variety of performance measures, including the area under the receiver operating characteristic curve (AUC-ROC) and precision-recall curve (AUC-PRC). The AUC-ROC and AUC-PRC values for the suggested model were 0.911 and 0.575, respectively, outperforming other cutting-edge models, according to the results. The paper indicates the possibility for predicting drug-target interactions using RFmodels with Kullback-Leibler divergence, which could help with drug development and discovery.

RF is a powerful ML algorithm in VS for predicting the activity of potential drug candidates. However, there are several limitations of RF that can impact its effectiveness in this context:

• **Overfitting**: This can be particularly challenging when trying to find drugs because the objective is to find compounds that have good activity in actual experiments.

• **Limited feature selection**: RF may not be the most effective method to find the most pertinent features in drug discovery because there may be thousands of potential traits to take into account.

• **Unbalanced data:** RF models may also have difficulty with unbalanced data sets, where the ratio of positive to negative examples is lopsided. As a result, the ability to predict the activity of uncommon or novel compounds that are poorly represented in the training data may suffer.

3.8.2. SVM- based methods

SVM are often used in VS because they can handle large datasets with high-dimensional feature spaces and are effective at classifying compounds based on their chemical features.

Using molecular descriptors, Jan M. Kriegl et al. [192] created an SVM model in 2005 to predict human cytochrome P450 inhibition. A dataset of 165 substances described by chemical descriptors and known to have cytochrome P450 inhibitory activity was employed by the researchers. To predict the cytochrome P450 inhibitory activity of a given chemical based on its molecular descriptors, the SVM model was trained on the dataset. Utilizing several performance indicators, including accuracy, sensitivity, and specificity, the model was assessed. The findings demonstrated that the SVM model was 90.5% accurate in predicting the action of cytochrome P450 inhibition. The study shows the ability of molecular descriptors and SVM models for predicting the inhibition of cytochrome 450, which is a crucial element in drug metabolism and toxicity. By combining fragmental descriptors from QSAR, the authors of [193] have presented a study that aims to build an SVM model for predicting the assignment of organic compounds to pharmacological classes. The researchers used a dataset of 165 organic compounds with known pharmacological activity that was represented using fragmental descriptors in order to achieve this. With the use of this dataset, the SVM model was trained to identify a compound's pharmacological group based on its fragmentary characteristics. The model's performance was assessed using a variety of criteria, including accuracy, sensitivity, and specificity. According to the study, the SVM model was 92% accurate in predicting the pharmacological group of the substances. The results emphasize the potential of fragmental descriptors and SVM models in foretelling the pharmacological actions of organic compounds, which could be useful in drug research and development efforts. In order to find possible Abl kinase inhibitors using SVM, the study of [194] sought to virtual screen vast chemical libraries. The SVM models were trained and validated using a dataset of known Abl inhibitors and non-inhibitors. In order to represent the chemicals, molecular fingerprints were used, and numerous SVM models with various kernel functions and parameters were created. Next, a large chemical library was screened to find possible Abl inhibitors using the top-performing SVM model. To determine their binding affinity to Abl kinase, molecular docking studies were used to further investigate the top-ranking drugs. In order to identify possible Abl kinase inhibitors from huge compound libraries, the study showed the potential of employing SVM models and VS. This could help researchers find new cancer treatments. In [195], Xiaowu Dong et al. proposed a QSAR model for SVM-based activity prediction of Akt/protein kinase B (PKB) inhibitors. The study's 260 molecular descriptors, which include topological, geometrical, constitutional, and electrostatic characteristics, were applied to a dataset of 134 Akt/PKB inhibitors. In order to predict the activity of a certain Akt/PKB inhibitor based on its chemical characteristics, the SVM model was trained on the dataset. Utilizing several performance indicators, including accuracy, sensitivity, and specificity, the model was assessed. The outcomes demonstrated that the SVM model had prediction accuracy for the activity of the inhibitors of 84.7%. The work shows the potential of QSAR models and SVMs for predicting the action of Akt/PKB inhibitors, which could help in the development of new medications for the treatment of cancer and other disorders. Creating a combinatorial SVM model for VS of selective multitarget kinase inhibitors was the goal of the study in [196]. The 647 kinase inhibitors in the dataset that the researchers used were represented using 2D and 3D molecular descriptors.

To predict the selectivity score of a specific inhibitor against a group of 20 kinases, the SVM model was trained on the dataset. Several performance indicators, including accuracy, sensitivity, specificity, and AUC, were used to assess the model's performance. The findings demonstrated that the combinatorial SVM model successfully predicted the inhibitors' selectivity score with an AUC of 0.89. In order to virtually screen for selective multitarget kinase inhibitors, the work shows the potential of combinatorial SVM models. This could help in drug development and discovery. In order to identify target-selective molecules, the work of [197] intended to build SVM-based ranking algorithms. To create molecular descriptors for each chemical, the researchers examined a dataset of substances with documented activity against a variety of targets. Using a variety of performance criteria, the SVM models were assessed after being trained to rank chemicals in accordance with their expected activity against a particular target of interest. The findings demonstrated the high accuracy with which target-selective molecules may be identified using SVM-based ranking techniques, highlighting the promise of this strategy for drug discovery. By applying problem-specific measures, the authors in [198] has created a linear SVM model for learning SAR. The study team made advantage of a big collection of chemical compounds with well-documented biological target functions. With the use of molecular characteristics including topological polar surface area, logP, and molecular weight, the chemical compounds were represented. Using the dataset, the SVM model was trained to forecast, based on a compound's chemical descriptors, its activity against a target. Accuracy, sensitivity, and specificity were only a few of the measures used to assess the model. According to the findings, the SVM model predicted the compounds' activity against the targets with a high degree of accuracy of 87%. The study shows the potential of a large-scale SAR analysis employing a linear SVM and problem-specific metrics, which could help with drug discovery and development. To virtually screen vast chemical libraries for selective multi-target serotonin reuptake inhibitors (SRIs), the researchers in [199] developed a combinatorial SVM technique. The study's dataset included 719 non-SRIs and 91 known SRIs, and it was represented by molecular descriptors such physicochemical and topological characteristics. To predict the selective SRI activity of a given chemical based on its descriptors, the SVM model was trained on the dataset. Several performance criteria, including accuracy, sensitivity, and specificity, were used to assess the model's performance. According to the findings, the combinatorial SVM technique had an accuracy rate of 87.9% when it came to foretelling selective multi-target SRIs. The paper shows the potential of combinatorial SVM models for VS of picky multi-target SRIs from big chemical libraries, which could help in medication development. The goal of Poorinmohammad N et al.'s study [200] was to computationally predict anti-HIV-1 peptides and assess their in vitro activity. A dataset of 57 peptides generated from HIV-1 P24 and information on their anti-HIV-1 efficacy was employed by the researchers. Different molecular descriptors, including electronic, topological, and physicochemical descriptors, were used to represent the peptides. Based on the molecular characteristics of the peptides, a predictive model for their anti-HIV-1 efficacy was created using the SVM algorithm. Several performance indicators, including accuracy, sensitivity, specificity, and AUC-ROC, were used to assess the model's performance. The findings revealed that the SVM model had an

excellent prediction performance with an accuracy of 83.5% and an AUC-ROC value of 0.842. The work shows the potential of using chemical descriptors and SVM-based computational models for predicting the anti-HIV-1 activity of peptides. According to [201], topological autocorrelation descriptors and SVM are used to identify high-affinity protease-inhibitor complexes on a wide scale. Over 100,000 protein-ligand complexes with established binding affinities were used by the researchers. The 2D molecular structures of the inhibitors were used to construct the topological autocorrelation descriptors, and the dataset was used to train SVM models to forecast the binding affinities of novel complexes. Different performance criteria, such as accuracy and AUC-ROC, were used to assess the models. The findings indicated the SVM models' potential for predicting the binding affinities of new protease-inhibitor complexes by demonstrating their excellent accuracy and AUC-ROC values. According to various chemical characteristics, including MACCS keys and Morgan fingerprints, the authors of [202] employed a dataset of 997 substances with documented inhibitory efficacy against LF. The accuracy, sensitivity, and specificity of the SVM models were calculated along with other performance metrics after they had been trained and refined using a grid search algorithm. The findings indicated that the SVM models had a high predictive accuracy of 96% and sensitivity of 97%, pointing to the promise of SVM models as a tool for the creation of anthrax LF inhibitors. The goal of the study by Onay et al. [203] was to categorize medications that influence the neurological system as either approved or withdrawn using ML techniques. The researchers analyzed a dataset of 165 medications, 85 of which were withdrawn and 80 of which were approved, and then used the ToxPrint method to calculate molecular descriptors. SVM models for classification were created using the descriptors as a starting point. AUC of the receiver operating characteristic (ROC) curve, accuracy, sensitivity, and specificity were some of the measures used to assess the models' performance. According to the findings, SVM models created using ToxPrint descriptors classified withdrawn and approved medications that affect the neurological system with good accuracy (up to 98.2%). The research shows how ToxPrint descriptors and SVM models may be used to classify drugs, which could help in medication development and toxicity prediction. Predicting drug target groups using several drug networks is the main goal of the paper in [204]. To build numerous drug networks, the scientists employed a variety of data sources, such as drug-target interactions, protein-protein interactions, and drug-drug interactions. They subsequently trained an SVM model for predicting the drug target group using these networks as features. Accuracy, sensitivity, and specificity are just a few of the evaluation criteria that were used to assess the SVM model's performance. The outcomes demonstrated that the SVM model performed well, with an accuracy of 84.5%. The paper shows how SVM models and different drug networks can be used to forecast drug target populations, which could help in medication development and discovery. The goal of the work of [205] is to create a ML-based Multiple QSAR technique to forecast a compound's activity against HIV/HCV coinfection. A collection of 682 compounds with anti-HIV and anti-HCV activity served as the basis for this study's results. In this work, molecular parameters like molecular weight, polar surface area, and the number of rotatable bonds were used as descriptors along with molecular fingerprints. Three ML algorithms SVM, RF, and

Multiple LR are used to build the Multiple QSAR model. Cross-validation and a third-party test set are used to validate the model. With an accuracy of 0.89 and an AUC of 0.92 on the external test set, the results demonstrate that the Multiple QSAR model utilizing SVM performs at the highest level. The study demonstrates the multiple QSAR method's potential for predicting a compound's activity against an HIV/HCV coinfection.

SVM is a commonly used ML algorithm in VS for activity prediction. However, SVM also has some limitations that can impact its effectiveness in this context:

• **Limited scalability**: SVM can be computationally expensive and may not scale well to very large datasets. This can be a challenge in VS, where large numbers of molecules need to be evaluated.

• **Slection of kernel function**: SVM uses a kernel function to move the input data into a higher-dimensional feature space where a hyperplane can divide it up. Finding the best kernel function, however, can be challenging. The choice of kernel function can have a major impact on the model's performance.

• **Sensitivity to parameter tuning**: The regularization parameter and the kernel parameter must both be tuned when using SVM. If these variables are not carefully selected, the model may either overfit or underfit the data, which would result in subpar performance.

• **Unbalanced data**: Similar to RF models, SVM models can be challenged by unbalanced data sets in which the proportion of positive and negative examples is lopsided.

3.8.3. NB- based methods

NB have been used in VS because they can model complex relationships between features and provide probabilistic predictions about the activity of compounds.

To help with lead identification for Alzheimer's disease treatment, Jiansong Fang et al.[206] have developed in silico algorithms for forecasting the actions of Butyrylcholinesterase inhibitors. To distinguish BuChEIs from non-inhibitors, 1870 structural descriptors (1235 from ADRIANA Code, 334 from MOE, and 301 from Discovery studio) were used to build SVM models and NB models. Correlation analysis and stepwise variable selection were used to identify activity-related descriptors and structural fingerprint descriptors were added to increase prediction power. Cross-validation, test set validation with 1001 compounds, and external test set validation with 317 different chemicals were all used to assess the models. The Matthews correlation coefficient for the top two models was 0.9551 for the test set and 0.9550 for the external test set, respectively. An internal dataset of 3601 compounds was virtually screened using the models, and 30 compounds were chosen for bioactivity testing. The considerable BuChE inhibitory activity of ten out of the thirty compounds allowed for the first time-ever discovery of three novel scaffolds as BuChE inhibitors. This research shows that it is possible to find new lead compounds for BuChE inhibition in the treatment of Alzheimer's disease by predicting the bioactivities of ligands. In order to determine if a drug is an inhibitor or non-inhibitor of the mammalian target of rapamycin (mTOR), a key regulator of cell growth, proliferation, metabolism, and angiogenesis, Ling Wang et al. [207]

developed in silico models based on multi-scaffolds. utilizing 1264 different chemicals classified as mTOR inhibitors and non-inhibitors, the researchers built combinatorial classification models based on physicochemical descriptors, fingerprints, and atom center fragments utilizing recursive partitioning and NB techniques. There were 253 models created in total, and the top models had prediction accuracies of more than 90% for both the training and external test sets. The objective of the study in [208] was to create classification models for the VS of neuraminidase (NA) inhibitors and non-inhibitors, which are essential for the development of anti-influenza drugs. With various ratios of active-to-inactive compounds in the training set and various chemical descriptors, the researchers generated SVM and NB models of NA inhibitors and non-inhibitors. The NA inhibitory actions of 15,600 chemicals in their internal database were predicted using four models with sensitivity or Matthews correlation coefficients greater than 0.9. To evaluate the NA inhibitory characteristics of 60 sample drugs in vitro, the results of the four best models were merged. A substantial contributor to drug-induced organ toxicity and the etiology of numerous disorders, drug-induced mitochondrial toxicity was the focus of the authors' work in [209], who sought to create a prediction model for it using an NB classifier. The study also evaluated how well the recursive partitioning classifier prediction model performed. The internal 5-fold cross-validation of the training set and external test sets revealed that the NBclassifier performed the best in terms of overall prediction accuracy, with average values of 95 0.6% and 81 1.1%, respectively. In addition to certain exemplary substructures of toxicants created by ECFP_6 fingerprints, the study found four significant molecular descriptors. According to the scientists, analyzing mitochondrial toxicity can be aided by the established NB prediction model, and the knowledge gleaned from this assessment can help medicinal chemists find new drugs and improve lead formulation. The goal of the study [210] was to construct a VS pipeline for locating possible VEGFR2 inhibitors from FDA-approved medications. VEGFR2 is a key pharmaceutical target for the creation of anti-angiogenic medicines for the treatment of cancer. To analyze 1841 FDA-approved medications, the researchers merged ligand-based NB models with structure-based molecular docking. High Matthews correlation values of 0.966 and 0.951 for the test set and external validation set, respectively, were produced by the best-validated NB model. Flubendazole, rilpivirine, and papaverine all had inhibitory effects in the VEGFR2 kinase test, which was used to determine the biological validity of nine top-ranked medicines. Three FDA-approved medications were identified in the study as novel VEGFR2 inhibitors, which could be helpful in designing and developing new antiangiogenic medicines for cancer therapy. The study also recommended an integrated VS pipeline. Using a Bayesian ML approach that incorporates several data sources, such as genomic, transcriptomic, and proteomic data, the work of [211] sought to find new therapeutic targets. The Cancer Genome Atlas, Gene Expression Omnibus, and the Cancer Cell Line Encyclopedia were a few of the publicly available datasets that the authors used. The input data underwent preprocessing to create gene expression profiles, which were then utilized to create a scoring measure for pharmacological targets. The outcomes demonstrated that this method was successful in identifying new therapeutic targets for a number of malignancies, including colon, lung, and breast cancer. The authors showed that this strategy performed

better in terms of precision and recall than other ML techniques, such as logistic regression and RF. The goal of the study described in [212] was to create NB models for the prediction of Vero cell cytotoxicity, which can be applied to drug discovery to find potentially harmful substances. The 402 chemicals that were tested for cytotoxicity on Vero cells made up the dataset that was used. A collection of 53 quantitative descriptors, including physicochemical characteristics, molecular weight, and 2D fingerprints, were used to describe the molecules that made up the compounds. The findings demonstrated that the NB models had a high level of predictive accuracy for Vero cell cytotoxicity, with an overall accuracy of 91.8% and an AUC-ROC of 0.89. The models could be used to select compounds for additional testing in drug discovery by being able to identify potentially harmful molecules. An in silico prediction model of ototoxicity was created by researchers in [213] in order to accurately analyze any potential cellular degeneration of the cochlear and/or vestibular systems brought on by medications. To create the model, they employed an NB classifier technique with a collection of 2612 different compounds. A collection of seven molecular descriptors thought to be crucial for ototoxicity were chosen using the genetic algorithm approach, and structural alarms for ototoxicity were discovered. For the training set and the external test set, the established model provided an overall prediction accuracy of 90.2% and 88.7%, respectively. According to the researchers, this model may be an effective computational tool for evaluating and screening chemical-induced ototoxicity in drug development and may offer recommendations for hit and lead optimization in drug design.

NB is a commonly used ML algorithm in VS for activity prediction. However, NB also has some limitations that can impact its effectiveness in this context:

• **Assumption of independence:** In drug development, features like chemical descriptors might be strongly coupled, which is frequently not the case as NB implies that all input features are independent of one another. Information loss and subpar performance may result from this.

• **Limited expressiveness:** Because of its limited expressiveness and potential inability to recognize intricate correlations between features, NB may not be able to predict the activity of molecules with sufficient accuracy.

• **Sensitivity to feature scaling:** NB is sensitive to feature scaling, and the performance of the model can be heavily impacted by the choice of scaling method used.

• **Imbalanced data:** Similar to RF and SVM, NB models may have trouble with data sets that are lopsidedly balanced between the amount of positive and negative samples.

3.8.4. GB- based methods

GB has been used in VS to estimate the activity of compounds and is effective at handling high-dimensional feature fields.

To predict the bioactivity of small compounds, the authors of [214] have created a ML model using the Extreme GB (XGBoost) algorithm. The ChEMBL database provided the data for this investigation, and the descriptors employed were based on molecular fingerprints. Several

metrics, including accuracy, precision, recall, F1 score, and ROC curve, were used to assess the XGBoost model's performance. The outcomes demonstrated that, in terms of accuracy and AUC values, the XGBoost model beat other ML models, such as RF, SVM, and NB. The study showed the XGBoost algorithm's potential for predicting the bioactivity of tiny compounds. The authors of the research [215] discuss the use of ML algorithms for predicting urinary tract toxicity brought on by substances. They used a dataset that contained details on the toxicity of 722 compounds to the urinary system. They employed a range of ML techniques, such as RF, GB, and ANNs, to predict the toxicity. The dataset was split between training and testing sets, and the effectiveness of each model was evaluated using measures including sensitivity, specificity, and accuracy. The most effective model was found to be the XGBoost method, which achieved an accuracy of 0.848 and an AUC value of 0.917. The study shows how ML techniques may be used to predict toxicity during the drug discovery process. Ping Xuan et al. proposed a study in [216] with the goal of creating a prediction model to discover potential interactions between medicines and target genes. The model was developed by the authors using GBDT techniques, and the effectiveness of the model was assessed using data from known drug-gene interactions. Both drug and gene descriptors, such as molecular fingerprints and gene ontology annotations, were included in the data used to train and test the model. With an AUC of 0.958 on the test set, the results demonstrated that the GBDT model performed better than other ML techniques. Additionally, the authors used their model to forecast probable drug-gene interactions for a number of medications and discovered several fresh drug-target interactions. Overall, the study showed that GBDT-based models can be used to foretell drug-target interactions. The goal of the work in [217] was to predict the binding affinity of protein-ligand complexes using the ML algorithm GB. The scientists determined 15 molecular characteristics for each ligand using a dataset of 1,311 protein-ligand complexes with known binding affinities. They next created a forecasting model for the binding affinities of each complex using GB. Results revealed that GB performed better than competing ML algorithms and attained a correlation coefficient between predicted and experimental binding affinities of 0.68, demonstrating outstanding performance in binding affinity prediction. The gradient-boosting method was used to identify the non-toxic antimicrobial peptide Hm-AMP2 from the leech metagenome, according to the publication of [218]. To train a GB ML model, the team employed a dataset of well-known antibacterial and non-antimicrobial peptides. The leech metagenome's putative antimicrobial peptides were then predicted using the model. After being identified as a promising antimicrobial peptide, Hm-AMP2 was created and its antibacterial activity was examined. The findings demonstrated that Hm-AMP2 is non-toxic to human cells and exhibits potent antimicrobial action against a wide range of bacteria, including both Gram-negative and Gram-positive strains. The research shows how ML techniques can be used to find new antibacterial peptides. For the precise identification of anticancer peptides, a novel technique called iACP-GE has been put out by the authors in [219]. To extract the informative characteristics and create the prediction model, the method employs GBDT and an additional tree. 983 anticancer peptides and 983 non-anticancer peptides were included in the dataset utilized for the investigation. According to the findings, iACP-GE attained accuracy rates of

98.67%, sensitivity rates of 98.87%, and specificity rates of 98.46%. The research shows that iACP-GE has the potential to be an effective technique for precisely identifying anticancer peptides. The goal of the study [220] was to classify chemical compounds that enhance longevity using the XGBoost algorithm. The scientists used a publicly accessible dataset of chemical compounds and their associated lifespan-extension effects in the nematode Caenorhabditis elegans. The collection included 42,886 individual compounds, each with 13 distinct characteristics or descriptors, such as molecular weight, logP, and polar surface area. The performance of the model was assessed using a number of criteria, including accuracy, recall, and F1-score, to categorize compounds as lifespan-extending or not. With an overall accuracy of 0.93, recall of 0.89, and F1-score of 0.93, the study demonstrated that the XGBoost algorithm successfully predicted lifespan-extension compounds. The study emphasizes the potential of employing ML techniques like XGBoost to forecast how certain chemical compounds would affect lifetime extension.

GB is a powerful ML algorithm that is commonly used in VS for activity prediction. However, there are several limitations of GB that can impact its effectiveness in this context:

• **Overfitting:** GB models can be prone to overfitting, especially if the model is too complex or if there are too many weak learners. This can result in poor performance on new, unseen data.

• **Computationally expensive:** GB can be computationally expensive, especially if the dataset is large or if there are many features to consider. This can make it difficult to scale the algorithm to large VS datasets.

• **Sensitive to hyperparameters:** GB requires tuning of several hyperparameters, such as the number of trees, the learning rate, and the depth of the trees. If these parameters are not chosen carefully, the model may overfit or underfit the data, leading to poor performance.

• **Imbalanced data:** GB models can also struggle with imbalanced data sets, where the number of active and inactive compounds is heavily skewed.

3.8.5. Neural Networks- based methods

VS using deep learning are a relatively new field of study. Due to its capacity to understand intricate non-linear correlations between variables and anticipate the activity of compounds, neural networks are being utilized more frequently in VS.

3.8.5.1. DNNs - based methods

The work by Dahl et al. [221] has used DNNs for making QSAR predictions. The authors used a large dataset of compounds with known activity values and a diverse set of molecular descriptors to train their models. The study demonstrated the effectiveness of DNNs for improving the accuracy of QSAR predictions compared to traditional ML methods. The authors also showed that their models were able to learn meaningful features from the molecular descriptors and identify important substructures associated with the activity of the compounds. Ma et al. [222] explored the use of DNNs for predicting DQSARsin chemical

compounds using a large dataset of compounds and their corresponding activity values. The authors extracted molecular descriptors such as molecular fingerprints, Dragon descriptors, and physicochemical properties from the compounds and used them as input to train and test DNN models. The results demonstrated the high accuracy of DNNs in predicting the activity of compounds, outperforming other QSAR modeling methods, with the best-performing DNN model achieving a correlation coefficient of 0.85 between predicted and observed activity values. The objective of the study by Mayr et al. [223] was to develop a deep learning approach for toxicity prediction that outperforms other ML methods in terms of accuracy and generalization ability. The authors used a large dataset of compounds with associated toxicity data and various molecular descriptors, including molecular fingerprints, MACCS keys, and Morgan fingerprints, to train their models. The authors used a DNN with multiple layers to learn the features that best predict toxicity from the molecular descriptors. The study showed that DeepTox achieved a mean average precision of 0.817, which outperformed several other ML methods, including RF, SVM, and neural network models with one or two hidden layers. The authors also demonstrated the interpretability of their models by analyzing the learned features and showing that the models were able to identify relevant substructures associated with the toxicity of the compounds. The study proposed by Aliper et al. [224] sought to examine the use of transcriptome data and deep learning to predict the pharmacological properties of medicines and medication repurposing. Gene expression patterns served as molecular descriptors to represent the medications in their models, which were trained using a sizable dataset of drug-target interactions and gene expression profiles. The study demonstrated that deep learning models outperformed other ML methods in accuracy and robustness for drug-target interaction prediction. K. Wu et al. [225] developed a deep learning approach using topology-based multitask DNNs for quantitative toxicity prediction of chemicals. They used a dataset of compounds with associated toxicity data and molecular descriptors, including topological fingerprints, to train their models. The multitask DNN outperformed other ML models, including single-task neural networks and SVMs, in predicting the toxicity of the compounds. The study indicated that multitask DNN could find pertinent aspects connected to the compounds' toxicity, and the outcomes revealed that the method significantly improved prediction accuracy when compared to the other models. The purpose of the work by [226] was to find out how well multi-task learning performed when DNNs were used to predict the molecular binding of human targets. Different multi-task and single-task networks were trained and tested using molecular interaction data from the ChEMBL database. The authors discovered that only for target sets with similar targets did multi-task learning outperform single-task learning, proving that similarity within a target set is essential for reliable multi-task learning. The authors created Multiple Partial Multi-Task learning as a result, which is appropriate for binding prediction for human drug targets. In order to increase the precision of predicting the binding residues on proteins for many types of molecules, including DNA, RNA, peptides, and carbohydrates, [227] conducted a study. A novel sequence-based approach known as MTD site was created by the authors utilizing a multiple task deep learning algorithm that simultaneously predicts binding residues/sites for various molecule types. On their separate independent test sets, they obtained precise and

reliable predictions with AUC values of 0.852, 0.836, 0.758, and 0.776 by combining four training sets for DNA, RNA, peptide, and carbohydrate-binding proteins. In comparison to other cutting-edge techniques, the outcomes were 0.52 to 6.6% better. This approach is the first to anticipate numerous molecule binding sites simultaneously utilizing a multi-task framework. The new approach might be useful for research into unstudied protein-molecule interactions. Researchers in [228] have proposed a large-scale ligand-based VS method for identifying potential inhibitors of the SARS-CoV-2 virus using DNNs. The authors collected a dataset of around 68,000 compounds from various publicly available databases and screened them against the virus using a DNN trained on known inhibitors. The neural network was trained on molecular fingerprints of compounds, which were generated using the extended connectivity fingerprints algorithm. The authors achieved promising results with the top-ranked compounds showing good binding affinities and predicted activity against the virus. The study provides a potential approach for identifying new drug candidates for treating COVID-19. Morgan fingerprints and pre-training SMILES embeddings are two alternative molecular input representations that the authors of [229] used to propose a deep learning framework for predicting toxicity. According to AUROC and balanced accuracy, the multi-task deep learning model successfully predicts toxicity for all endpoints, including clinical toxicity. In order to boost confidence and explain the model's predictions, the authors additionally use a post-hoc contrastive explanation method, which returns pertinent positive and negative properties that closely resemble well-known toxicophores. The technique reveals a bias for in vitro and in vivo experimental data over clinical outcomes in known toxicophore data, with recovery by applicable feature analysis catching more of the former than the latter. The authors assert that this is the first contrastive explanation for predictions of clinical and in vivo molecular toxicity utilizing both present and absent substructures. By transferring completely trained layers from one DNN to another DNN associated with less training data, Ruifeng Liu et al. in [330] sought to create DNN models to predict the bioactivity features of compounds. The transferability of dense layers of a pre-trained DNN to forecast related or unrelated features based on scant data was investigated by the authors. The study discovered that if the correlation r between the two assay datasets is larger, transfer learning is more effective in lowering prediction errors linked to the smaller dataset DNN predictions. For every 0.1 increase in r2 between the datasets, the mean squared prediction errors were reduced by 10 to 20%, with the transfer of the first thick layer leading to the greatest error reduction. Depending on the dataset correlation, the study hypothesizes that the training sample size could be decreased by up to ten times without a reduction in prediction accuracy.

3.8.5.2. CNN - based methods

The objective Duvenaud et al. [231] was to develop a deep learning model that could learn molecular fingerprints directly from graphs of molecules, without the need for hand-crafted descriptors. The authors used a dataset of 133,885 small molecules from the ZINC database and represented each molecule as a graph, where the atoms were nodes and the bonds were edges. They then used a CNN to learn molecular fingerprints from these graphs, which were used to predict various properties of the molecules, such as solubility and bioactivity. The

results showed that their model outperformed state-of-the-art methods based on hand-crafted descriptors, achieving a mean absolute error of 0.40 log units on the solubility prediction task. Defferrard et al. [232] applied their proposed method of CNNs on graph-structured data using localized spectral filtering to two molecular datasets: one consisting of molecules and the other of proteins. In both cases, the goal was to classify the molecules or proteins based on their properties. The study demonstrated that their approach outperformed existing methods in terms of classification accuracy for both datasets. The study introduced a CNN-based method for VS on graphs, which can predict binding affinity and activity of ligands. The model was trained on a dataset of known ligands and non-ligands. Coley CW et al. [233] have proposed a method to predict physical properties of molecules using a deep learning model called Convolutional Embedding of Attributed Molecular Graphs. The authors used a dataset of around 128,000 molecules with their corresponding molecular structures and physical properties. The molecular structures were represented as attributed molecular graphs, which contain both atom and bond attributes. The model was trained on this dataset using a supervised learning approach to predict a variety of molecular properties, including boiling point, heat capacity, and density. The CEAG model outperformed other ML models and achieved state-of-the-art performance in predicting several physical properties. AtomNet is a name of method proposed by Wallach et al. [234] using a deep CNN for predicting bioactivity of small molecules in structure-based drug discovery. The objective of the work is to develop a method that can predict the activity of small molecules on biological targets using their 3D structures, which are represented as voxel grids. The authors trained the CNN on a large dataset of 128 million bioactivity measurements on more than 1.2 million compounds, which were extracted from publicly available databases. The data were preprocessed to generate 3D voxel grids, which represent the electronic structure and the shape of the molecules. The CNN model uses a 3D convolutional layer followed by fully connected layers to make predictions. The results showed that the AtomNet model outperformed other ML methods, including traditional 2D descriptors and RF models, in predicting the bioactivity of small molecules. The study demonstrates the potential of deep learning models for drug discovery and provides a promising approach for predicting the bioactivity of small molecules. Shi et al. [235] have developed a molecular image-based CNN for predicting ADMET properties of drug molecules. The data used for model development consisted of 10,679 compounds with known ADMET properties from the ChEMBL database. The molecular images were generated from 2D molecular structure images using the extended-connectivity fingerprints descriptor and principal component analysis. The CNN was trained on 80% of the data and validated on the remaining 20%. The results showed that the CNN model achieved good performance with an average AUC of 0.91 across all five ADMET properties and outperformed several baseline models. The study demonstrated the potential of using molecular images and CNNs for predicting ADMET properties of drug molecules. The research conducted by Fernandez et al. [236] intended to use deep learning to predict the toxicity of chemical compounds purely based on their graphical representations. In order to create the CNN model "ToxNet," the scientists used a vast dataset of chemical compounds with known toxicity values. Over 1.4 million molecular structure photos that had undergone preprocessing and been converted to

grayscale for use in the CNN made up the bulk of the data utilised. The model's performance was assessed using a variety of measures, such as accuracy, AUC, and F1 score. It produced remarkable results, with an AUC of 0.99 and an accuracy of 94.9%. A dataset of novel compounds was used by the scientists to evaluate the model, and they showed that it has the potential for high-throughput drug discovery screening. A novel approach for discovering possible anticancer peptides using Graph Convolutional Networks (GCNs) has been proposed by the authors in [237]. A dataset of 1,150 peptides, 575 of which are anticancer peptides and the rest 575 not, was employed by the authors. The peptide sequences were converted into a graph format, where the amino acids are shown as nodes and the connections between them as edges. On this dataset, the GCN model was trained, and its performance was assessed using a number of metrics, including accuracy, sensitivity, specificity, and AUC. The proposed ACP-GCN method performed better than existing cutting-edge techniques for predicting anticancer peptides, with accuracy, sensitivity, specificity, and AUC values of 95.18%, 95.96%, and 94.40%, respectively. According to the study, the GCN model may be a useful tool for locating possible anticancer peptides. Using VS, the research [238] suggested a deep learning algorithm that can forecast the 3CLpro inhibitory action in SARS-CoV for unidentified compounds. The model processes chemical molecule descriptors and predicts active compounds using CNN architecture. When tested on the test set, the CNN model outperformed other ML techniques with an accuracy of 0.86, sensitivity of 0.45, specificity of 0.96, precision of 0.73, recall of 0.45, F-measure of 0.55, and ROC of 0.71. Several databases were screened using the model to find possible anti-SARS-CoV compounds, including 315 FDA-approved medications. When drug-like compounds were prioritized using Lipinski's rule of five, multiple hit molecules were discovered, including 9 anti-SARS-CoV medicines from the flavonoid class. The proposed CNN model can help with the creation of fresh anti-SARS-CoV compounds in general. Without explicit chemistry knowledge, "Chemception," a deep CNN for chemical property prediction using only 2D molecular drawings, was presented in the paper of [239]. The goal is to show that deep learning methods, in which feature engineering is mostly handled by the algorithm, may be used in computational chemistry research. The results reveal that Chemception performs comparably to expert-developed QSAR/QSPR models for predicting toxicity, activity, and solvation parameters. The data utilized to train Chemception span from 600 to 40,000 compounds. In particular, Chemception somewhat underperforms MLP and DNNs trained using ECFP fingerprints in the prediction of activity and solvation and slightly outperforms them in the prediction of toxicity. The authors in [240] have proposed using graph neural networks and CNNs to learn low-dimensional vector representations of molecular graphs and protein sequences. They demonstrate that using a neural attention mechanism can help analyze deep learning models and identify important subsequences in proteins for predicting interactions. The mechanism also provides clear visualizations even when using real-valued vector representations.

3.8.5.3. DBN- based methods

DBNs were suggested as a method by the authors of [241] for multi-target drug VS. The suggested approach uses multi-label classification as its foundation and seeks to forecast the

targets of a certain medicinal molecule. On a benchmark dataset, the technique is assessed and contrasted with other cutting-edge techniques, such as single-target and multi-label classification techniques. The experimental findings demonstrate that the proposed strategy performs better than the alternatives in terms of accuracy and productivity. The proposed strategy, according to the authors' analysis, is a promising one for multi-target drug VS and can hasten the drug discovery process. In [242], the authors have proposed a new approach using DBNs for hybrid fingerprint features. Two approaches for combining fingerprints are explored: initial combining and latter combining. The experiments show that the best way to combine fingerprints for DBN architecture is initial combining. Six protein target classes were used for the experiments, and the results showed an improvement in VS performance using the proposed approach. In [243], Fahimeh Ghasemi et al. proposed a technique to deal with the difficulties in high-throughput screening drug design, such as the vast number of descriptors and suitable parameter initialization to prevent over-fitting. To initialize DNNs for predicting the biological activity of compounds, the researchers used DBNs. In order to examine the model performance, the study employed Kaggle datasets with over 70,000 compounds and 15 targets. With a mean and variance of squared correlation of 0.618 and 0.485, respectively, the suggested model with improved parameter initialization beat the DNN model, according to the results. A DBN was used in the study [244] to create a druglikeness classification model for small compounds and authorized medications using data from the ZINC database. As data characteristics, different binary fingerprints were used, including the Macc 166-bit, PubChem 881-bit, and Morgan 2048-bit. The model was developed using an unsupervised pre-training stage. The accuracy, precision, and recall of the model were 97%, 96%, and 99%, respectively, using Macc characteristics, indicating strong performance and generalizability. The algorithm might also identify tiny compounds that were not medicines but met bioavailability requirements as possible drugs in the future. The model has potential for use in drug development as a drug filter. In [245], Long Yu published a deep learning model for categorizing compounds as inhibitors or non-inhibitors of the CYP450 1A2 enzyme, which metabolizes approximately 90% of therapeutic medicines in the human liver. A multi-tiered DBN that was trained using a dataset of more than 13,000 compounds that was obtained from PubChem forms the basis of the model. With the help of 139 2D and 53 3D descriptors of the compounds, the model automatically extracts various levels of distributed representation via unsupervised learning. SVM and ANN shallow ML models, as well as other feature combinations, are used to compare the DBN model's performance. The experimental findings demonstrated that the DBN model performed superiorly to the other models in terms of prediction ability, and that it was able to obtain the best forecast accuracy when incorporating both 2D and 3D features. To predict drug-target interactions in VS by utilizing DBNs, Aman Shakya et al. [246] have created a framework. The objective is to find ligands that are not dockable in order to narrow the search space. A logistic regression layer is stacked as an output layer after DBN is used to extract high-level features from 2D chemical substructures stored in fingerprint format. The results of this pre-training phase are utilized to initialize the parameters for fine-tuning after the DBN model has been trained in a greedy layer-wise unsupervised manner. The results of the studies demonstrate that the proposed

DBN model doubles throughput while predicting drug and target interactions with accuracy of about 90%. Drug-target combinations can be conveniently categorized using this method, which also simplifies the process of separating dockable from non-dockable ligands.

3.8.5.4. RNNs- based methods

The use of deep learning and deep architectures in chemoinformatics, specifically for forecasting drug-like compounds' water solubility, has been covered in the study in [247]. The authors outline deep learning techniques and explain how recursive neural network methods might be used to solve this issue. They devised strategies to take into account an ensemble of recursive neural networks associated with all potential vertex-centered acyclic orientations of the molecular graph in order to address the issue that molecules are typically described by undirected cyclic graphs while recursive approaches typically use directed acyclic graphs. Using four benchmark datasets, the performance of deep learning methods is assessed and contrasted with that of other cutting-edge approaches, with the results demonstrating that deep learning methods perform on par with or better than other methods. The authors also offer a web-based tool called AquaSol for estimating the solubility of drug-like compounds in water, which is accessible through the ChemDB portal. Using the fact that a single molecule can have several Simplified Molecular SMILES strings, the authors of [248] have described a method for data augmentation of a molecular QSAR dataset. The dataset was expanded and utilized to train an LTSM cell-based neural network by employing numerous SMILES strings to represent the same molecule. When compared to a model created using just one canonical SMILES string per molecule, the network's performance performed better on a test set with a higher correlation coefficient and a lower root mean square error. The method also performed well during the prediction phase, and a further improvement was seen by averaging the predictions for each molecule of the listed SMILES. RNNs have been investigated by Marwin H et al. in [249], who employed computational techniques to create new compounds with high affinity for biological targets. The authors demonstrate how generative models for molecular structures may be created using RNNs and how the properties of the resulting molecules are comparable to those used to train the model. The model was able to replicate a sizable portion of test molecules created by medicinal chemists against Staphylococcus aureus and Plasmodium falciparum by fine-tuning it with tiny sets of molecules known to be active against a specific biological target. The approach can provide vast sets of new compounds for drug development when combined with a scoring system. PharmaNet is a ML method that the authors of [250] have developed to predict the activity of drugs against certain cell receptors in huge databases. To create a fingerprint chemical image and determine the molecule's target, the algorithm uses a SMILES representation of the molecule, a convolutional encoder, and an RNN. PharmaNet outperformed earlier approaches, scoring 97.7% on the ROC-AUC and 65.5% on the Normalized Average Precision curve, according to the researchers' analysis of PharmaNet's performance. By locating possible inhibitors of human farnesyl pyrophosphate synthase (FPPS) and suggesting one candidate, CHEMBL2007613, as a potential antiviral due to its participation in the PCDH17 pathway, the algorithm was shown effective. The major protease (Mpro) of the SARS-CoV-2 is a promising target for drug development due to its

crucial role in viral replication and absence in humans. The study of [251] has developed a deep learning platform for de novo design of putative inhibitors of this enzyme. A general chemistry-based generative model was trained and validated as part of the process, and the model was then adjusted for the chemical space of SARS-CoV-Mpro inhibitors and trained as a classifier for bioactivity prediction via transfer learning. The refined model produced a variety of distinct, legitimate, and innovative structures, including novel scaffolds that may be used to investigate new chemical series. On an external test set, the classification model outperformed the baseline area under the precision-recall curve and the open-source model Chemprop. Nine of the top-20 predicted hits were found using molecular docking to exhibit binding poses and interactions similar to those of experimentally verified inhibitors. A deep learning application that is utilized to find SARS-CoV-2 important enzyme inhibitors in a variety of diseases, plants, and medications was discussed in the publication in [252]. This application's deep learning model is built on the LSTM network, which can learn and recall long-term dependencies. The dataset utilized in the training procedure consists of a variety of sources that are rich in the thirteen major inhibitors of the coronavirus-2, with each entry annotated with the inhibitors that it specifically contains. The effectiveness of the model is assessed using a variety of metrics, including accuracy, precision, recall, and F1-score. The findings demonstrate that the model has an overall accuracy of 87.4% in detecting the presence of important SARS-CoV-2 inhibitors in different sources. By locating potential sources of important inhibitors in foods, plants, and medications, this work illustrates the potential of deep learning models, in particular LSTM, to assist in drug discovery.

Deep learning, including CNNs, RNNs, and DBNs, has shown great promise in VS for activity prediction. However, there are several limitations and challenges associated with deep learning in this context:

• **Data availability and quality:** To function well, deep learning models need a lot of high-quality training data. However, because experimental testing is expensive and time-consuming, it might be difficult to gather big and diverse datasets in the field of drug discovery.

• **Interpretability:** Deep learning models are often seen as "black box" models, making it difficult to interpret the underlying logic of the model and understand why it is making certain predictions. This can be particularly problematic in the field of drug discovery, where understanding the reasons behind a predicted activity is crucial for selecting promising candidates for further testing.

• **Computationally intensive:** Deep learning models can be computationally expensive, especially if they are large and complex. This can make it difficult to apply these models to large VS datasets or for high-throughput screening.

• **Generalization:** Deep learning models trained on one set of molecules may not generalize well to other molecules or targets, which can limit their usefulness in VS.

• **Limited experimental validation:** Although deep learning models can provide accurate predictions in silico, experimental validation is still necessary to confirm the predicted activities. This can be time-consuming and expensive, particularly for large VS campaigns.

• **Limited availability of pre-trained models**: Unlike traditional ML models, deep learning models require large amounts of computing power and time to train. This can make it difficult for researchers with limited resources to train their own models, and pre-trained models may not be available or suitable for their specific needs.

3. 9. Challenges and limitations of VS

VS has become a widely used computational technique in drug discovery due to its potential to identify potential drug candidates with increased speed and reduced costs. However, VS also has several challenges and limitations. The accuracy of VS methods is limited by the accuracy of the computational models used to predict the interactions between drug candidates and target proteins. The size and diversity of the database of compounds being screened also affect the effectiveness of VS. Additionally, the dynamic nature of target proteins in the human body, including conformational changes when interacting with ligands, presents a challenge for computational models to predict the appropriate protein conformations accurately. As a result, VS should be used in combination with experimental techniques to validate the results and overcome these limitations.

3. 10. Conclusion

Virtual screening, which employs the application of computational methods to uncover new drug candidates, has evolved as an effective and economical way of drug discovery. It enable researchers to quickly screen a large number of compounds and choose those with the most promise for further development by utilizing a wide range of data sources and sophisticated algorithms. VS are not a replacement for conventional experimental techniques, but it has emerged as a crucial tool in the early phases of drug development. It is positioned to play an increasingly significant role in the development of new medications for a wide range of ailments as a result of the expanding availability of chemical and biological data as well as the quick development of computational approaches. VS, in general, represents a huge advancement in drug development, enabling researchers to more rapidly and precisely identify viable drug candidates and ultimately bring new medicines to market more successfully.

Chapter 4

Proposed Approaches Based on Deep Learning Using 2DPF for Activity Prediction

Chapter 4: Proposed Approaches Based on Deep Learning Using 2DPF for Activity Prediction

Activity prediction is a crucial aspect of drug discovery, where the goal is to identify small molecules that can bind to a specific target and modulate its function. It involves not only predicting the activity of molecules but also discovering patterns that reveal the underlying relationship between molecular features and activity. Predicting the activity of molecules can help researchers prioritize the most promising compounds for further investigation. This chapter focuses on our main contribution within the framework of our thesis, which is the proposal of two approaches for virtual screening to predict the activity of molecules with a specific receptor using 2D pharmacophore fingerprint. The chapter is organized as follows: In the second section, we provide an explanation of the type of data utilized in our research and we emphasize the importance of this data in molecular description and its relevance to the virtual screening process. Furthermore, we discuss the selected target and provide comprehensive motivations behind its choice, elucidating the significance of this particular receptor in our study. In the third section, we present our initial approach for VS, in which we meticulously outline each step of the approach, including the generation of fingerprints, the selection of relevant features, the proposed predictive models, and the comprehensive results obtained from experimental evaluations. Section four introduces the second approach for virtual screening, which bears a resemblance to the first approach but incorporates notable differences in its execution. We present an explanation of this approach, highlighting the modifications and unique aspects that distinguish it from the first approach.

4.1. Introduction

In recent years, the development of new drugs has become increasingly important as the number of diseases and illnesses continues to rise. The process of discovering new drugs involves identifying compounds that have the potential to interact with a specific target in the body, such as a protein or enzyme, and studying their effects on the target. However, traditional methods of drug discovery can be time-consuming and expensive, which has led to an increased interest in the use of ML algorithms to accelerate the drug discovery process. The use of deep learning algorithms in drug discovery is becoming increasingly popular due to their ability to analyze large datasets and extract relevant features automatically. However, there are still some challenges that need to be addressed in VS, such as the need for efficient and accurate methods to predict the biological activity of chemical compounds.

Despite the level of scientific advancement we have reached, cancer remains one of the deadliest diseases of our day. It is a complicated category of illnesses characterized by the body's cells growing and spreading abnormally. In order to replace worn-out or damaged cells, the body's cells typically divide and multiply in a planned and controlled manner. However, in cancer, cells can begin to divide and grow out of control, which can result in tumor development or the invasion of surrounding organs and tissues. Through a process known as metastasis, cancer can potentially travel to different regions of the body through the lymphatic or blood systems. One characteristic of cancer is the dysregulation of essential proteins involved in controlling cell division and the cell cycle. CDK1 (Cyclin-dependent kinase 1) is a critical protein involved in ensuring that cells divide and replicate their DNA accurately and faithfully, and it is essential for the proper progression of the cell cycle.

Dysregulation of CDK1 activity can lead to uncontrolled cell proliferation, which is a hallmark of cancer. In our works, we have proposed novel virtual screening procedures in drug discovery that incorporate deep learning algorithms to predict the biological activity of chemical compounds on the CDK1 receptor.

Central to our approach is the utilization of 2D pharmacophore fingerprints, which capture the essential features and patterns within the molecules. These fingerprints are generated using our proposed distance ranges, carefully selected to capture the relevant patterns related to the target receptor. By leveraging deep learning models, we learn from these fingerprints and effectively classify chemical compounds based on their activity. Our research aims to contribute to the field by demonstrating the potential of deep learning in utilizing the proposed pharmacophore fingerprints as powerful descriptors for accurate activity predictions. Overall, the objective of our work is to contribute to the growing body of research on the application of deep learning in drug discovery and to demonstrate its potential in transforming. They have the potential to accelerate the development of new drugs by enabling faster and more accurate predictions of the biological activity and molecular properties of chemical compounds. This can significantly reduce the time and cost associated with drug discovery and lead to the development of more effective drugs.

4.2. Data and Concepts

4.2.1. Why CDK1 receptor?

Research and technological advancements have significantly improved cancer prevention, detection, and treatment, which is a serious public health concern. But considerable work needs to be done in order to comprehend the underlying causes of cancer and create better, more individualized treatments. Numerous cancers, including breast, lung, and colorectal cancers, have been discovered to have elevated levels of CDK1. Aside from that, CDK1 mutations or those of its regulators have been linked to the emergence and spread of cancer. Consequently, CDK1 is a promising target for cancer therapy, and it is the topic of active study at the moment. Understanding the role of CDK1 in cancer is critical for developing new and more effective treatments for this devastating disease.

4.2.1.1. CDK1 in cells

CDK1 is a member of the Cyclin-dependent kinase family of proteins that is located in the cytoplasm and nucleus of eukaryotic cells (Figure 4.1), which play a critical role in controlling the cell cycle (Figure 4.2).

The series of activities that take place in a cell that prepare it for division and duplication is known as the cell cycle. It consists of four main phases where each phase is characterized by different cellular processes and is regulated by various proteins and signaling pathways. Here's a brief overview of each phase [253,254]:

• **G1 phase:** The cell expands in size and produces the proteins and organelles required for DNA replication during this phase. The cell also monitors its internal and external

environment to ensure that it is ready to proceed to the next phase. This phase is regulated by various proteins, including Cyclin D and Cyclin E.

• **S phase:** During this phase, the cell synthesizes a copy of its DNA. Each chromosome is replicated, resulting in two identical copies of each chromosome. This phase is regulated by Cyclin A.

• **G2 phase:** The cell goes through a number of procedures at this stage to get ready for cell division. The cell enters the G2 phase after DNA replication is finished in the S phase to check for errors and make sure the cell is prepared for mitosis. The cell examines its DNA for damage during the G2 phase and corrects any faults. The cell also produces the proteins and organelles necessary for cell division, including the spindle fibers that aid in chromosomal segregation during mitosis. Cyclin A and Cyclin B control this phase.

• **M phase:** The M phase, which includes both mitosis and cytokinesis, is the last stage of the cell cycle. The duplicated chromosomes are divided into two identical sets during mitosis, and these sets are then distributed to the daughter cells. The method by which a cell physically separates into two daughter cells is called cytokinesis. Cyclin B and Cyclin A are two of the proteins that control this phase.

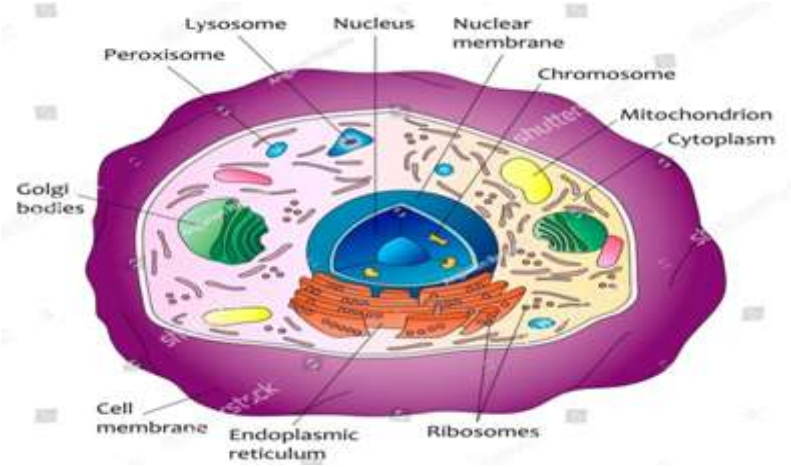


Figure 4.1. Human cell [255].

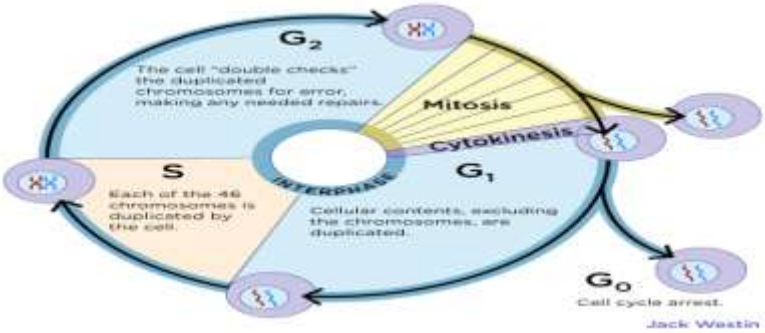


Figure 4.2. Cell cycle [256].

CDK1 is specifically involved in the regulation of the G2 phase of the cell cycle, which is the stage just before cells divide. During the G2 phase, CDK1 is activated by a protein called Cyclin B, which binds to CDK1 and activates its kinase activity. This allows CDK1 to phosphorylate various downstream targets involved in the cell cycle progression, including other Cyclin-dependent kinases and the proteins involved in DNA replication and cell division. Dysregulation of CDK1 activity can lead to abnormal cell division and proliferation, which is a hallmark of many cancers.

4.2.1.2. CDK1 contribute to cancer

There are several ways in which CDK1 can contribute to cancer [257-258]:

• **Overexpression of CDK1**: Overexpression of CDK1 means that there is an abnormally high level of CDK1 protein in cells. In normal cells, CDK1 levels are tightly regulated and controlled. However, in cancer cells, there can be an overproduction of CDK1 due to mutations or dysregulation of the genes that control its expression. This overproduction of CDK1 can result in an increase in its activity, leading to uncontrolled cell proliferation and tumor growth. Therefore, overexpression of CDK1 is a potential target for cancer therapy. Overexpression of CDK1 has been observed in several types of cancer, including breast, lung, and ovarian cancer. Increased CDK1 activity can promote cell cycle progression and cell proliferation, which can contribute to tumor growth.

• **Mutations in CDK1 or its regulators:** Mutations in CDK1 or its regulators refer to changes or alterations in the genetic sequence of CDK1 or its regulatory proteins. These mutations can lead to abnormal activation or inhibition of CDK1 activity, which can disrupt the normal cell cycle and contribute to cancer. For example, mutations in the CDK inhibitor gene p16INK4a can lead to increased CDK1 activity and cell proliferation. Similarly, mutations in other regulatory proteins, such as cyclins, can also lead to abnormal CDK1 activity and contribute to cancer. These mutations can be inherited or acquired over time due to environmental factors, such as exposure to radiation or chemicals. Understanding the genetic mutations that contribute to CDK1 dysregulation is important for developing targeted therapies to treat cancer.

• **Abnormal expression of CDK1 regulators:** Abnormal expression of CDK1 regulators refers to changes in the levels of proteins that regulate the activity of CDK1. The activity of CDK1 is tightly controlled by a number of different proteins, including cyclins, CDK inhibitors, and other regulatory proteins. Abnormal expression of these regulators can disrupt the normal cell cycle and contribute to the development of cancer.

4.2.1.3. CDK1 in drug discovery

Because of its critical role in the cell cycle and its involvement in cancer, CDK1 has become an important target for drug discovery. Inhibiting CDK1 activity can potentially stop the proliferation of cancer cells and induce cell death. Several small molecule inhibitors of CDK1 have been developed and tested in clinical trials for the treatment of various cancers. Predicting the activity of small molecules against CDK1 is a crucial step in identifying

potential drug candidates for CDK1 inhibition. Targeting CDK1 is a promising strategy for cancer therapy, and ongoing research is focused on developing more effective and selective inhibitors that can be used in combination with other therapies for better outcomes.

Deep learning models can be trained on large datasets of chemical and biological data to predict the activity of small molecules against protein targets like CDK1. These models can take into account the chemical structure of the molecules, as well as their interactions with the target protein and other relevant biological factors. The availability of large datasets of CDK1-related data, including crystal structures and biochemical assay data, makes it a suitable target for deep learning-based activity prediction models. In summary, CDK1 is a critical protein involved in the regulation of the cell cycle, with dysregulation of its activity being linked to various diseases, including cancer. CDK1 is an important target for drug discovery, and predicting the activity of small molecules against CDK1 is crucial in identifying potential drug candidates. Deep learning-based activity prediction models are a powerful tool for predicting the activity of small molecules against CDK1 and other protein targets.

4.2.1.4. Few researches related to the use of ML methods for activity prediction of molecules with CDK1

In fact, there are few research papers on CDK1 that have used ML methods to predict activity, which is all the more reason to do our research on it. In the literature we find the following:

In 2019, a conference paper of Isabella Mendolia et al. [259] presented a VS procedure using CNN to classify candidate compounds based on their biological activity on CDK1. The proposed approach uses molecular fingerprints to describe molecules. 1D and 2D CNNs are trained on different types of molecular fingerprints.

Zhou et al. [260] have suggested a new technique for predicting alterations in CDK1 in 2020. The procedure entails enhancing various aspects of CT images related to the tumor using a novel technique for tumor image enhancement, and modeling the prediction of a CDK1 gene mutation using a deep neural network with a multi-strategy fusion loss function to deal with asymmetric and difficult samples. Comparative studies show that the suggested approach enhances classifier accuracy and surpasses alternative loss functions in terms of AUC.

4.2.2. 2D Structure of molecules

The 2D structure of molecules is an important concept to understand in the field of chemistry and drug discovery. It is a representation of the molecule on a flat surface. A flat surface refers to a two-dimensional plane, which can be represented using two coordinates. In chemistry, a 2D structure is typically represented using a two-dimensional drawing, where the atoms in the molecule are depicted as symbols and the bonds between the atoms are represented by lines (Figure 4.3). Each atom in the 2D structure is assigned a position on the plane using two coordinates, usually denoted as x and y. The way in which atoms are bonded together determines the shape of the molecule and the specific values of these coordinates can be used to determine the relative positions of the atoms in the molecule and the angles

between the bonds. By analyzing the 2D structure in this way, chemists and other researchers can gain insights into the chemical properties of the molecule and its potential interactions with other molecules. The 2D structure typically shows the bonds between the atoms in the molecule, as well as any functional groups or other features that are relevant to its chemical properties and interactions with other molecules. Table 4.1 shows the coordinate of ethanol's atoms in 2D space.

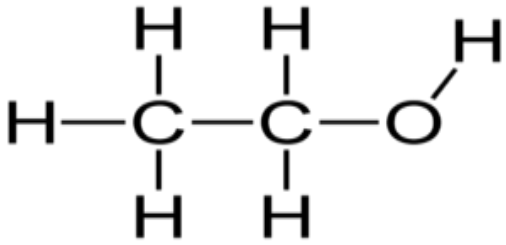


Figure 4.3. 2D Structure of ethanol (C2H6O).

Table 4.1. Coordinates of atoms in ethanol.

Aom	X	Y
O	3.7320	0.2500
C	2.8660	-0.2500
C	2.0000	0.2500
H	2.4675	-0.7249
H	3.2646	-0.7249
H	2.3100	0.7869
H	1.4631	0.5600
H	1.6900	-0.2869
H	4.2690	-0.0600

Understanding the 2D structure of molecules is critical in drug discovery. In order for a drug to be effective, it must be able to interact with specific molecules in the body, such as enzymes or receptors. The 2D structure of a drug molecule can provide important information about its potential interactions with these target molecules. The use of computational methods to predict and analyze 2D structures of molecules has revolutionized drug discovery. In silico drug design allows researchers to screen millions of potential drug candidates quickly and efficiently, narrowing down the options to those with the most promising 2D structures for further study. This can greatly accelerate the drug discovery process and lead to the development of new and more effective treatments for a variety of diseases. In summary, the

2D structure of molecules is a fundamental concept in chemistry and drug discovery. Understanding the shape and orientation of functional groups on a molecule can provide important information about its potential interactions with target molecules in the body, and computational methods for predicting and analyzing 2D structures have greatly accelerated the drug discovery process. The development of drugs highlights the importance of 2D structures in drug design and their potential to improve human health.

4.3. First approach for activity prediction of molecules with CDK1 [274]

In this section, we present our suggested methodology for activity prediction, which is briefly demonstrated in the flowchart in Figure 4.4. Utilizing chemical compounds that are represented in 2D space, each atom in the compound is placed according to its coordinates in this space. Utilizing the 2D pharmacophore fingerprint, we describe these compounds. We use feature selection algorithms to choose the top 1024 features because the generated fingerprints are rather huge. After many experimental assessments, this number is decided. In the end, we suggest two deep learning models to forecast the biological activity/inactivity of these compounds. The first uses a deep neural network DNN and a 1D vector as input, which stands for the estimated 2DPF. The second model is a CNN, which takes the exact same input data but is represented as a 2D vector to allow the application of 2D filters and max-pooling strategies, further shrinking the data size.

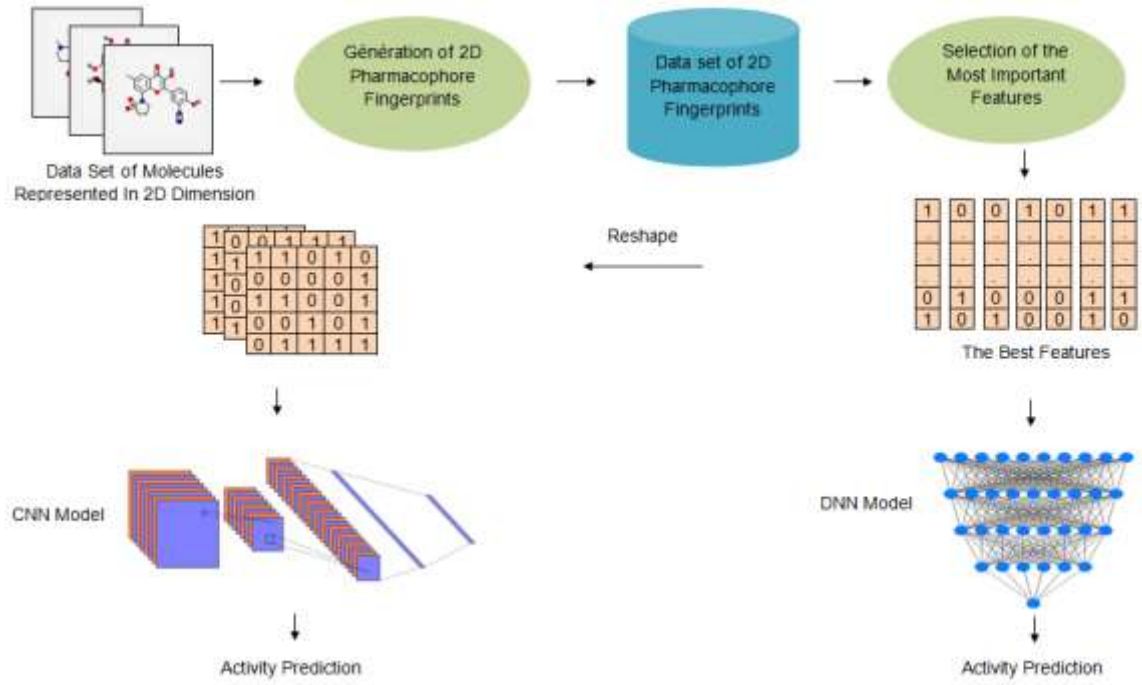


Figure 4.4. Proposed approach for activity prediction of molecules with CDK1 using 2DPF.

4.3.1. 2D Pharmacophore Fingerprint (2DPF)

A 2D pharmacophore fingerprint can be defined as a binary vector representation that captures features and pattern within a chemical compound, indicating the presence or absence of specific pharmacophore.

4.3.1.1. Pharmacophore concept

Pharmacophore, according to the International Union of Pure and Applied Chemistry (IUPAC), is the collection of steric and electronic features, arranged in a specific pattern, required to achieve the best supramolecular interactions with a particular biological target structure and to activate (or inhibit) its biological response.

This means that pharmacophores are crucial patterns of molecular features of drugs that facilitate their interaction with specific biological targets, such as enzymes or receptors. These patterns encompass electronic and steric properties, which are necessary to achieve optimal supramolecular interactions with the target structure. Electronic characteristics refer to the ability of the pharmacophore to form chemical bonds such as hydrogen bonds and electrostatic interactions, while steric properties relate to its size, shape, and orientation, which influence its fit into the active site of the target molecule.

Pharmacophores play a vital role in drug design, as they allow scientists to identify and optimize the key structural features of a drug that are responsible for its therapeutic activity.

Figure 4.5. Pharmacophore model concept [261].

In Figure 4.5, we present an example of molecule that carries a pharmacophore model (Figure 4.5 (A)) where the circles in this model represent the features used to represent certain chemical properties that are important for the interaction between a ligand and a receptor. The arrow inside a circle represents a directional feature (Figure 4.5 (B)) which is a type of feature that indicates the preferred direction of a ligand's interaction with a receptor. The use of directional features in pharmacophore models can help to improve the accuracy of predicting ligand-receptor interactions by taking into account not only the presence of specific chemical groups but also their spatial orientation and directionality. (Figure.5 (C)) represents the final model of pharmacophore which illustrates the distance between the crucial features that necessary of the interaction with specific biological target. This model can be used as

query to search which molecules that carry it, and thus they can be considered as a potential drugs.

4.3.1.2. Pharmacophoric features

Pharmacophoric features are the essential structural elements of a drug molecule (Figure 4.6). They can include both chemical and physical properties of a molecule, such as its size, shape, polarity, and charge distribution. Common pharmacophoric features include hydrogen bond donors and acceptors, aromatic rings, and hydrophobic groups.

A. Hydrogen bond acceptor (HA)

A hydrogen bond acceptor is a functional group or an atom in a molecule that can form a hydrogen bond by accepting a hydrogen atom bound to another electronegative atom (Figure7). In general, hydrogen bond acceptors are characterized by having at least one lone pair of electrons available for bonding. Some common examples of hydrogen bond acceptors include oxygen, nitrogen, and sulfur atoms. These atoms are more electronegative than hydrogen, which allows them to form relatively strong hydrogen bonds with hydrogen atoms bound to other molecules. In biological systems, hydrogen bond acceptors play an important role in the formation of protein structures, DNA and RNA, as well as in the binding of ligands to receptor sites. By forming hydrogen bonds with other molecules, these functional groups can help to stabilize and direct the interactions between different molecules and promote specific biological activities [263].

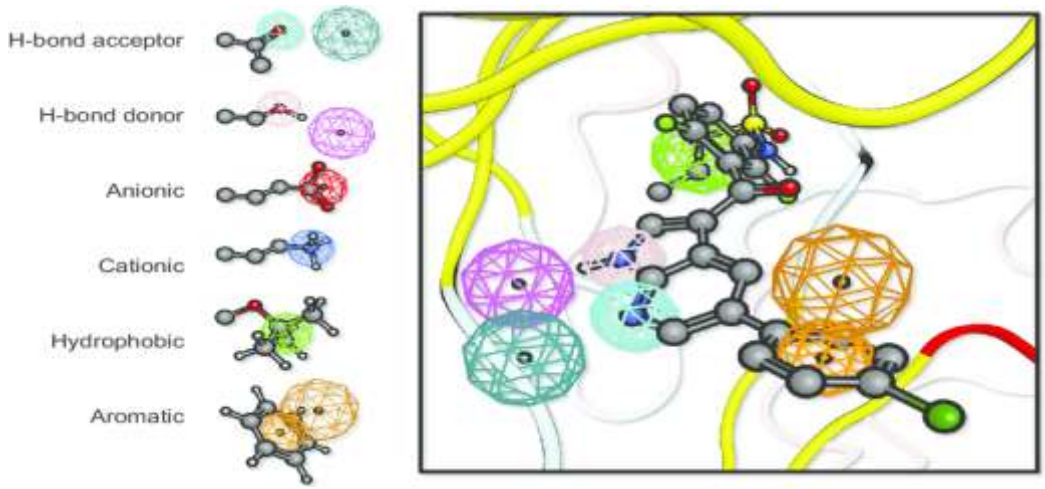


Figure 4.6. Pharmacophoric Features Types [262].

B. A hydrogen bond donor (HD)

A hydrogen bond donor is a functional group or an atom in a molecule that can donate a hydrogen atom to form a hydrogen bond with another electronegative atom (Figure 4.7).

Typically, hydrogen bond donors are characterized by having a hydrogen atom bonded to an electronegative atom, such as nitrogen, oxygen, or fluorine. When a hydrogen bond donor is in the presence of a hydrogen bond acceptor, such as another electronegative atom, it can donate its hydrogen atom to form a relatively strong electrostatic interaction known as a hydrogen bond. This type of interaction is important in many biological processes, such as protein folding, DNA replication, and enzyme-substrate binding. The presence of hydrogen bond donors and acceptors can significantly influence the chemical and biological properties of a molecule and are often considered in drug design and other chemical applications.

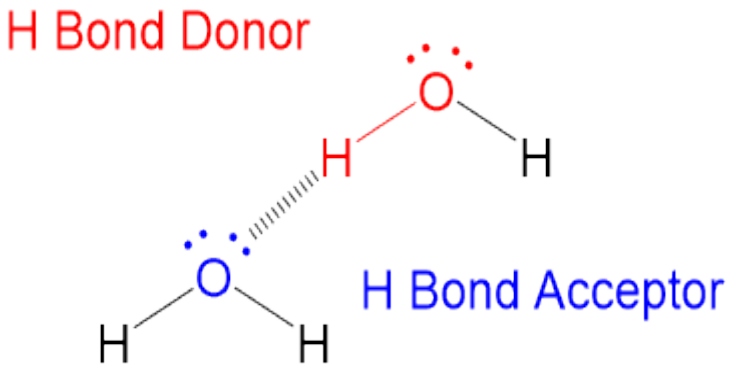


Figure 4.7. Hydrogen Bond [264].

C. Anionic (negative functional groups)

Anionic functional groups are negatively charged functional groups in a molecule, which can donate electrons to form a covalent bond with other atoms or molecules (Figure 4.8). The presence of an anionic functional group in a molecule can significantly affect its chemical and physical properties, including its solubility, reactivity, and biological activity. Anionic functional groups are often found in biological molecules such as amino acids, nucleotides, and carbohydrates. They play important roles in cellular signaling, enzymatic reactions, and other biological processes [265].

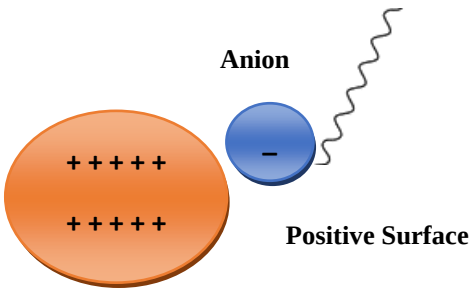


Figure 4.8. Anion Attracted to Positive Surface.

D. Cationic (positive functional groups)

Cationic functional groups are positively charged functional groups in a molecule, which can accept electrons to form a covalent bond with other atoms or molecules (Figure 4.9). The presence of a cationic functional group in a molecule can significantly affect its chemical and physical properties, including its solubility, reactivity, and biological activity. Like the previous one, cationic functional groups are often found in biological molecules such as amino acids, peptides, and proteins. They play important roles in cellular signaling, enzymatic reactions, and other biological processes [266].

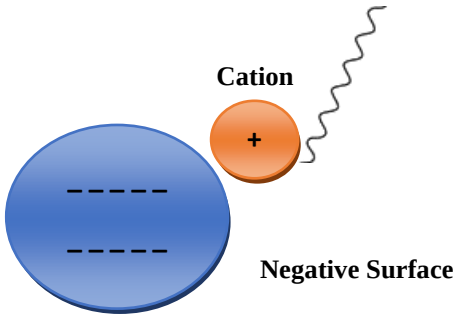


Figure 4.9. Cation Attracted to Negative Surface.

E. Hydrophobic

Hydrophobicity refers to the tendency of a molecule or a part of a molecule to avoid interaction with water molecules (Figure 4.10). This occurs because water molecules are highly polar and interact strongly with other polar molecules or ions through electrostatic interactions. In contrast, hydrophobic molecules or regions of molecules have low polarity and interact poorly with water molecules, preferring instead to interact with other hydrophobic molecules or regions. The hydrophobic effect plays a significant role in many biological processes, including protein folding, membrane formation, and ligand binding. In biological systems, hydrophobic molecules and regions are often found in the interior of proteins and membranes, where they are shielded from contact with water molecules. Hydrophobic interactions between ligands and proteins also play a critical role in drug discovery and design [**267**]. In pharmacophore modeling, hydrophobicity is often represented as a feature that describes the presence of hydrophobic groups or regions in the ligand that interact with hydrophobic regions of the target protein.

F. Aromatic

Aromatic features refer to the presence of one or more aromatic rings in a molecule that can interact with a target receptor or enzyme (Figure 4.11). These features are represented in pharmacophore models as a set of points that correspond to the positions of the ring atoms and are used to define the spatial constraints of the feature, such as the distance between the ring atoms or the orientation of the ring plane. Aromatic features can play an important role in

the activity of many drugs and biologically active molecules. In pharmacophore modeling, the inclusion of aromatic features can help researchers to better understand the molecular mechanisms underlying drug activity and design more effective drugs with improved specificity and potency [269].

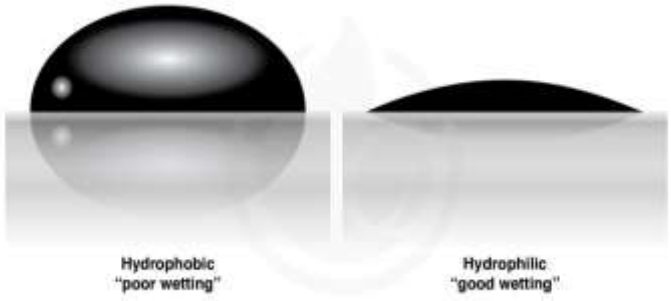


Figure 4.10. Hydrophobic Group [268]

The identification and optimization of pharmacophoric features is a crucial step in the drug discovery process, as it allows researchers to design and modify drug molecules with specific biological activities. By understanding the interactions between a drug molecule and its target, scientists can develop more effective and targeted therapies for a wide range of diseases.

Figure 4.11. Aromatic Compound.

4.3.1.3. Generation of 2D pharmacophore fingerprint

A 2D pharmacophore fingerprint (2DPF) is created by integrating diverse distance ranges and features. Distance ranges represent the different intervals of distances between various features within a chemical compound that are taken into account during the generation of the fingerprint. They are typically defined as intervals or sub-ranges of a larger distance range, which is commonly used in 2DPF. The resulting fingerprint can capture more detailed information about the spatial relationships between the features in a chemical compound. we can achieve a number of objectives by using 2DPF, including molecular activity prediction

for specific biological targets and quick comparison of molecular structures to detect similarities and discrepancies. Finding compounds with the necessary pharmacological properties may be simpler, which is beneficial for the creation of new medications. The chemical compounds that we use to extract these fingerprints must be represented in a two-dimensional space. The definition of the pharmacophore space and computation of the fingerprint are the two processes required to generate a 2DPF. The two stages are given as follows:

A. Definition of pharmacophores space

The pharmacophores space is a set of all possible combinations between features (or called points) with different distance ranges. The combinations can be between two features, three, four, or more.

If we consider $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ a set of pharmacophore features and $D = \{d_1, d_2, d_3, \ldots d_n\}$ a set of distance ranges, where: di represents a range of distance and n represents the number of distance ranges, then the pharmacophores space is $W = \{w1, w2, w3, \ldots, wN\}$ where:

- $w_1 \neq w_2 \neq w_3 \ldots \neq w_N$

- With two-points $w_i = (f_j, f_k, d_o)$, where : $1 \leq j, k \leq 6$ and $1 \leq o \leq n$.

- With three-points $wi = \{(f_j, f_k, d_o), (f_k, f_l, d_p), (f_l, f_j, d_q)\}$, *where:* $1 \leq j, k,l \leq 6$ and $1 \leq o, p, q \leq n$.

In our experiment, we use a distance range between 0 and 21 ˚A (Angstrom). we divide it into six sub-ranges as follows: ***[(0,4), (4,7), (7,10), (10,14), (14,17), (17,21)].***

The selection of distance ranges and their division in a 2DPF depends on various factors, including the specific biological target, the chemical nature of the ligands, and the available experimental data.

One approach is to use prior knowledge of the target structure and ligand interactions to guide the selection of distance ranges. For example, if there is evidence that a particular group of atoms is involved in hydrogen bonding with the target; a narrower distance range can be selected to capture this interaction. On the other hand, if the target has a large, flexible binding site that accommodates a variety of ligand conformations, a broader range of distances may be appropriate.

Another approach is to explore different distance ranges and their division systematically to determine the best-performing fingerprint. This can be done by generating a range of fingerprints using different distance ranges and evaluating their performance in VS experiments. The distance ranges that result in the most accurate and efficient predictions can be selected.

We relied on the second approach in determining distance range and its divisions. To create two different forms of 2DPF, we form the pharmacophores using two and three points, respectively. The created pharmacophores space has N combinations that are possible; N varies depending on the number of points and distance ranges used to create the pharmacophores. (Figure 4.12(a)).

B. Calculation of fingerprint

The process of calculating a fingerprint entails building a binary vector and placing each combination of pharmacophores in the vector. A number of 1 indicates the presence of a pharmacophore in the chemical, while a value of 0 indicates its absence (Figure 4.12 (b)).

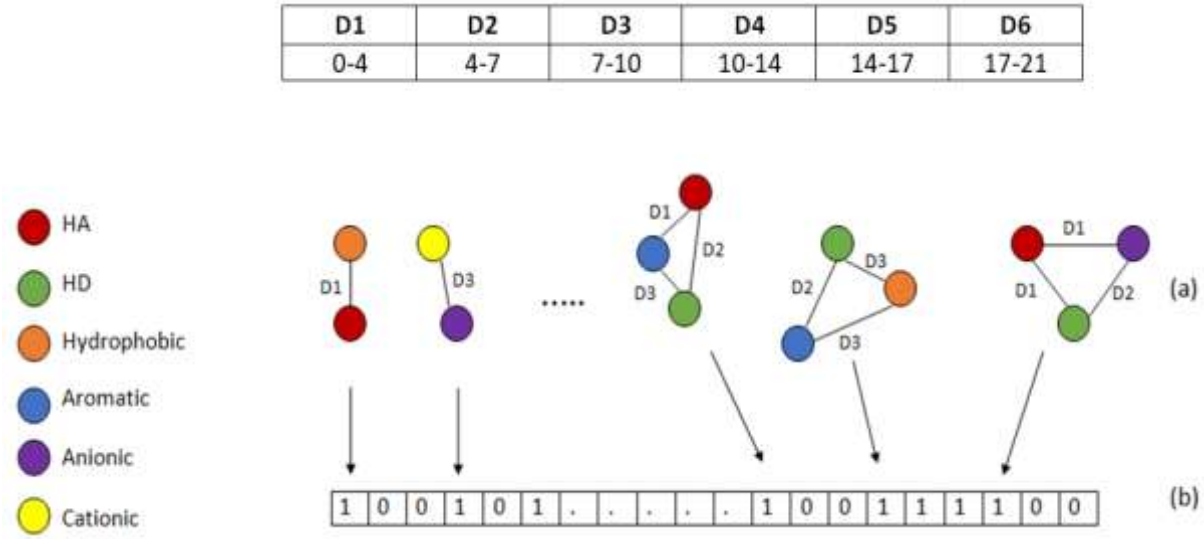


Figure 4.12. Generation of 2DPF.

4.3.2. Features selection

When generating pharmacophore fingerprints using more than two points, the resulting fingerprint size is large. It is not practicable to use such a big fingerprint because it takes a lot of computing power and training time. The most important features should be chosen throughout the feature selection process in order to reduce the size of the fingerprint. With this method, the model can be more accurately interpreted and the training process is more rapid without using up a lot of storage space. Choosing a subset of pertinent features to utilize in the construction of a predictive model is the general definition of feature selection. By choosing the most significant characteristics, feature selection in ML and data analysis plays a critical role in lowering the dimensionality of the data, hence increasing the model's accuracy and effectiveness.

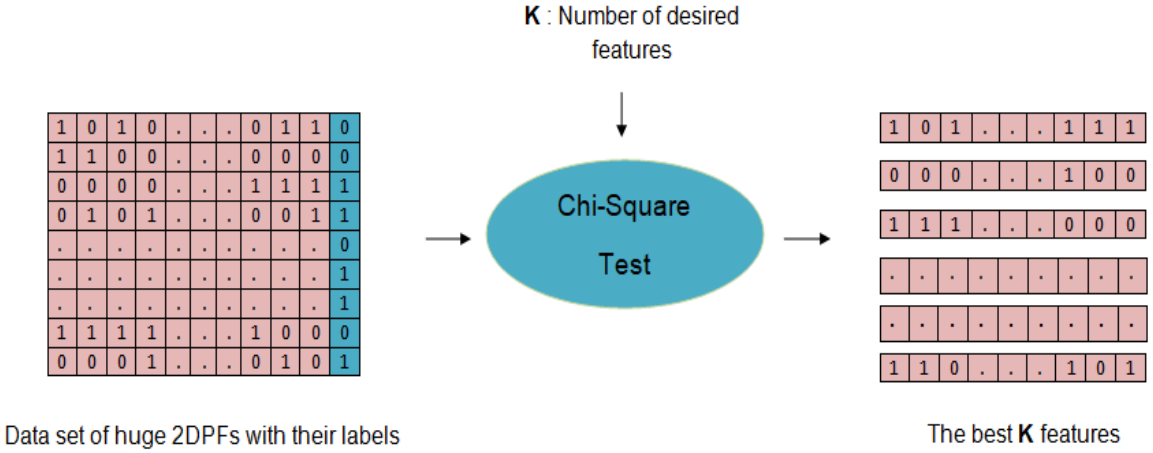


Figure 4.13. Feature selection scheme on 2DPF.

In this study, we employed the Chi-square test to address this problem, where it takes as parameters all generated fingerprints with their labels (Active or inactive) and the number of desired features (Figure 4.13). This latter is determined by several experiences to achieve the optimal size of the fingerprint that gives a good result.

The Chi-square test, a popular statistical technique for feature selection in the literature, is used to ascertain whether there is a statistically significant relationship between two categorical variables. Using the actual observed frequencies of the categories, it is used to test the null hypothesis that there is no correlation between the variables. The chi-square statistic has the following formula:

$$\chi^2 = \Sigma (O_i - E_i)^2 / E_i \qquad \text{Eq (4.1)}$$

Where:

O_i = the observed frequency in category i

E_i = the expected frequency in category i

The application the chi-square test for feature selection in our study is illustrated as follows:

Table 4.2 that shows the frequency of occurrence of each feature (pharmacophore) for each class (Actives and Inactives). Where a, b, c, d, e, f are the number of samples in each class that have an identified pharmacophore (Pharm) and n represents the number of features.

Table 4.2. Occurrence of features in classes.

	Pharm 1	**Pharm 2**	**.**	**.**	**Pharm n**
Actives	a	b	**.**	**.**	c
Inactives	d	e	**.**	**.**	f

A. Calculate the expected frequencies

Calculate the expected frequency for each cell in the table under the assumption of independence between the feature and the class. The formula for the expected frequency of

cell (i, j) is:

Expected frequency of cell (i, j) = (row i total) x (column j total) / (N) Eq (4.2)

Where **N** is the total number of samples.

For example, the expected frequency for cell (1,1) in the Pharm 1 is:

E_{11} = [(a + b + … + c) x (a + d)] / (a + b +…+ c + d + e +… + f)

It must calculate the expected frequency for each cell in table.

B. Calculate the chi-square statistic for each feature

Using the Eq (4.1), calculate the chi-square statistic for each feature(pharmacophore) by summing up $(O_i - E_i)^2 / E_i$ for each cell in the corresponding table. For example, the chi-square statistic for pharm 1 would be:

χ^2 (Pharm 1) = [$((a - E_{11})^2 / E_{11}) + ((d - E_{21})^2 / E_{21})$]

C. Calculate the degrees of freedom

The degrees of freedom for each feature in a chi-square test are calculated as the number of categories in the feature minus 1. In our case we have two categories (actives and inactives), so the degrees of freedom would be 2-1=1.

D. Calculate the p-value

The p-value is a statistic that expresses the strength of the evidence against a null hypothesis. If the null hypothesis is correct, it represents the likelihood of observing a result that is equally extreme to or more extreme than the one that was actually observed. A null hypothesis is a statement that, until we have evidence to the contrary, is assumed to be true when conducting a hypothesis test. The assertion we are testing, the alternative hypothesis, is one that holds if there is sufficient evidence to reject the null hypothesis. Then, using the observed data and the null hypothesis, the p-value is determined. The alternative hypothesis is accepted if the p-value is low (less than the selected significance level, often 0.05 or 0.01). As a result, we can draw the conclusion that there is a statistically significant association between the variables under investigation and that there is strong evidence to refute the null hypothesis. On the other hand, we fail to reject the null hypothesis if the p-value is high (higher than the selected significance level). This means that we cannot draw the conclusion that there is a substantial association between the variables because there is not enough data to support the alternative hypothesis.

Figure 4.14 shows Chi-squared distribution table where the rows represent the degrees of freedom (DF) and the columns represent the probability (p-value) associated with the chi-squared statistic for a given level of significance. Each cell in the table represents the critical value of chi-squared for a specific combination of degrees of freedom and probability level.

E. Select the significant features

Generally, to select the significant features, we must compare each p-value to a predetermined significance level, and select the features with p-values less than the significance level. These features are considered statistically significant and can be used for binary classification.

In our case, we predetermined a number of features; we do not need to specify a significance level. Instead, we can rank the features in ascending order based on their p-values and choose the top k features where k is the predetermined number of features we want to select. The p-value threshold for significance will be determined by the rank of the k-th feature, and any feature with a p-value less than or equal to this threshold will be considered significant and selected for further analysis.

	P										
DF	0.995	0.975	0.2	0.1	0.05	0.025	0.02	0.01	0.005	0.002	0.001
1	.0004	.00016	1.642	2.706	3.841	5.024	5.412	6.635	7.879	9.55	10.828
2	0.01	0.0506	3.219	4.605	5.991	7.378	7.824	9.21	10.597	12.429	13.816
3	0.0717	0.216	4.642	6.251	7.815	9.348	9.837	11.345	12.838	14.796	16.266
4	0.207	0.484	5.989	7.779	9.488	11.143	11.668	13.277	14.86	16.924	18.467
5	0.412	0.831	7.289	9.236	11.07	12.833	13.388	15.086	16.75	18.907	20.515
6	0.676	1.237	8.558	10.645	12.592	14.449	15.033	16.812	18.548	20.791	22.458
7	0.989	1.69	9.803	12.017	14.067	16.013	16.622	18.475	20.278	22.601	24.322
8	1.344	2.18	11.03	13.362	15.507	17.535	18.168	20.09	21.955	24.352	26.124
9	1.735	2.7	12.242	14.684	16.919	19.023	19.679	21.666	23.589	26.056	27.877
10	2.156	3.247	13.442	15.987	18.307	20.483	21.161	23.209	25.188	27.722	29.588
11	2.603	3.816	14.631	17.275	19.675	21.92	22.618	24.725	26.757	29.354	31.264
12	3.074	4.404	15.812	18.549	21.026	23.337	24.054	26.217	28.3	30.957	32.909
13	3.565	5.009	16.985	19.812	22.362	24.736	25.472	27.688	29.819	32.535	34.528
14	4.075	5.629	18.151	21.064	23.685	26.119	26.873	29.141	31.319	34.091	36.123
15	4.601	6.262	19.311	22.307	24.996	27.488	28.259	30.578	32.801	35.628	37.697
16	5.142	6.908	20.465	23.542	26.296	28.845	29.633	32	34.267	37.146	39.252
17	5.697	7.564	21.615	24.769	27.587	30.191	30.995	33.409	35.718	38.648	40.79
18	6.265	8.231	22.76	25.989	28.869	31.526	32.346	34.805	37.156	40.136	42.312
19	6.844	8.907	23.9	27.204	30.144	32.852	33.687	36.191	38.582	41.61	43.82
20	7.434	9.591	25.038	28.412	31.41	34.17	35.02	37.566	39.997	43.072	45.315

Figure 4.14. Chi-squared distribution table [270].

4.3.3. First proposed predictive model: Deep Neural Network

4.3.3.1. Architecture

The used DNN model has eight interconnected layers in total, as shown in Figure 4.15. After being multiplied by weights and transmitted to the first hidden layer, which is made up of 512 neurons, the first layer serves as the input layer with 1024 bits, receiving input data. The 256

neurons in the following layer link to the 128 neurons in the layer below. 64 and 32 neurons, respectively, make up the fourth and fifth buried layers. One neuron makes up the output layer, which is connected to the final hidden layer, which has 16 neurons. An equal number of neurons in the input layer with the size of the input data allow the model to capture all the information contained in the input and learn relevant features and patterns during training, leading to better performance and more accurate predictions. Having the same number of neurons as input features also simplifies the model design and avoids the need for complicated preprocessing or feature engineering steps.

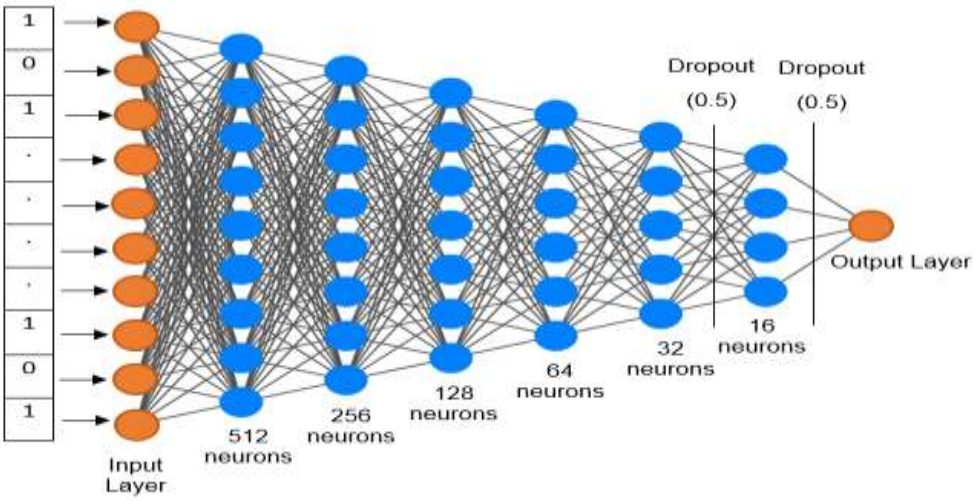


Figure 4.15. The Proposed DNN architecture.

We notice that in each layer there are a lower number of neurons than in the previous layer because having fewer neurons in one layer than in the previous in a deep neural network can prevent overfitting, improve computational efficiency and improve generalization performance. However, the optimal architecture of a neural network depends on various factors, and experimentation is often required to find the best configuration for a particular task. So we have had several experiments to fix this architecture. Using powers of 2 for the number of neurons in each layer can have practical advantages such as more efficient memory allocation and simplified hyperparameter search. Each neuron in the network is an activation function that takes the input data and the neuron's weights, adds a value, and applies a non-linear activation function (Figure 4.16).

X [x_1, x_2, x_3...x_n] is the input to the neuron, W [w_1, w_2, w_3...w_n] is the weights vector, b is the bias term, and f is the activation function. Then, the output y of the neuron is given by:

$$y = f(WX + b) \qquad \text{Eq (4.3)}$$

Where WX + b is the dot product of the input data x and the neuron's weights w, plus the bias term b. This linear combination is then passed through the activation function f to introduce

non-linearity and produce the output y. The weights w and bias b are learned during the training process to optimize the network's performance on the given task.

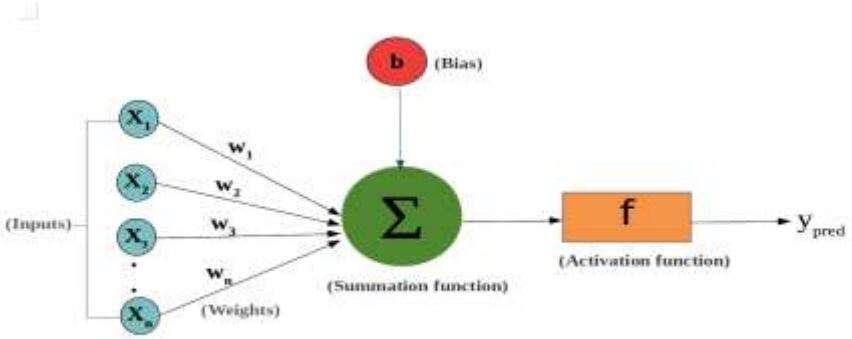
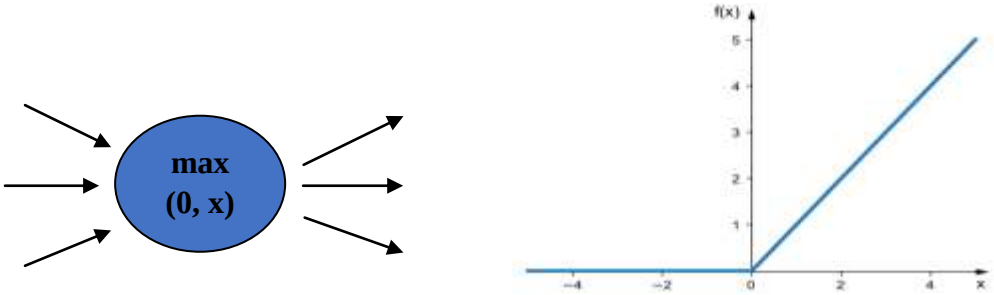


Figure 4.16. A Neuron in the deep neural network .

The Rectified Linear Unit (ReLU) function is used as the activation function in all neurons except the output neuron (Figure 4.17 (A)), and its formula Eq (4.4) is applied.

$$f(x) = \max(0, x) \qquad \text{Eq (4.4)}$$

Where x is the input to the activation function, and max () returns the maximum of its arguments. In other words, the ReLU function returns 0 for any negative input and returns the input itself for any positive input (Figure 4.17(B))



A **B**

Figure 4.17. ReLU function.

We use The ReLU as an activation function because it is popular in deep learning due to its simplicity, non-linearity, sparsity, gradient stability, and empirical success. It is a simple and efficient function to compute, allowing for easy implementation in neural networks. Its non-linear nature allows for the modeling of complex non-linear relationships in data. It provides a sparse representation of data, which can help reduce overfitting in the model. The constant gradient of either 0 or 1 makes training neural networks with ReLU activation functions more stable and easier to optimize. The empirical success of ReLU has been demonstrated across a wide range of deep learning applications, making it a standard choice for many neural network architectures.

The ReLU activation function is not used in the output layer of neural networks because it cannot directly produce a binary output of 0 or 1. The ReLU function outputs values in the range [0, +∞), and to produce binary outputs, a threshold needs to be applied to the output. This thresholding can introduce additional complexity and may require additional training, making it less practical than other activation functions such as the sigmoid function, which is specifically designed to produce a binary output in the range [0, 1]. The sigmoid function is frequently employed in binary classification jobs because it converts any real value to a probability value between 0 and 1, which can then be thresholded to generate the binary output. So, we utilize the sigmoid function, which is represented by Eq (4.5), as the activation function for the output layer.(Figure 4.18 (A))

$$f(x) = 1 / (1 + e^{(-x)}) \qquad \text{Eq (4.5)}$$

Where x is the input value and e is the mathematical constant referred to as Euler's number, which is roughly 2.71828. The sigmoid function has an S-shaped curve with values near to 0 for extremely negative inputs and values close to 1 for highly positive inputs. The sigmoid function has a value of 0.5 at x=0 (Figure 4.18 (B)), which is frequently employed as a threshold for categorizing binary outcomes. The sigmoid function aids in introducing non-linearity into the network, enabling it to learn more intricate functions.

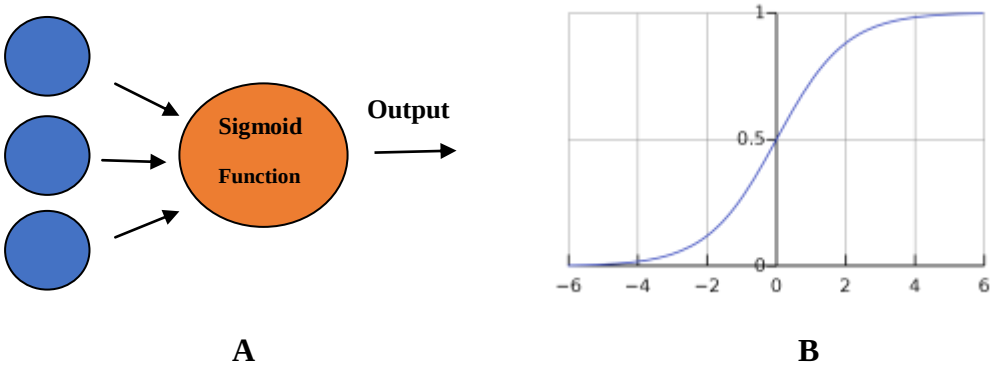


Figure 4.18. Sigmoid function.

4.3.3.2. Hyperparameters

Two matrices are used as input during the training and testing of the model in order to determine the best values for the parameters (or weights). The first matrix is of size [n * m], where n stands for the number of features (1024), and the columns m stand for the number of compounds chosen for the training and testing, and the second matrix is of size [1*m], which stands for the label of each compound in the training and test set. Initializing the weights is the first step in training a deep neural network. Next, input data is fed through the network using forward propagation, the loss is calculated using a loss function, the gradient of the loss function with respect to each layer's weights is computed using backpropagation, the weights are updated using an optimization algorithm, and the process is repeated until the loss

function falls below a certain threshold. The neural network is trained by this procedure to anticipate new input correctly.

After determining the proper parameters that yield an acceptable result with the test set, we can use the model to predict the class of any molecule specified in 2DPF with a size of 1024 bits. MSE was employed as the loss function and ADAM was chosen as the optimizer in order to change the parameters of our model. To address overfitting problems, we included a dropout penalty in the final two hidden layers. The remaining hyperparameters used in our model are shown in Table 4.3.

Table 4.3. Hyperparameters setting of DNN.

Optimizer	Loss function	Learning rate	Dropout	batch size
ADAM	MSE	0.002	(0.5)	64

• Learning rate: 0.002

When a neural network is being trained, one of the hyperparameters that controls the step size at each iteration is the learning rate. It regulates how frequently the model parameters are changed to minimize the loss function. The model may converge fast as a result of a high learning rate, but it may also overshoot and miss the ideal solution. A low learning rate, on the other hand, could take a while to converge and might get stuck in a local minimum. (Figure 4.19).

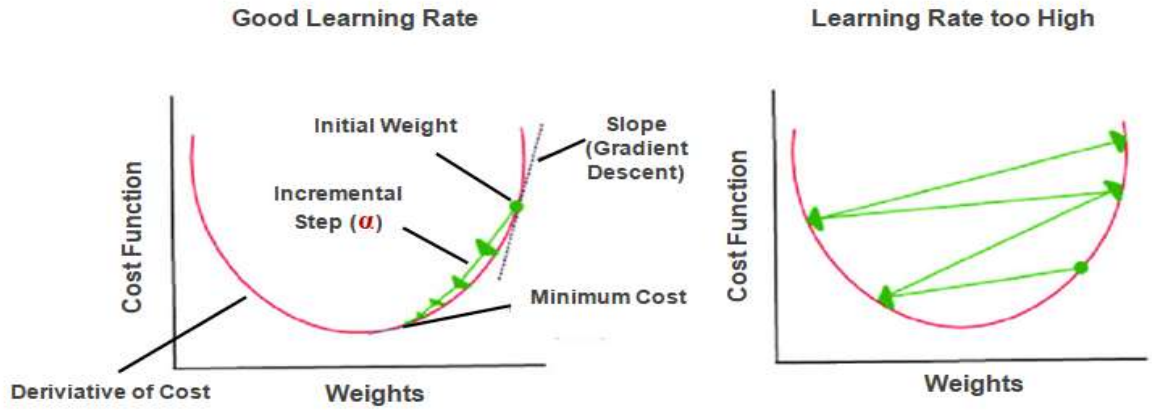


Figure 4.19. Learning Rate during training.

The update equation for a parameter **w** at iteration **t** using learning rate **α** and gradient **g** is as follows:

$$w(t+1) = w(t) - \alpha * g \quad \text{Eq (4.6)}$$

The parameter w is updated by subtracting the product of the learning rate α and the gradient g of the loss function with respect to w. This update rule moves the parameter in the direction of steepest descent, which is the direction that minimizes the loss function. The learning rate determines the step size of this movement, and it is an important hyperparameter to tune in order to achieve optimal performance in ML models.

The learning rate is typically set to a small value, such as 0.001 or 0.01, and adjusted based on the performance of the model during training.

• Optimizer: ADAM

An optimization approach called ADAM (Adaptive Moment Estimation) optimizer is frequently used in deep learning, notably in neural networks. Adaptive learning rates are used for each parameter depending on the expected first and second moments of the gradients in this extension of stochastic gradient descent. At iteration t, the update equation for a parameter w using the ADAM optimizer is as follows:

$$m(t) = \beta1 * m(t-1) + (1-\beta1) * g(t) \quad \text{Eq (4.7)}$$

$$v(t) = \beta2 * v(t-1) + (1-\beta2) * (g(t))^2 \quad \text{Eq (4.8)}$$

$$m\_hat(t) = m(t) / (1 - \beta1^t) \quad \text{Eq (4.9)}$$

$$v\_hat(t) = v(t) / (1 - \beta2^t) \quad \text{Eq (4.10)}$$

$$w(t+1) = w(t) - \alpha * m\_hat(t) / (sqrt(v\_hat(t)) + \varepsilon) \quad \text{Eq (4.11)}$$

Where: g(t) is the gradient of the loss function with respect to the parameter w at iteration t, α is the learning rate, and ε is a small constant added to the denominator to avoid division by zero. The momentum parameter β1 and the second moment parameter β2 control the decay rates of the historical gradients, and their default values are 0.9 and 0.999, respectively.

The first moment estimate m(t) and the second moment estimate v(t) are used to calculate the adaptive learning rate for each parameter. The values m_hat(t) and v_hat(t) are bias-corrected estimates of the first and second moments, respectively, which are then used to update the parameter w. The ADAM optimizer is known for its robustness and fast convergence, and is widely used in deep learning applications.

• Loss function: MSE

A common loss function in regression issues, including neural networks, is mean squared error (MSE). The average squared difference between the expected and actual values is what is measured. MSE is calculated as the squared difference between the expected and actual values multiplied by the sample size (Eq (4.12)).

$$MSE = (1/n) * \Sigma(y\_predicted - y\_True)^2 \quad \text{Eq (4.12)}$$

Where:

y_predicted: predicted output value

y_True: true output value

n: number of samples

The goal of training a neural network is to minimize the MSE loss function, which corresponds to reducing the overall difference between the predicted and actual values.

• **Dropout: 0.5**

Dropout is a regularization method used to stop neural networks from overfitting. During training, a fraction of the neurons in a layer are randomly eliminated (set to zero) (Figure 4.20). This has the effect of lessening neuronal co-adaptation and promoting the network to learn stronger features. To achieve the best performance, the dropout rate, a hyperparameter, must be tweaked and is commonly set between 0.2 and 0.5. All neurons are employed during test or validation, but their outputs are scaled by the dropout rate to account for the fact that training used fewer neurons than test or validation.

When a model grows overly complicated and begins to fit the noise in the training data instead of the underlying patterns, this is known as overfitting. Poor performance on fresh, untested data may result from this. During each training iteration, a specific number of neurons in a layer are randomly eliminated (set to zero) via Dropout. Dropout does this by forcing the remaining neurons to acquire more durable properties that are effective for predicting the target variable. This prevents the model from relying too heavily on any one input or feature. As a result, the model becomes less sensitive to minute input fluctuations and is more likely to generalize well to new data.

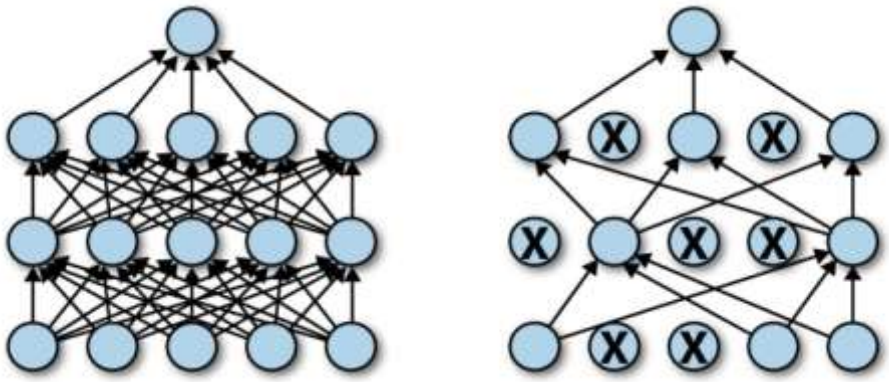


Figure 4.20. Dropout technique [280].

Dropout can be applied in any layer of a deep learning model, including the last layers. The advantage of applying dropout to the last layers is that it can help to reduce the impact of outliers or noisy inputs on the final prediction. By randomly dropping out a certain percentage of neurons in the output layer during training, the model becomes less sensitive to small variations in the input and is more likely to produce robust predictions.

The dropout operation is a type of ensemble learning in which distinct sub-models are trained on various subsets of the data and their predictions are pooled at test time. It is a straightforward but efficient method that is now a common component of much deep learning architecture.

• **Batch size: 64**

In neural network training, the batch size refers to the number of samples that are processed together in a single forward/backward pass. Batch size is a hyperparameter that can be tuned during training. A small batch size allows the model to update its weights more frequently, which can result in faster convergence and better generalization. However, a smaller batch size also means that the model is less efficient in utilizing the hardware resources and may have a slower training time. On the other hand, a larger batch size allows the model to utilize the hardware resources more efficiently and may result in a faster training time. However, larger batch sizes also require more memory. The optimal batch size can vary depending on the dataset and the model architecture. In general, larger batch sizes are more common in deep learning due to the large amount of data and computational resources required. Common batch sizes range from 32 to 512, depending on the specific application and available resources.

4.3.4. Second proposed predictive model: Convolution Neural Network

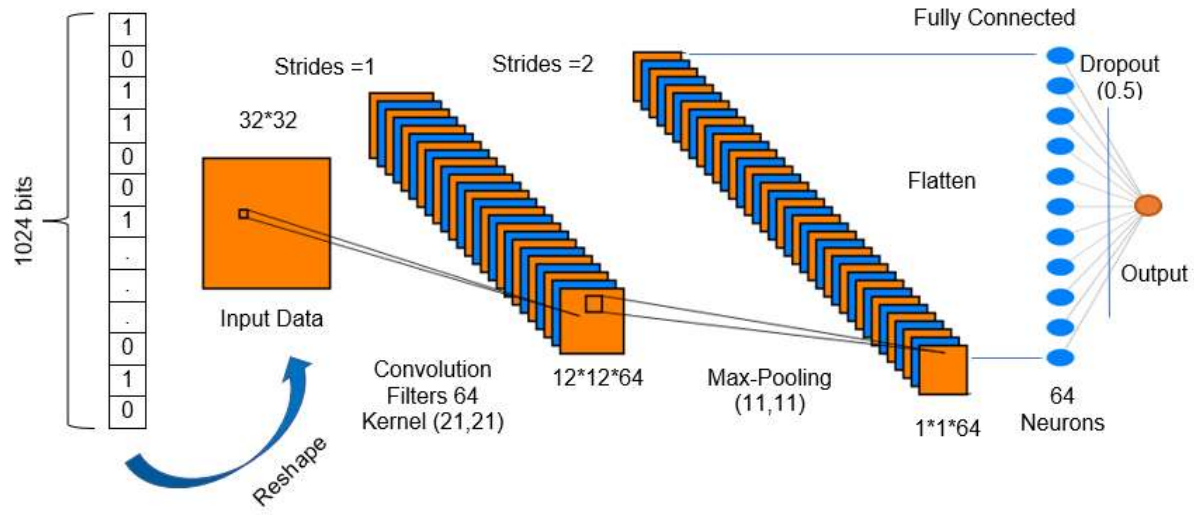


Figure 4.21. The Proposed CNN architecture.

4.3.4.1. Architecture

The application of CNNs becomes essential when working with massive input data. A specified number of filters are used by CNNs to extract features from the data. In order to effectively minimize the quantity of the data without sacrificing important information, max-pooling is frequently applied to the output of the filters. In order to do this procedure, the

maximum value within a clearly defined data patch must be determined. After convolution, padding, a typical procedure in CNNs, involves adding pixels to the matrix. To accomplish a particular objective, a CNN model may have one or more convolution layers, pooling layers, and fully linked networks.

The suggested CNN model is shown in Figure 4.21 and consists of a max-pooling layer with a pool size of 11 * 11 and a convolution layer with 64 filters and a kernel size of 21 * 21. A layer of 64 neurons and a single neuron make up the output layer of the fully connected network. A binary matrix of size 32 by 32 that was created by reshaping a binary vector encoding a 2D pharmacophore fingerprint of size 1024 bits serves as the input to the CNN model. Filters and the max-pooling operation are used to cut down on the amount of features, and the outcomes from the two models are contrasted using the same data.

We have chosen the number of filters and the size of kernel and pooling through a process of trial and error, where different configurations are tested on a validation set to find the optimal hyperparameters. We used kernels with odd dimensions because it has some advantages in CNN. One major advantage is that kernels have a center pixel, which helps to preserve the spatial dimensions of the input data. When kernels with even dimensions are used, it is impossible to have a center pixel that represents a single point in the input data. Another advantage is kernels with odd dimensions have a symmetric shape, which can help to capture symmetrical patterns in the input data. Symmetric filters can also help to reduce the likelihood of overfitting, as they have fewer parameters to learn. Additionally, when using padding to preserve the spatial dimensions of the input data, using kernels with odd dimensions, in their size allows for an equal number of padding pixels to be added to both sides of the input data, ensuring that the output feature map has the same spatial dimensions as the input data.

The first layer of the fully connected network consists of 64 neurons because the output at the end of the convolution and max-pooling operations is 64 bits.

The Sigmoid activation function is applied to the output layer and the ReLU activation function applied to all other layers. ReLU is used in convolutional layers (Figure 4.22). It is applied to the output of each convolutional operation (More details in Figure 4.25), which generates the corresponding activation map, which is then passed on to the next layer in the network, where it is convolved with another set of filters, and the process is repeated. In fully connected layers, the activation functions are also applied after the linear transformation of the input data. It is applied like DNN**.**

4.3.4.2. Hyperparameters

Unlike the previous DNN model, we employ Mean Absolute Error as the loss function, and ADAM was selected as the optimizer to change the model's parameters. In the last hidden layer, we also included a dropout penalty. In Table 4.4, the remaining hyperparameters applied to our CNN model are listed.

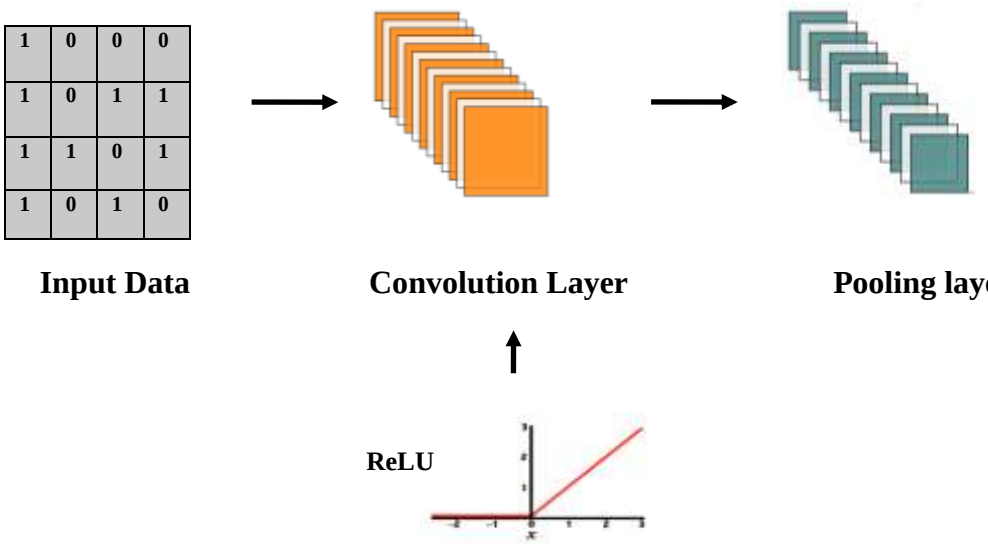


Figure 4.22. Using ReLU function in convolution layer .

Table 4.4. Hyperparameters setting of CNN.

Optimizer	Loss Function	Learning Rate	Kernel Size 2D	Max-pooling	Padding	Dropout	Batch Size
ADAM	MAE	0.002	(21,21) Strides =1	(11,11) Strides =2	Valid	(0.5)	64

- **Loss function : MAE**

The MAE (Mean Absolute Error) loss function is a type of regression loss function used in neural networks. It is calculated as the average of the absolute differences between the predicted and true values.

The formula for MAE is:

$$MAE = (1/n) * \Sigma |y\_true - y\_pred| \quad \text{Eq (4.13)}$$

Where:

y_true: the true values of the target variable

y_pred: the predicted values of the target variable

n: the number of samples in the dataset

MAE is a simple and robust loss function that is less sensitive to outliers than other loss functions, such as the MSE.

- **Kernel : 21 * 21**

Generally, the input data of CNN is an image or binary matrix. A convolutional layer consists of multiple filters (64 in our work), each of which is represented by a kernel which is a small matrix of weights that are used to extract features from it and this operation is called convolution.

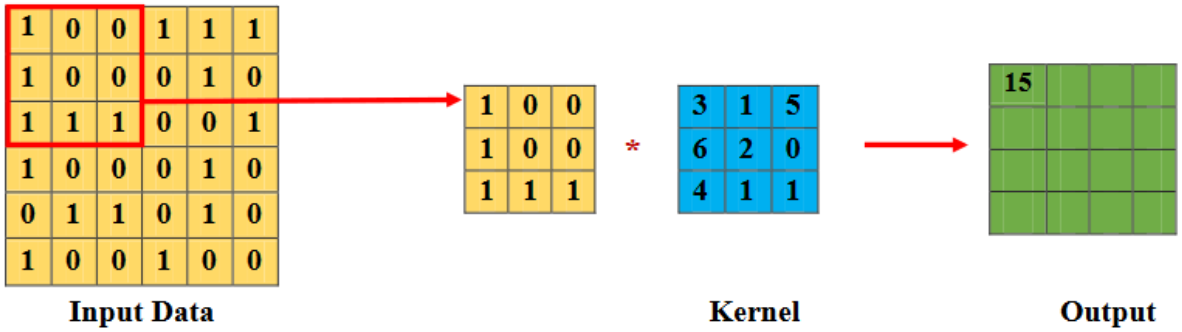


Figure 4.23. Application of Kernel technique in CNN.

The kernel slides over the input data, computing the dot product between the weights in the kernel and the values in the corresponding region of the input matrix. This dot product operation results in a single output value, which is then stored in the output feature map as shown in the Figure 4.23 where: the value 15 in output matrix is calculated as follows:

3*1+1*0+5*0+6*1 + 2*0 + 0*0 +4*1 + 1*1 + 1*1 =15

Like DNN, The values (weights) in the kernel matrix in CNN are learned during training, along with the weights of the other layers in the network, using backpropagation and gradient descent. During backpropagation, the network computes the error between its predicted outputs and the true outputs, and it updates the weights to minimize this error. The update rule for the weights is presented in Eq (4.6).

The number of kernels used in a convolutional layer is a hyperparameter that may be changed during training. Kernels might be varied sizes and forms. The network's capacity to learn and extract relevant characteristics from the input data can be influenced by the size, shape, and number of kernels employed in a convolutional layer.

• Stride :

The hyperparameter stride in CNN controls how big of a step the kernel or pooling takes. It establishes the distance in pixels that the kernel travels in each direction. The kernel travels one pixel at a time when the stride is 1, two pixels at a time when the stride is 2, and so on (Figure 4.24). Increasing the stride can result in a smaller output feature map and fewer computations, which can aid in lowering overfitting and enhancing computational effectiveness. It may, however, also lead to information loss and decreased accuracy. The task at hand and the features of the incoming data determine the ideal stride value.

The output after applying a filter passes through the ReLU function, to introduce nonlinearity into the network. Figure 4.25 clarifies more this operation, the ReLU function is applied element-wise to each element in the output matrix. The result after applying the ReLU

function is a new matrix with same size of output. The resulting activation map highlights the presence of features that are important for the required task.

• **Max-pooling: 11 * 11**

CNNs frequently employ max-pooling, a particular kind of pooling operation, to cut down on the output feature maps' spatial dimensions. Using a mathematical operation over a sliding window of the feature map, the downsampling of the input feature maps is what is known as "pooling." The output value for that region in max-pooling is the highest value that is chosen from the sliding window (Figure 4.26). With max-pooling, the size of the feature maps can be reduced while still preserving the most crucial data.. By reducing the size of the feature maps, the network becomes less computationally expensive and less prone to overfitting. Max-pooling is typically applied after each convolutional layer in a CNN, although other types of pooling operations, such as average pooling, can also be used. The size and stride of the pooling window are hyperparameters that can be tuned during training to control the amount of down sampling.

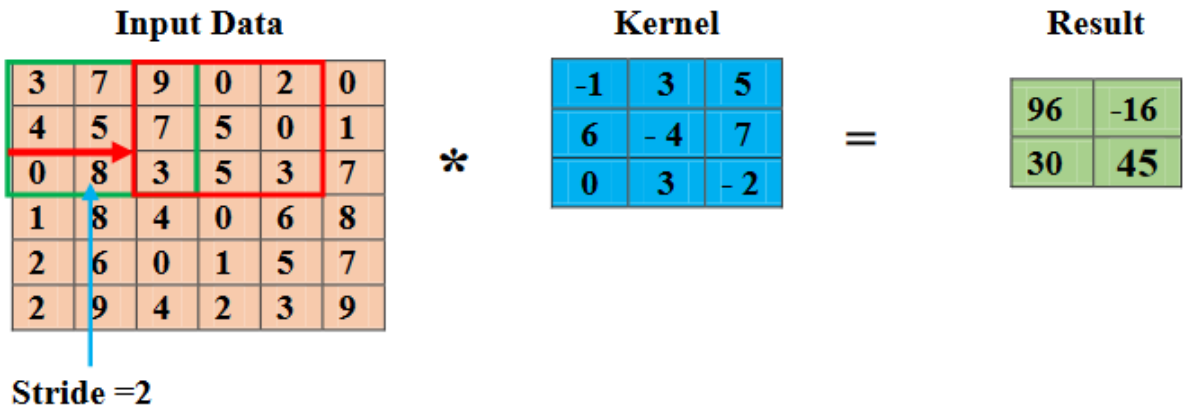


Figure 4.24. Stride of kernel in CNN.

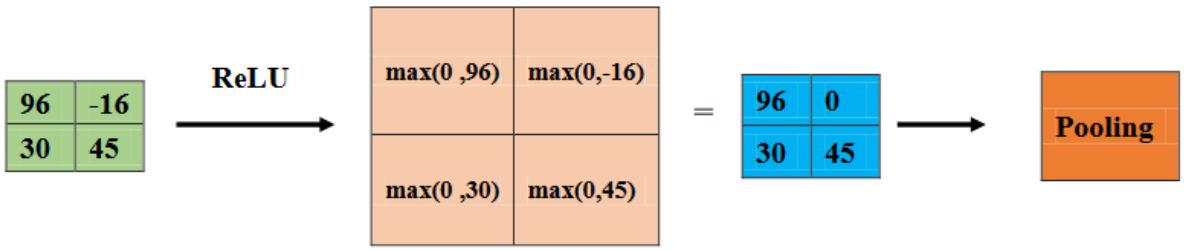


Figure 4.25. Apply ReLU function after convolution operation.

• **Padding: valid**

When performing convolution operations, CNN uses a method called padding to maintain the spatial dimensions of the input data. Because the kernel cannot be applied to the edges of the data matrix, the output feature map is smaller than the input feature map in size. Before performing convolution operations, padding entails adding zeros to the matrix's outer edges (Figure 4.27). This makes it possible to apply the filter on the matrix's edge pixels and guarantees that the output feature map will have the same spatial parameters as the input data. Padding comes in two varieties: "valid" padding and "same" padding. As in our model, no padding is added to the input data in valid padding; hence the size of the output feature map is smaller than that of the input. In order to make the size of the output feature map match that of the input image, padding is applied to the input image.

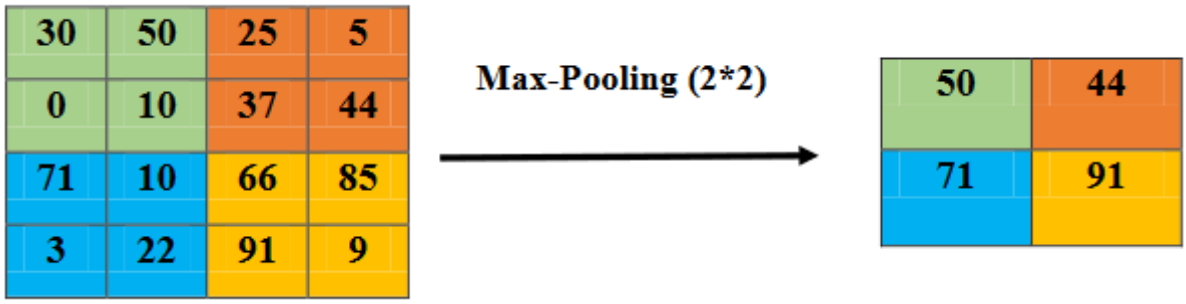


Figure 4.26. Max-Pooling operation.

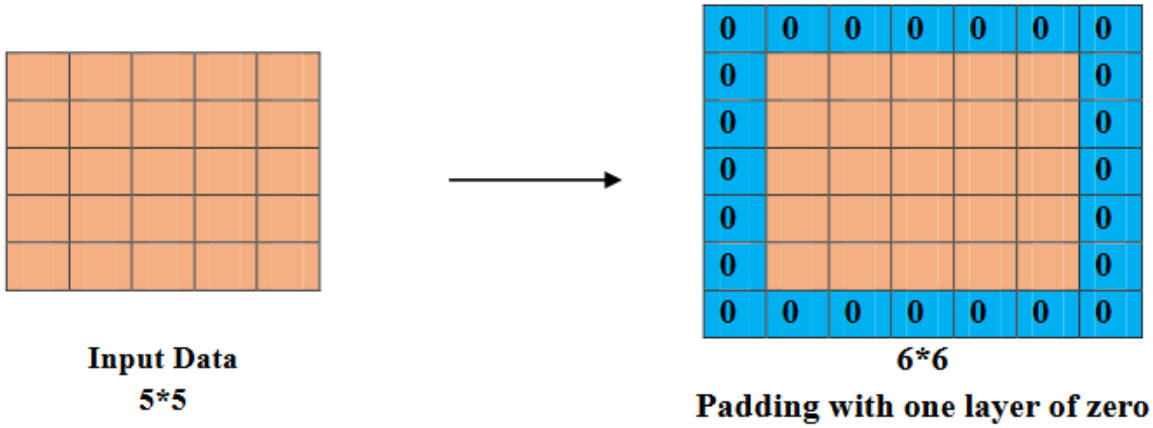


Figure 4.27. Padding operation.

4.3.5. Experimental results

4.3.5.1. Data sets

As we mentioned earlier our study focused on the cyclin-dependent kinase 1 receptor (CDK1) for implementing our proposed approach. A molecule is active if half-maximal inhibitory concentration (IC50) of $\leq$ 9 micrometers [259]. IC50 is a metric for measuring the effectiveness of a chemical compound in inhibiting a receptor protein. Our dataset comprises

2002 samples (active and inactive), a molecule is active with CDK1 means that the molecule is capable of interacting with or modifying the activity of the this protein, Conversely,a molecule is inactive with CDK1 means that the molecule has not been shown to interact with or modify the activity of the CDK1 protein in laboratory assays or experiments. The dataset is divided into three subsets. The first subset consists of 1452 samples and is used for training our models. The second subset, consisting of 270 samples, is used for validating our models and tuning the hyperparameters, while the remaining 280 samples form the test set that is used to evaluate our models. The details of each subset are presented in Table 4.5. In fact, our division of the global dataset was not random, when we collected data from the database, we depended on the IC50 value to distinguish between active and inactive molecules and we noticed that the same IC50 value is shared by several molecules, for example there are 35 active molecules having the same IC50 value which is equal to 1 micrometer, same thing for the rest of the IC50 values with a difference in the number of molecules at each value. In summary, we split the overall dataset into several sets based on the IC50 value, and then we split each set into 3 subsets (training, validation, and testing) to create the final subsets. The objective of this operation is to train the models using balanced data containing all types. We extracted all data information from the CHEMBL [271] database and obtained the two-dimensional structure of compounds from the PubChem [272] database.

Table 4.5. The Division of Dataset.

	Training Data		**Validation Data**		**Test Data**	
	Active	**Inactive**	**Active**	**Inactive**	**Active**	**Inactive**
Protein Complex	298	250	41	32	23	42
Single Protein	464	440	99	98	117	98
Total	762	690	140	130	140	140

• **A single protein** refers to individual protein that is not bound or associated with any other proteins.

• **A protein complex** refers to a group of two or more proteins that are bound together to carry out a specific function. The proteins in a complex may interact with each other in various ways, such as by physical contact or through chemical reactions, to create a larger, functional unit.CDK1, in its natural form, is a protein that forms a complex with other proteins called cyclins to regulate the cell cycle. The CDK1-cyclin complex phosphorylates other proteins to promote the progression of the cell cycle through its various phases, including DNA replication and cell division.

4.3.5.2. Overall performance

In this work, two different kinds of 2DPFs were used. Three points were combined to create the first type, while two and three point pharmacophores were used to create the second type. With an early termination criterion of 30 consecutive epochs for the CNN model and 40 consecutive epochs for the DNN model when the loss value of the validation set is no longer dropping, hyperparameters were tuned using the validation set.

Early stopping is a regularization strategy that helps to avoid overfitting by keeping an eye on the model's performance throughout training and interrupting the process before it begins to do so. Early stopping's fundamental tenet is to keep an eye on the validation error while you train. Typically, as training goes on, the validation error goes down at first, but then starts to go up again as the model starts to overfit (Figure 4.28). Early stopping is a technique that can be used at the point where the validation error starts to rise. After each training epoch, the validation error is monitored in order to determine when to quit early. Training is halted early and the model is reset to the condition with the best validation error if the validation error has not decreased after a predetermined number of epochs. This prevents the model from continuing to learn the training data and overfitting. It is a simple and effective technique for preventing overfitting in deep learning, and is widely used. However, it is important to choose the right values for the patience parameter, as setting it too low may cause the model to stop training too early, while setting it too high may allow the model to overfit.

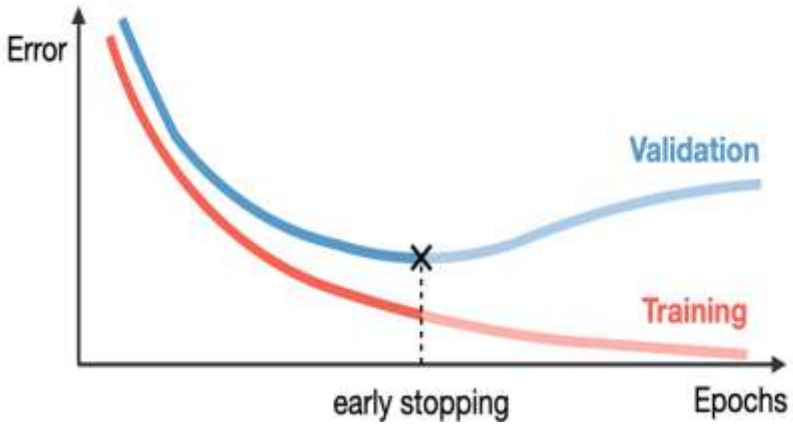


Figure 4.28. Early stopping during training [281].

The models were implemented using Python and the TensorFlow library. The RDKIT library was used to calculate the 2DPF of each compound. The constructed models were evaluated using six performance metrics**:**

• **Accuracy:** The percentage of examples that are correctly classified across all occurrences in a dataset is measured by the performance parameter known as accuracy. The score of the model goes from 0 to 1, with a higher value signifying greater model performance. Accuracy, however, can be deceiving when the dataset is unbalanced, that is, when the number of cases in one class is much larger than the number of examples in another class. Other performance metrics, like precision, recall, and F1-score, may be a better way to assess the success of the model in certain circumstances.

Accuracy = (TP + TN) / (TP + TN + FP + FN) Eq (4.14)

Where:

TP = true positives (the number of accurately anticipated positive instances)

TN = true negatives (the number of cases where a negative outcome was correctly anticipated).

FP = False positives which are occasions where the outcome was projected to be positive but actually wasn't.

FN = False negatives which are instances where the outcome was projected to be negative but actually was not.

• **Sensitivity (SEN) :** Sensitivity, commonly referred to as recall or true positive rate, is a performance indicator that quantifies the percentage of true positive cases that a binary classification model properly classifies. The sensitivity equation is:

SEN = TP / (TP + FN) Eq (4.15)

The ratio of real positive instances to all other positive instances is known as sensitivity. A high sensitivity score means the model is effective at identifying positive occurrences, whereas a low score means the model frequently overlooks positive instances.

• **Specificity (SP E) :** A binary classification model's ability to accurately identify true negative cases is known as specificity, which is a performance parameter. The formula for specificity is:

SPE = TN / (TN + FP) Eq (4.16)

The ratio of true negative instances to all actual negative instances is known as specificity. A high specificity score means the model does a good job of removing negative examples, whereas a low score means the model frequently misclassifies negative examples as positive.

• **F 1 – Score :** Precision and recall, two crucial variables in binary classification tasks, are balanced by the performance indicator known as the F1-score. Its score, which goes from 0 to 1, is the harmonic mean of precision and recall, with a larger number suggesting a more effective model. Formula for the F1 score is:

F1-score = 2 * (precision * recall) / (precision + recall) or

F1-score = 2 * TP / (2 * TP + FP + FN) Eq (4.17)

Where: precision = TP / (TP + FP) (the proportion of true positive predictions among all positive predictions). The weighted average of recall and precision, with equal weight assigned to each statistic, is known as the F1-score. It is an effective metric when the expense of false positives and false negatives is comparable. By properly identifying positive examples while limiting false positives and false negatives, a model with a high F1-score has both high precision and high recall..

• **The Matthews correlation coefficient (MCC)**: A performance statistic known as the Matthews correlation coefficient (MCC) assesses the accuracy of binary (two-class) classifications. It produces a value between -1 and 1, where 1 represents a perfect forecast, 0 indicates a random prediction, and -1 represents a complete discrepancy between the prediction and the true labels. It also takes into account true and false positives and negatives. If the value is 0, the classifier is not superior to random. The Matthews correlation coefficient (MCC) is calculated as follows:

MCC = (TP * TN - FP * FN) /

sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)) Eq (4.18)

The MCC takes into account all four possible outcomes of a binary classification task and produces a value that is more informative than accurate, especially when the data is imbalanced. A high MCC score indicates a strong correlation between the predicted and true labels, while a low or negative score indicates a weak or negative correlation.

• **AUC:** For binary classification issues, the AUC (Area Under the ROC Curve) metric is a frequently used evaluation metric in deep learning. It evaluates, across a variety of classification thresholds, a binary classifier's capacity to distinguish between positive and negative data. The true positive rate (TPR) vs the false positive rate (FPR) is plotted on the ROC (Receiver Operating Characteristic) curve as the classification threshold varies. The area under this curve (Figure 4.29), which spans from 0.0 to 1.0, is the AUC measure. AUC values of 1.0 and 0.5 denote perfect classification performance and random guessing, respectively. The AUC metric is frequently selected over other assessment metrics like accuracy or precision because it can give a more accurate view of the classifier's performance across a range of threshold values and is less sensitive to class imbalance. The AUC statistic can also be used to assess the effectiveness of several binary classifiers and to choose the best threshold value for a particular classification task.

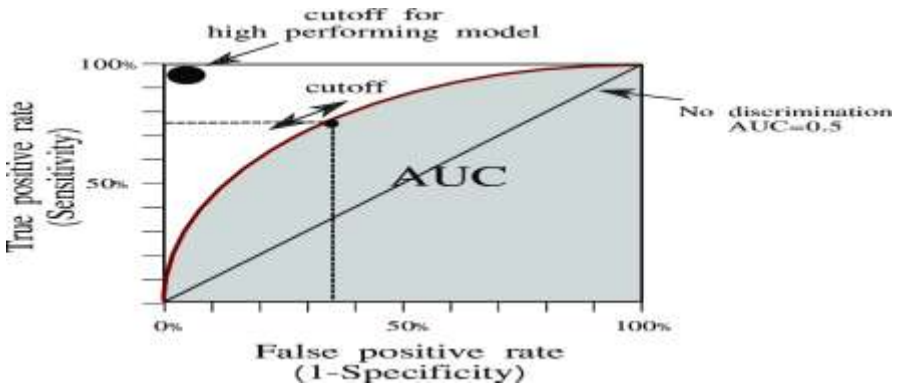


Figure 4.29. Area Under the ROC Curve [177].

A. 2DPF with pharmacophores of three points

We have merged three points that have varying distances, resulting in a fingerprint with 22320 features, which is a very large number. Given this, it is essential for us to employ feature selection techniques to identify the top 1024 features.

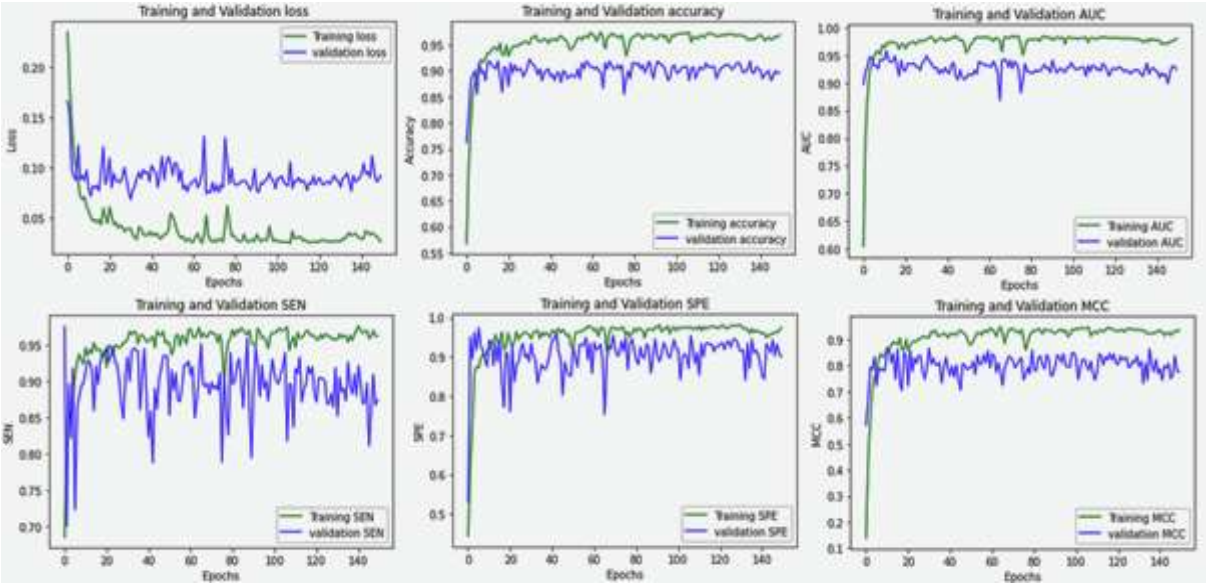


Figure 4.30. The performance evolution of the DNN model using 2DPF of three features.

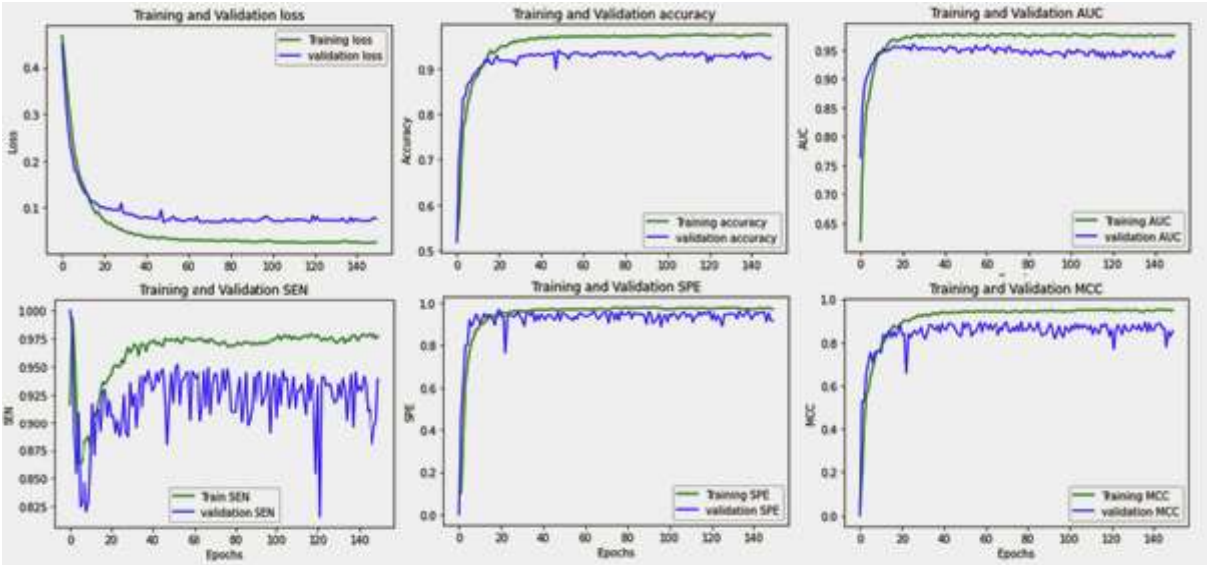


Figure 4.31. The performance evolution of the CNN model using 2DPF of three Points.

Figure 4.30 shows the progression of the DNN model's performance and demonstrates how lowering the loss function improves model performance. The training set and validation set both displayed good accuracy at epoch 87, where values of 0.9605 and 0.9185, respectively, were recorded for each epoch. Additionally, the AUC values, which measure the model's ability to distinguish between the two classes, were also high, with 0.9784 and 0.9354 for the training and validation sets, respectively. Moreover, the SEN, SPE, and MCC metrics also showed very good results for both sets during the same period, as listed in Table 4.6.

Figure 4.31, on the other hand, shows how the CNN model's performance changed as it learned from the same data after changing its representations. According to the plot, decreasing the loss function improves CNN performance. Around epoch 49, performance finally converged, with the training set obtaining a fantastic accuracy of 0.9702 and the validation set achieving an accuracy of 0.9407. It is remarkable that the CNN model outperformed the DNN model in less time, proving superior performance. Additionally, the CNN model's training and validation curve gaps were less than those of the DNN model during the evolution of the loss function. Overall, the CNN model's performance progression was more predictable than the DNN model's.

Table 4.6. The performance of DNN and CNN model using 2DPF of three points.

Model	Data Set	Accuracy	LOSS	AUC	SEN	SPE	MCC
DNN	Training Data	0.9605	**0.0320**	**0.9784**	0.9540	**0.9710**	0.922
	Validation Data	0.9185	0.0764	0.9354	0.9260	0.9448	0.8647
	Test Data	0.9250	0.0864	0.9386	0.9214	0.9285	0.8500
CNN	Training Data	**0.9702**	0.0370	0.9753	**0.9735**	0.9654	**0.9393**
	Validation Data	0.9407	0.0694	0.9512	0.9159	0.9150	0.8285
	Test Data	0.9214	0.0851	0.9480	0.9357	0.9071	0.8432

We compared the results of the suggested models with those of KNN, SVM, RF, and NB, four widely used ML techniques in this sector, in order to assess the predictive power of the models. The results of the tests we used to evaluate the models for both the CNN and DNN models were very good across all criteria. A comparison between these outcomes and those attained using the four conventional ML techniques is shown in Figure 4.32. With good accuracy (0.9214 for CNN and 0.925 for DNN), the CNN and DNN models beat the ML techniques, however the accuracy of random forest (0.91) was superior to that of the other ML models. With an accuracy score of 0.6535, NB was judged to be the least effective. With values of 0.9285, 0.9257, and 0.85 for SPE, F1-score, and MCC, respectively, the DNN model demonstrated stronger predictive values, although the CNN model had the best SEN value at 0.9357. These outcomes show that the suggested models are quite effective

We compared the accuracy of our models to three widely used deep learning techniques in the literature, RNN, GRU, and LSTM, in order to further validate the efficacy of our models. The findings are shown in Table 4.7, where it can be shown that the DNN and CNN models, with

values of 0.9250 and 0.9214, respectively, are the most accurate. The LSTM model had a reasonable accuracy score of 0.8100, while the RNN model also shown good accuracy with a value of 0.8566. The GRU model, which came in last in the rankings, received the lowest accuracy score of 0.7813. These results offer additional proof of the potency of the models we have suggested.

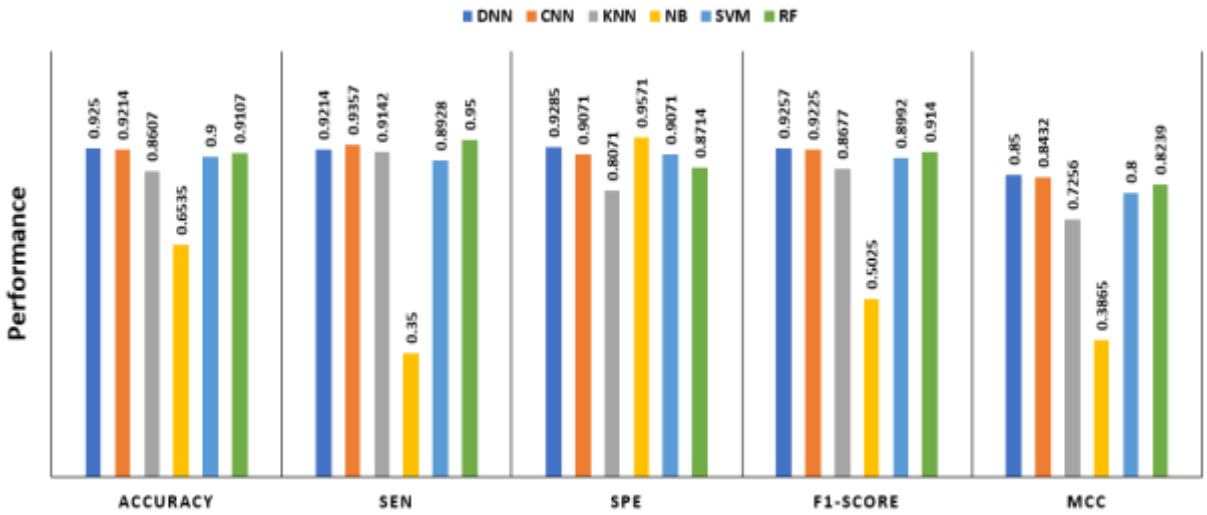


Figure 4.32.The performance comparison between the proposed models and ML methods using 2DPF of three points.

Table 4.7. Accuracy comparison between the proposed models and some deep learning methods using 2DPF of three points.

CNN	DNN	RNN	GRU	LSTM
0.9214	0.9250	0.8566	0.7813	0.810

B. 2DPF with pharmacophores of two and three points

The pharmacophores space of two points, which had a dimension of 216, and the pharmacophores space of three points, which had a dimension of 22320, were joined in the second instance. This resulted in a 2DPF that was the same size as the entire pharmacophores space we were given, which was 22536. By using the Chi-square test, we chose 1024 features, just like in the prior instance.

The DNN and CNN model performances during training and validation on the fingerprints of two and three points are shown in Figures 4.33 and Figure 4.34. The DNN model has good convergence in epoch 92, with accuracy values for training and validation data of 0.9716 and 0.9259, respectively. The AUC, SEN, SPE, and MCC values for both sets were also quite good. As the loss function shrank, the CNN model's performance increased, and by epoch 46, both sets' accuracy had converged to exceptional values (0.9639 for training and 0.9407 for validation). Both sets' AUC values (0.9652 and 0.9491) were outstanding, showing greater

discrimination. The SEN, SPE, and MCC values in the training set were more similar to those in the validation set. The values of all measures for both models are shown in Table 4.8. Similar to the prior instance, the performance evolution of the CNN model was more predictable than that of the DNN model. The evolution of the loss function for the DNN model, as opposed to the CNN model, differed significantly from that of the two curves.

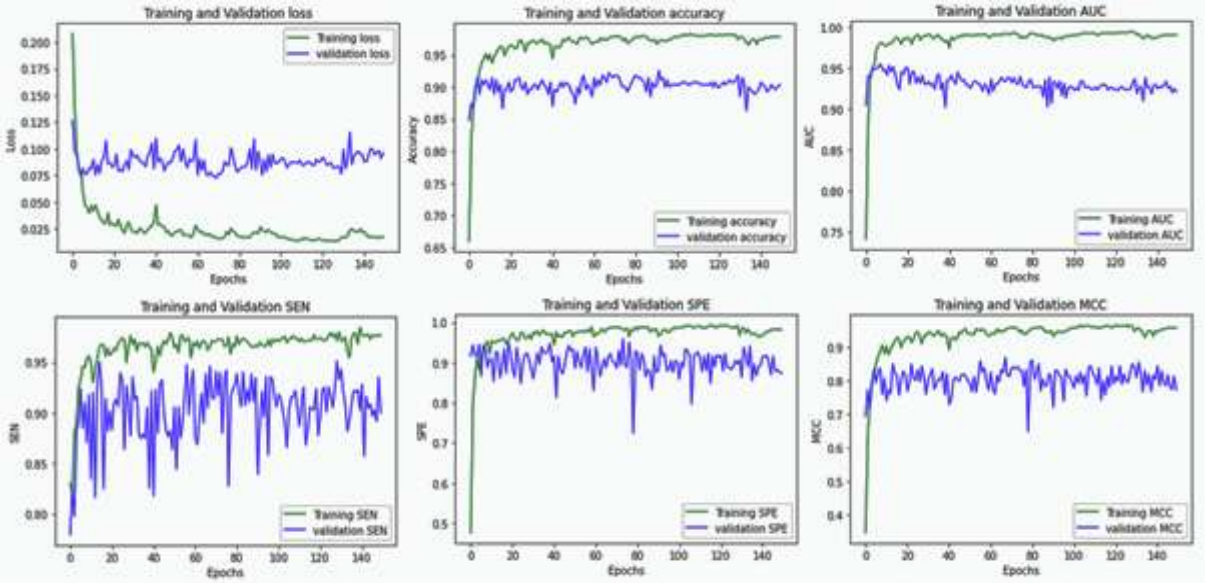


Figure 4.33. The performance evolution of DNN model using 2DPF of two and three points.

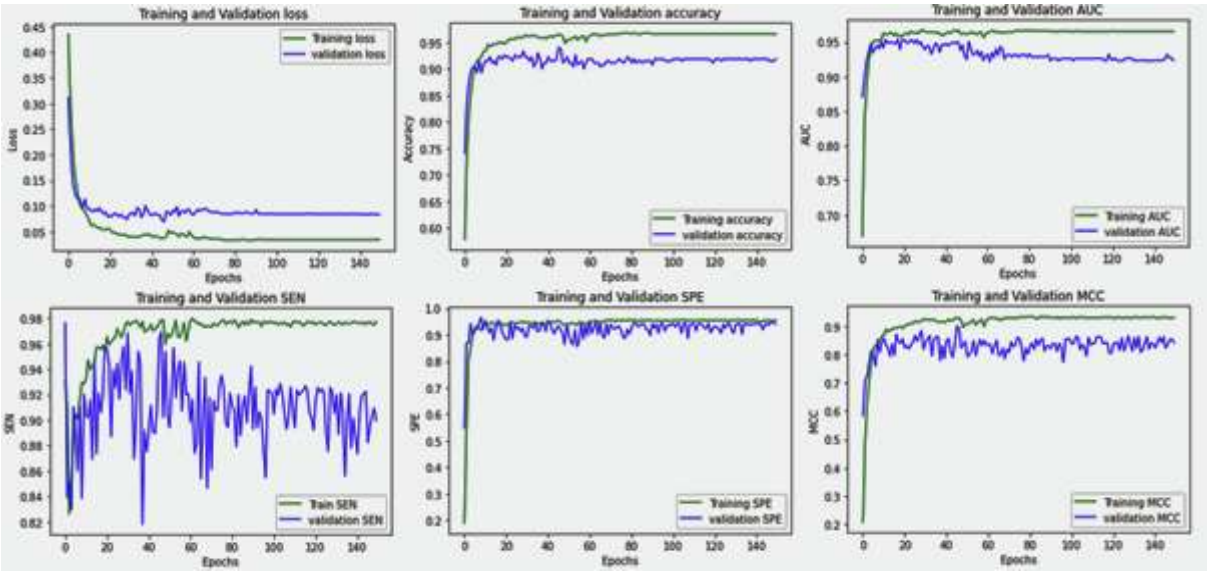


Figure 4.34. The performance evolution of the CNN model using 2DPF of two and three points.

Using four ML techniques, Figure 4.35 examines the prediction ability of our models on the test set. The outcomes demonstrate that the CNN model, which has an accuracy score of 0.9321, is the most accurate. With an accuracy score of 0.9285, the DNN model again performed admirably. The other ML models were routinely surpassed by RF, while KNN and

SVM also generated accurate results. The NB model, in comparison, got the lowest accuracy score of 0.6335, primarily because of its lower SEN score of 0.35.

Our proposed models outperformed RNN, LSTM, and GRU in terms of accuracy. Table 4.9 displays the results obtained, and similar to the previous case, CNN and DNN achieved the highest accuracy among all methods. RNN and LSTM also produced good accuracy results, while GRU was ranked last.

Table 4.8. The performance of DNN and CNN model using 2DPF of two and three points.

Model	Data Set	Accuracy	LOSS	AUC	SEN	SPE	MCC
DNN	**Training Data**	**0.9716**	**0.0240**	**0.9856**	0.9698	**0.9693**	**0.9408**
	Validation Data	0.9259	0.0757	0.9320	0.8885	0.9426	0.8310
	Test Data	0.9285	0.0721	0.9413	0.9428	0.9142	0.8574
CNN	**Training Data**	0.9639	0.0378	0.9652	**0.9761**	0.9529	0.9294
	Validation Data	0.9407	0.0710	0.9491	0.9636	0.9345	0.9003
	Test Data	0.9321	0.0732	0.9317	0.9357	0.9285	0.8643

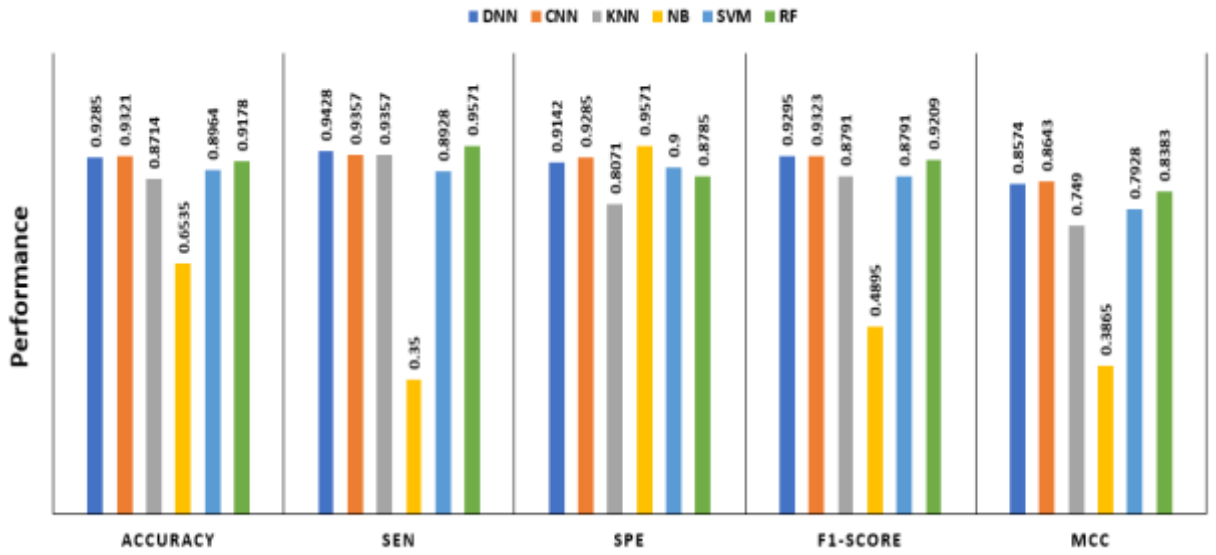


Figure 4.35. The performance comparison between the proposed models and ML methods using 2DPF of two and three points.

Table 4.9. Accuracy comparison between the proposed models and some deep learning methods using 2DPF of two and three points

CNN	DNN	RNN	GRU	LSTM
0.9321	0.9285	0.8279	0.7526	0.8064

4.4. Second proposed approach for activity prediction of molecules with CDK1 [275]

This section presents our second suggested approach for molecular activity prediction utilizing the 2DPF as a numerical descriptor. Figure 4.36 shows the flowchart of our suggested method for designing a DNN model to predict the activity or inactivity of substances. Always using the 2D structure of the molecules, we created the fingerprints based on the suggested distance ranges. Following multiple experiments, the feature count was predetermined. In order to train a DNN model and then test it for evaluation, these fingerprints were lastly inputted into the model.

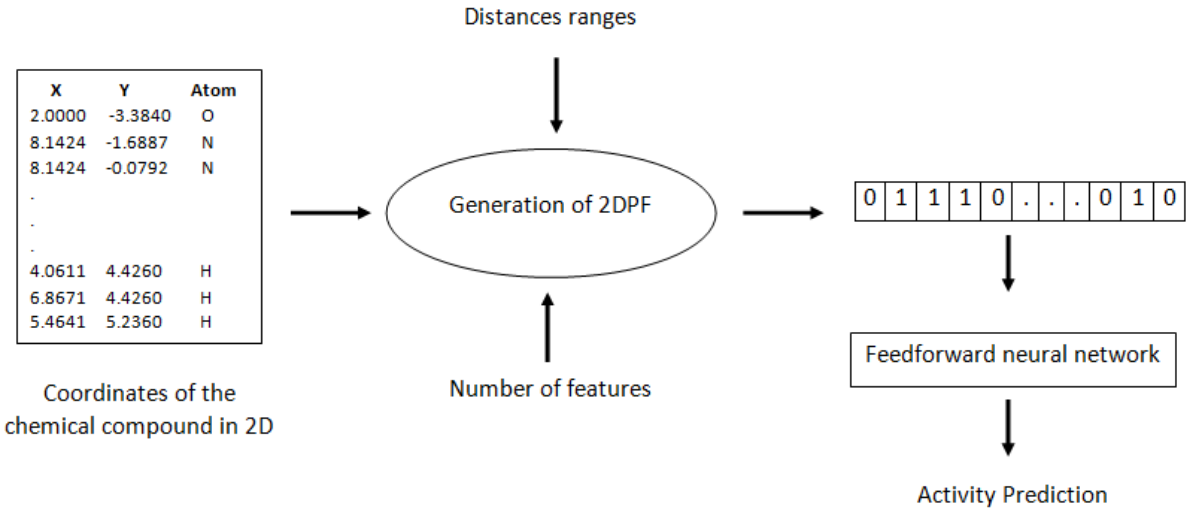


Figure 4.36. Flowchart of second proposed approach.

4.4.1. Generation of 2DPF

We generated the 2DPF in the same way as before proposition, but with a change in the distance interval and their division. In this experiment, we utilized a distance range of 0 to 5 Å and divided it into three sub-ranges: [(0-2), (2-4), (4-5)). We generated three types of 2DPF by combining features from two points and three points, respectively. We then concatenated the two feature spaces to create a mixed fingerprint.

4.4.2. The proposed predictive model

4.4.2.1. Architecture

The DNN2 model we utilized has six interconnected layers, as seen in Figure 4.37. The input layer, which is the initial layer, has 512 neurons and is where the binary vector encoding the 2DPF is sent. Three hidden layers, each with 256 neurons, are then added. The output layer, which has one neuron and 64 extra neurons, is connected to the final hidden layer. Binary vectors representing 512-bit fingerprints are the input data.

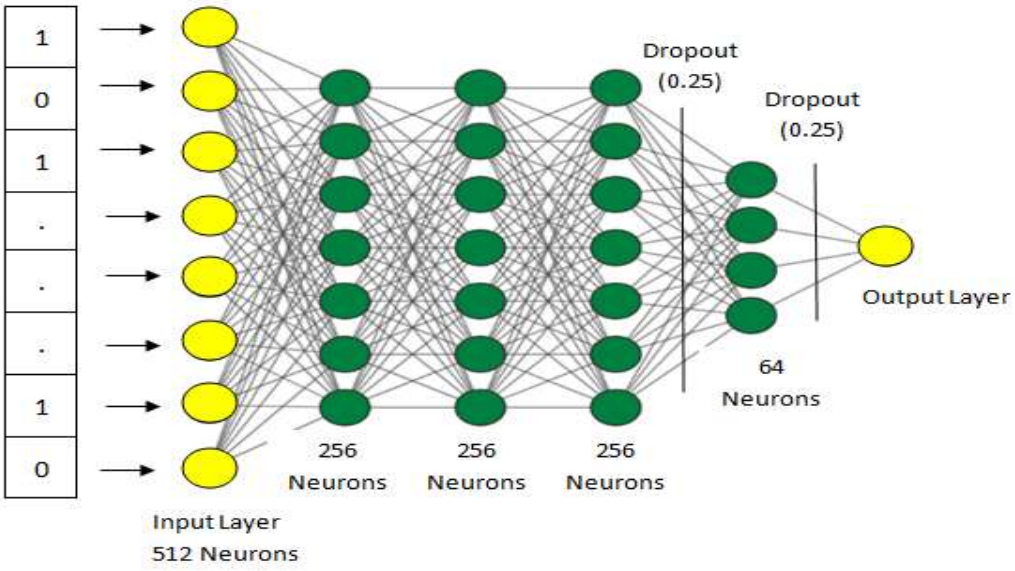


Figure 4.37. The Proposed Deep Neural Network 2 architecture.

The activation function used in all neurons except the output neuron is the ReLU function. For the output layer, we used the tanh (hyperbolic tangent) function (Figure 4.38), which is represented in Equation19

$$\text{Tanh}(x) = (e^{x} - e^{(-x)}) / (e^{x} + e^{(-x)}) \qquad \text{Eq (4.19)}$$

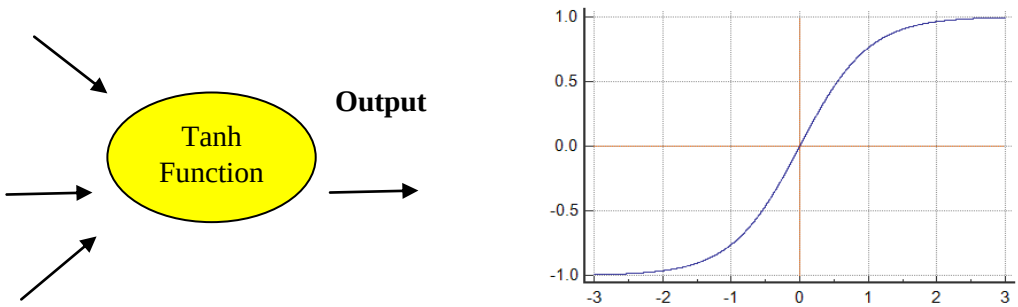


Figure 4.38. Tanh Function.

The tanh function is a commonly used activation function in deep learning that ranges from -1 to 1, with a midpoint of 0. Like other activation functions such as the sigmoid function, the

tanh function is used to introduce non-linearity into a neural network, allowing it to model complex relationships between inputs and outputs. The tanh function is often used as an alternative to the sigmoid function in neural networks, as it has a slightly steeper gradient in the regions where the inputs are far from 0, which can lead to faster convergence during training.

4.4.2.2. Hyperparameters

Table 4.10 shows the hyper parameters of DNN2 model which are fixed in training using validation data, always with ADAM as optimizer, MSE as loss function. Learning rate is fixed in 0.002.

Table 4.10. Hyperparameters setting of DNN2.

Test Split	Validation Split	Optimizer	Loss Function	Learning Rate	Dropout	Batch Size
0.2	0.2	ADAM	MSE	0.002	(0.25)	64

4.4.3. Experimental results

4.4.3.1. Data sets

In this experiment, we utilized a dataset comprising of 1259 samples, with 650 samples labeled as active and 609 as inactive on CDK1 receptor as a single protein. We split the dataset into three subsets: one subset was used for training our model, another for validating and fine-tuning our models by adjusting the hyperparameters, and the final subset was reserved for evaluating the performance of our model. The division of the dataset was done randomly, with 20% of the samples being used for testing, 20% for validation, and the remaining for training.

4.4.3.2. Overall performance

We used three different kinds of 2DPF in this study. The first type was produced by combining two points, the second type by combining three points, and the third type was produced by mixing the pharmacophores of both the first two and third types of points. We adopted an early stopping approach during the model training phase, wherein training was stopped if the loss value of the validation set stopped decreasing for 20 consecutive epochs. This prevented overfitting. We used the Tensorflow package to build the model in Python, and RDKIT to determine the 2DPF of each compound.

We used the five performance metrics Accuracy, SEN, SPE, F1-Score, and MCC to assess how well the built model performed. In order to further evaluate the efficacy of our strategy,

we also contrasted our findings with those of seven other methods: KNN, SVM, RF, NB, LSTM, RNN, and GRU.

A. 2DPF using pharmacophores of two points

Here, we combined two features with various distance ranges to create a 2DPF, which has 108 features. The evolution of the loss function served as a performance indicator as we monitored the DNN2 model's effectiveness throughout training. This development over time is depicted in Figure 4.39. The best performance was attained in epoch 72, with outstanding accuracy values of 0.9682 and 0.9008 for the training and validation sets, respectively. Additionally, we computed the AUC metric, which assesses the model's capacity to distinguish between the two classes. As indicated in Table 4.11, the AUC values for the training and validation sets were 0.9931 and 0.9296, respectively.

Figure 4.39. The Performance Evolution of DNN2 Model using 2DPF of Two Points.

We evaluated our suggested model through a series of tests to gauge its predictive power. All measures showed that the outcomes were very satisfactory. The effectiveness of our model in contrast to other approaches is shown in Table 4.12. We found that our DNN2 model had reasonable SPE values of 0.8333, F1-Score of 0.8735, and MCC of 0.7399, as well as high accuracy (0.925). These outcomes were superior to what was attained using the other techniques. The best SEN score was achieved by NB, with a score of 0.9846.

Table 4.11. The performance of DNN2 model with 2DPF using pharmacophores of two Points.

Data Set	Accuracy	LOSS	AUC
Training Data	0.9682	0.0244	0.9931
Validation Data	0.9008	0.0943	0.9296
Test Data	0.8690	0.0961	0.9451

Table 4.12. The performance comparison between DNN2 and ML Methods with 2DPF Using pharmacophores of two points.

	DNN2	SVM	NB	KNN	RF	RNN	GRU	LSTM
ACC	0.8690	0.8611	0.5714	0.8134	0.7261	0.8412	0.7579	0.8015
SEN	0.9047	0.9126	0.9846	0.8412	0.8650	0.7785	0.5928	0.7642
SPE	0.8333	0.8095	0.1269	0.7857	0.5873	0.7357	0.7714	0.6785
F1-Score	0.8735	0.8679	0.7032	0.8185	0.7595	0.7622	0.6509	0.7328
MCC	0.7399	0.72605	0.3118	0.6279	0.4709	0.5147	0.3702	0.4444

B. 2DPF using pharmacophores of three points

In this instance, a fingerprint with 3240 features was created by merging three points with different distance ranges to create a 2DPF. Then, we selected 512 of them we would utilize in our model.

Figure 4.40 depicts the development of the DNN2 model's performance during training. With great accuracy ratings of 0.9642 and 0.9008 for the training and validation sets, respectively, epoch 80 showed the best performance convergence. The AUC values were likewise quite good, with scores for the training and validation sets of 0.9897 and 0.9308, respectively. Table 4.13 displays these outcomes.

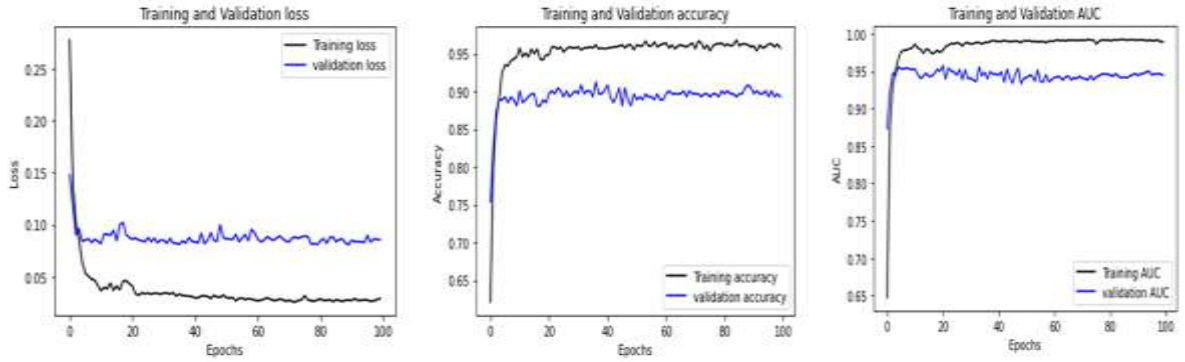


Figure 4.40.The performance evolution of DNN2 model using 2DPF of three points.

We achieved extremely excellent results for all measures after assessing our model using the 2DPF created by combining three points with different distance ranges. We contrasted the effectiveness of our DNN2 model with different methods in Table 4.14. DNN2 model outperformed the other approaches with an outstanding accuracy score of 0.9047. SVM, however, has a higher accuracy rating (0.8769) than the other ML models. RF has the lowest accuracy rating, 0.7182, making it the least effective technique.

With scores of 0.9523, 0.8571, 0.9090, and 0.8132, respectively, the DNN2 model surpassed the other models in terms of predicting SEN, SPE, F1-score, and MCC. These outcomes show how effective our DNN2 model is at predicting chemicals' bioactivity when compared to other approaches.

Table 4.13. The performance of the DNN2 model with 2DPF using pharmacophores of three points.

Data Set	Accuracy	LOSS	AUC
Training Data	0.9642	0.0277	0.9915
Validation Data	0.9008	0.0808	0.9461
Test Data	0.9047	0.0907	0.9500

Table 4.14. The performance comparison between DNN2 and ML methods with 2DPF using pharmacophores of three points.

	FNN	SVM	NB	KNN	RF	RNN	GRU	LSTM
ACC	0.9047	0.8769	0.7341	0.8492	0.7182	0.8373	0.8095	0.8174
SEN	0.9523	0.8968	0.9761	0.8650	0.9603	0.8	0.7714	0.8642
SPE	0.8571	0.8571	0.4920	0.8333	0.4761	0.7071	0.6857	0.6071
F1-Score	0.9090	0.8793	0.7859	0.8515	0.7731	0.7645	0.7397	0.7658
MCC	0.8132	0.7545	0.5351	0.6987	0.4988	0.5093	0.4588	0.4878

C. 2DPF using pharmacophores of two and three points

In the third instance, we joined the two-point pharmacophores space with a size of 108 with the three-point pharmacophores space with a dimension of 3240 to create a space with 3240 pharmacophores. The first 512 features of the acquired fingerprint were chosen.

Figure 4.41 shows the progression of the DNN2 model's performance throughout training and validation using fingerprints. With an accuracy of 0.9669 for the training data and 0.9048 for the validation data, the DNN2 model had good convergence in epoch 38. Additionally, as demonstrated in Table 4.15, the AUC values for both sets are excellent.

Table 4.16 presents a comparison of our model's predictive capabilities on the test set with other methods. The DNN2 model shows the highest accuracy with a value of 0.8809. Among the other methods, RNN performs the best, while KNN and SVM provide satisfactory predictions. In contrast, Naive Bayes shows the lowest accuracy with a value of 0.5753.

Table 4.15. The performance of the DNN2 model with 2DPF using pharmacophores of two and three Points.

Data Set	Accuracy	LOSS	AUC
Training Data	0.9722	0.0206	0.9943
Validation Data	0.9127	0.0868	0.9581
Test Data	0.8809	0.1035	0.9326

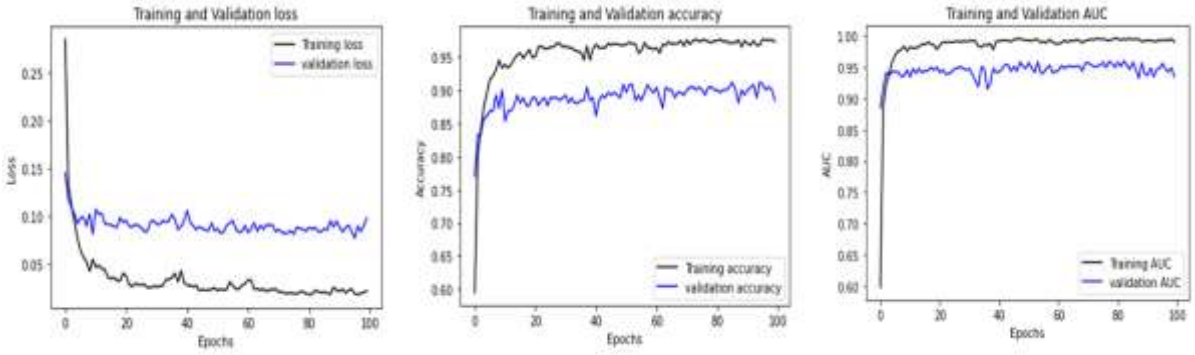


Figure 4.41. The Performance evolution of the DNN2 model using 2DPF of two and three points.

Table 4.16. The performance comparison between DNN2 and ML methods with 2DPF using pharmacophores of two and three Points.

	DNN2	SVM	NB	KNN	RF	RNN	GRU	LSTM
ACC	0.8809	0.8492	0.5753	0.8293	0.7261	0.8650	0.7817	0.7817
SEN	0.9834	0.8571	0.9769	0.8492	0.9206	0.8071	0.7928	0.7857
SPE	0.8512	0.8412	0.1428	0.8095	0.5317	0.75	0.6142	0.6214
F1-Score	0.9224	0.8503	0.7036	0.8326	0.7707	0.7847	0.7278	0.7260
MCC	0.8421	0.6985	0.3006	0.6592	0.4910	0.5580	0.4137	0.4127

4.5. Conclusion

In This chapter, we presented two approaches of VS to for activity prediction. The first approach utilized a DNN and CNN to predict the activity of a set of molecules on CDK1 receptor, taking a 2D pharmacophore fingerprint as input data. We generated all the fingerprints and used feature selection to select the most important features. We also used a manual division of the dataset based on the IC50 of molecules to create a balance in the three

subsets: training, test, and validation. The second approach utilized architecture of DNN to predict the activity of a set of molecules on same receptor. We used a predetermined number of features without using feature selection, and the division of the dataset was randomly done. The difference between the two approaches lies in architectures of models, the distance ranges used in the generation of the fingerprints, the size of the fingerprints taken to use as input data, using the feature selection technique in the first and in the second we took a fixed number of features, and the data set used. Despite these differences, both approaches yielded excellent results, which were comparable with the most famous ML methods used in this field. In summary, our proposed approaches have demonstrated the potential of deep learning using 2D pharamcophore fingerprints to predict the activity of molecules on CDK1 receptor. We believe that our findings will contribute to the development of new drugs and therapies for various diseases.

Chapter 5

Proposed Approaches Based on Deep Learning Using 3DPF for Activity Prediction

Chapter 5: Proposed Approaches Based on Deep Learning Using 3DPF for Activity Prediction

The utilization of 3D pharmacophore fingerprints (3DPF) in virtual screening is essential because it enables the identification of compounds with similar 3D patterns, enhancing the accuracy and efficiency of activity prediction in drug discovery. This chapter centers on our second contribution, which involves proposing two approaches for virtual screening to forecast the activity of molecules using 3DPF.

The chapter is structured as follows: in the second section, we discuss the significance of the 3D structure of molecules, including the notion of conformation and rotation bonds, relevant to this type of representation. In the third section of this chapter, we introduce our initial approach, which is based on a CNN model utilizing 3DPFs for activity prediction with the CDK1 receptor. We provide a comprehensive explanation of every step involved in this approach, beginning with the generation of fingerprints and concluding with the presentation of experimental results obtained. In Section Four, we introduce the second proposed approach using 3DPF for activity prediction with the BCR1 gene. We begin by explaining the crucial role of this gene in cancer, followed by a description of the proposed pharmacophore model utilized in creating the 3DPF, the proposed predictive model, and the extensive results obtained from experimental evaluations. In Section five, we apply the proposed predictive model to forecast the activity of a set of molecules with unknown activity.

5.1. Introduction

With the advent of deep learning algorithms, researchers are now able to leverage 3DPFs in virtual screening to improve the accuracy of these predictions. In drug discovery, understanding the 3D structure of molecules is critical for predicting their biological activity. The 3D arrangement of atoms and functional groups in a molecule determines its interactions with target proteins, enzymes, or other biological targets. To predict the activity of a potential drug compound, researchers often use pharmacophore models, which describe the spatial arrangement of features that are important for binding to a specific target. 3DPFs take this approach further by encoding the 3D shape and electrostatic properties of the molecule as a set of molecular descriptors. These descriptors can be used to compare the similarity between different molecules and to identify potential drug candidates with similar pharmacophore fingerprints to known active compounds. By incorporating information about the 3D structure of molecules, 3DPFs provide a powerful tool for virtual screening and activity prediction in drug discovery.

We aim in this chapter to explore the performance of using the 3DPFs with deep learning in virtual screening for activity prediction. We have chosen two targets to apply our proposition which is CDK1 and BCRA1. This latter is a tumor suppressor gene that plays a key role in DNA repair and maintenance of genome stability. Mutations in the BRCA1 gene are associated with a significantly increased risk of breast, ovarian, and other types of cancer and it has been the subject of intense research in cancer biology and drug discovery. Both approaches used the generated 3DPFs of a set of active and inactive molecules to train the proposed deep learning models. With CDK1 receptor we used the concept of conformations of molecules to generate multiple fingerprints of the same molecules. With BRCA1 gene we

have proposed a pharmacophore model to create the fingerprints which were used to learn a proposed deep neural network and fixe its parameters then we used the final model to screen a set of molecules whose activity was unknown.

5.2. Molecules in 3D space

5.2.1. 3D Structure of molecules

Molecules can exist in different shapes or conformations, which are determined by the arrangement of their constituent atoms and the bonds between them. These confirmations can be described using three coordinates, representing the position of each atom in space.

There are several methods to determine the 3D structure of a molecule, including X-ray crystallography, nuclear magnetic resonance spectroscopy, and computational modeling. X-ray crystallography involves crystallizing the molecule and using X-rays to determine the positions of its atoms in space. NMR spectroscopy uses magnetic fields and radio waves to determine the positions and interactions of atoms within a molecule. Computational modeling uses software to calculate the positions of atoms based on their electronic properties and the rules of chemical bonding. Once the 3D structure of a molecule is known, it can be analyzed and visualized in various ways. One common method is to use ball-and-stick models, which represent atoms as balls and bonds as sticks. Another method is to use space-filling models, which represent atoms as spheres proportional to their size and show the overall shape of the molecule. The 3D structure of a molecule is important in many fields, including drug discovery, materials science, and biochemistry, where it is used to understand the function and properties of proteins, DNA, and other biomolecules. In drug discovery the importance of 3D structure of bioactive chemical compound is it contribution in determining the strength and selectivity of their protein–ligand interactions.

Figure 5.1 shows a representation of aspirin (C9H8O4) in 3D space and the coordinates of its atoms in space (X, Y, Z) are shown in Table 5.1

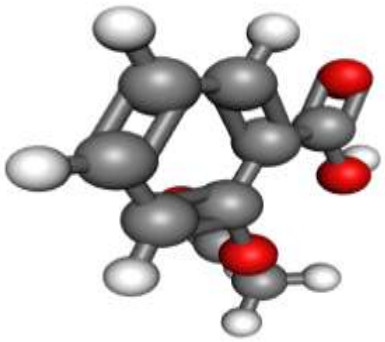


Figure 5.1. 3D Representation of aspirin.

5.2.2 Conformations of molecules

5.2.2.1. Conformation concept

A molecule can adopt different shapes or conformations, which represent different arrangements of its constituent atoms in space. These conformations can be changed by rotation around single bonds in the molecule. Each distinct conformation represents a different potential energy state, which is determined by the balance of attractive and repulsive forces between different parts of the structure. We refer to each possible conformation as a conformer. In theory, a molecule could take an infinite number of conformations, but in practice, some conformations are more stable or energetically favorable than others. The relative stability of different conformations is influenced by factors such as steric hindrance, electrostatic interactions, and hydrogen bonding [282,283].

An example of a molecule with multiple conformations is ethane (CH3CH3), which has a single carbon-carbon bond that can rotate freely. As shown in Figure 5.2, the molecule can adopt different conformations depending on the angle of rotation around this bond. The staggered conformation is the most stable, as it minimizes steric hindrance between the hydrogen atoms on adjacent carbon atoms. The eclipsed conformation, in which the hydrogen atoms are directly aligned with each other, is less stable due to increased repulsive forces.

Understanding the different conformations that a molecule can adopt is essential for predicting its properties and interactions with other molecules. In drug discovery, for example, researchers must consider the different conformations of potential drug candidates and their influence on the molecule's ability to bind to target proteins or enzymes.

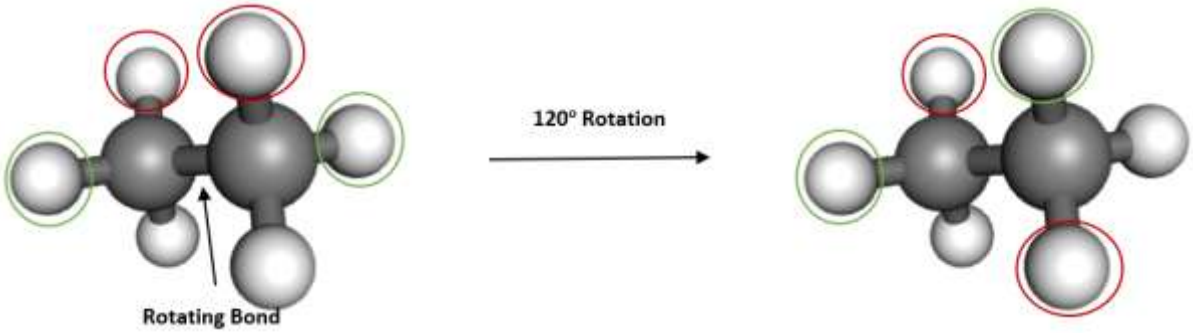


Figure 5.2. Ethane rotation .

The X, Y, and Z coordinates of Aspirin's atoms in three-dimensional space are shown in Table 1 for three conformations. We notice a difference of atoms coordinates in the three cases where each conformation results of rotations around single bonds. This difference in the coordinates leads to the change in the distance between the atoms, and thus the difference in 3D pharmacophore fingerprint extracted from each conformation.

5.2.2.2. Rotation bonds

Rotation bonds, also known as single bonds or sigma bonds are chemical bonds between two atoms that allow free rotation around the bond axis. In a rotation bond, the two atoms are held together by a covalent bond, which involves the sharing of electrons between the atoms. This type of bond is characterized by the overlap of orbitals between the two atoms along the bond axis, forming a single, symmetric bond. The ability of a bond to rotate freely is determined by the energy required to overcome the potential energy barriers between different conformations. In the case of rotation bonds, the potential energy barriers are caused by steric hindrance, electrostatic interactions, and other factors that affect the stability of different conformations. As a result, some rotation bonds may have restrictions on their rotation, or they may only rotate under certain conditions. Rotation bonds play an important role in the structure and function of organic molecules, as they allow for the flexibility and conformational changes that are necessary for many biological processes. For example, in proteins, rotation bonds allow for the folding and unfolding of the protein structure, as well as the movement of different parts of the protein during enzymatic reactions or other biological interactions [284,285].

In drug discovery, the ability of a potential drug compound to adopt different conformations is important for predicting its interactions with target proteins or enzymes. By understanding the different conformations that a molecule can adopt and the potential energy barriers between them, researchers can design drugs that are more likely to bind to their target and have the desired biological activity . The most important rotation bonds in organic chemistry include:

• **Carbon-carbon (C-C) single bonds:** These are the most common rotation bonds in organic molecules and are found in all alkanes, as well as in many other functional groups such as ethers, amines, and amides. The ability of the C-C bond to rotate freely is essential for the conformational flexibility of many organic molecules.

• **Carbon-oxygen (C-O) single bonds:** These bonds are found in many functional groups such as alcohols, ethers, and carbonyl compounds. The rotation of the C-O bond can affect the stereochemistry and reactivity of these functional groups.

• **Carbon-nitrogen (C-N) single bonds:** These bonds are found in many important functional groups such as amines, amides, and nitriles. The rotation of the C-N bond can also affect the stereochemistry and reactivity of these functional groups.

Table 5.1. Three-dimensional coordinates of three different conformations of aspirin.

First conformation				Second conformation			Third conformation		
Atom	**X**	**Y**	**Z**	**X**	**Y**	**Z**	**X**	**Y**	**Z**
O	1.2333	0.5540	0.7792	1.2670	0.3852	-0.8565	1.1549	-0.7284	-0.2991
O	-0.6952	-2.7148	-0.7502	-0.9410	-2.6646	0.7430	-1.8687	-2.0527	-1.0750
O	0.7958	-2.1843	0.8685	0.5126	2.2842	-0.9501	-1.5916	-2.1805	1.1680
O	1.7813	0.8105	-1.4821	2.5628	-0.5976	0.8272	2.6382	0.5077	1.0244

C	-0.0857	0.6088	0.4403	-0.0182	0.5849	-0.4489	0.1891	0.2273	-0.1900
C	-0.7927	-0.5515	0.1244	-0.8365	-0.4919	-0.1071	-1.1441	-0.1232	0.0232
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.
H	4.2045	0.6969	-0.6924	3.7322	1.7157	0.8212	4.4731	-0.7142	-0.2592
H	3.7105	-0.3659	0.6426	2.0743	2.2955	1.1335	3.4132	-0.7883	-1.6828
H	-0.2555	-3.5916	-0.7337	-0.6062	-3.5854	0.6953	-2.1361	-2.9935	-0.9987

5.3. First proposed approach for activity prediction with CDK1 using 3DPF

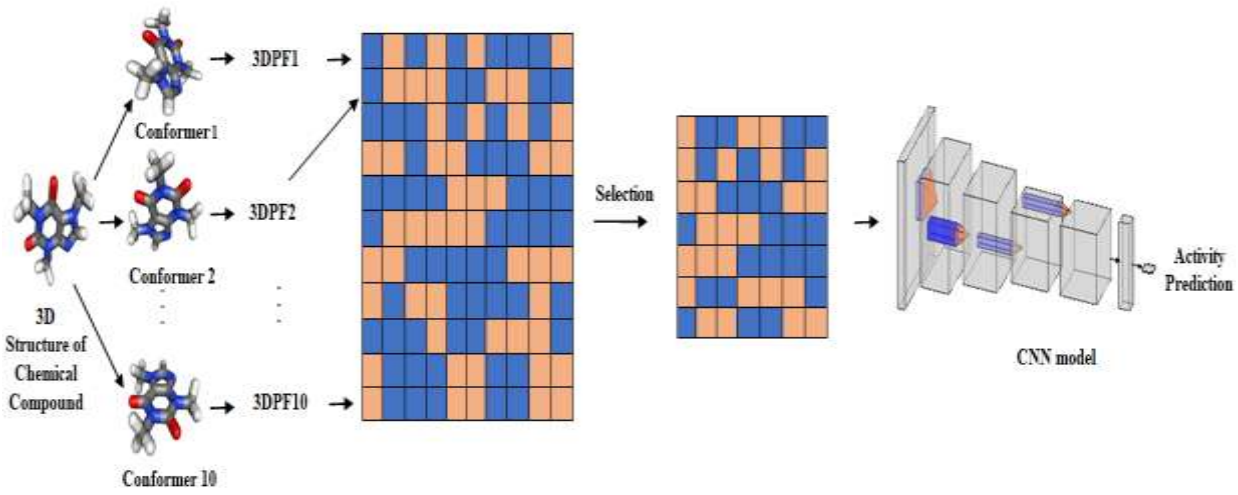


Figure 5.3. The proposed approach for activity prediction of molecules with CDK1.

In this approach we use CNN with 3D pharmacophore fingerprints (3DPF) for the prediction of biological activity with the CDK1 receptor (Figure 5.3). We generated 10 conformations of the molecule under consideration where each conformation represents a different potential energy state, representing different possible orientations of the atoms and functional groups. Each of these conformations is then used to generate a unique 3DPF. These fingerprints are then fed into a CNN, which is trained to predict the biological activity of the molecules with CDK1 receptor.

5.3.1 Generation of 3D pharmacophore fingerprint

To generate a 3DPF, a similar process is followed to that of generating a 2DPF, which involves two main steps: creating the pharmacophore space and generating the fingerprint (Figure 5.4). In our approach, we utilized the distance parameter of the PMAPPER library, which calculates the distance between pharmacophore features. We generated the 3DPF for

each conformation by encoding the pharmacophore space into a binary fingerprint, which represents the presence or absence of each pharmacophore feature as a bit in a binary string.

The advantage of a 3D pharmacophore fingerprint is that it provides a quantitative representation of the 3D structure of a molecule that can be used for similarity searching and virtual screening. Similar molecules can be identified by comparing their 3D pharmacophore fingerprints, and compounds that have a similar fingerprint to a known active compound are likely to have similar biological activity. 3DPF are also useful for identifying key pharmacophoric features that are important for binding to a specific target. By comparing the fingerprints of active and inactive compounds, it is possible to identify the key pharmacophoric features that are necessary for activity, and these features can be used to guide the design of new compounds with improved potency and selectivity.

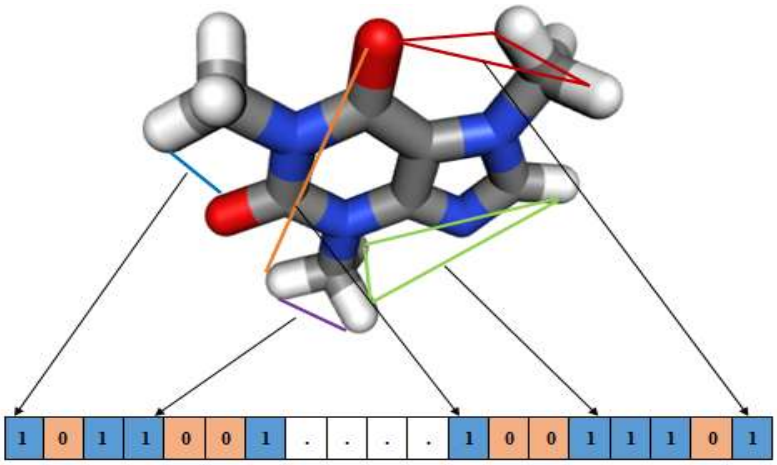


Figure 5.4. Generation of 3D pharmacophore fingerprint.

5.3.2. Selection of discriminating pharmacophores

After calculating all the fingerprints representing the different molecule conformations, we got a huge size of the overall fingerprint that represents the chemical compound. It is therefore necessary to extract the discriminating pharmacophores in order to reformulate a new fingerprint containing except the effective pharmacophores to distinguish the active and inactive ligands. We used a statistical method that relies on calculating the difference between the groups by comparing the mean values of the data from each group called analysis of variance (ANOVA). This method helped us to select the best pharmacohpres (Figure 5.5); its formula is represented as following:

$$F = (SSB / dfB) / (SSW / dfW) \quad \text{Eq (5.1)}$$

Where F is the F-statistic, SSB is the between-group sum of squares, dfB is the between-group degrees of freedom, SSW is the within-group sum of squares, and dfW is the within-

group degrees of freedom. This formula allowed us to determine which pharmacophores were most important for distinguishing between active and inactive ligands, and we used this information to create a more efficient and effective fingerprint for our deep neural network to predict the biological activity of potential drug candidates.

The application the ANOVA method for feature selection in our study is illustrated as follows: Table 5.2 shows the presence or absence of each phramcophore in molecules.

A. Calculate the mean value of each feature for the active and inactive classes

We must compute the mean value of each feature for the active and inactive calsses separately, the calculation for Feature 1 for 4 molecules as follows:

Active Class: Mean value = (1+0)/2 = 0.5

Inactive Class: Mean value: (1+1)/2 = 1

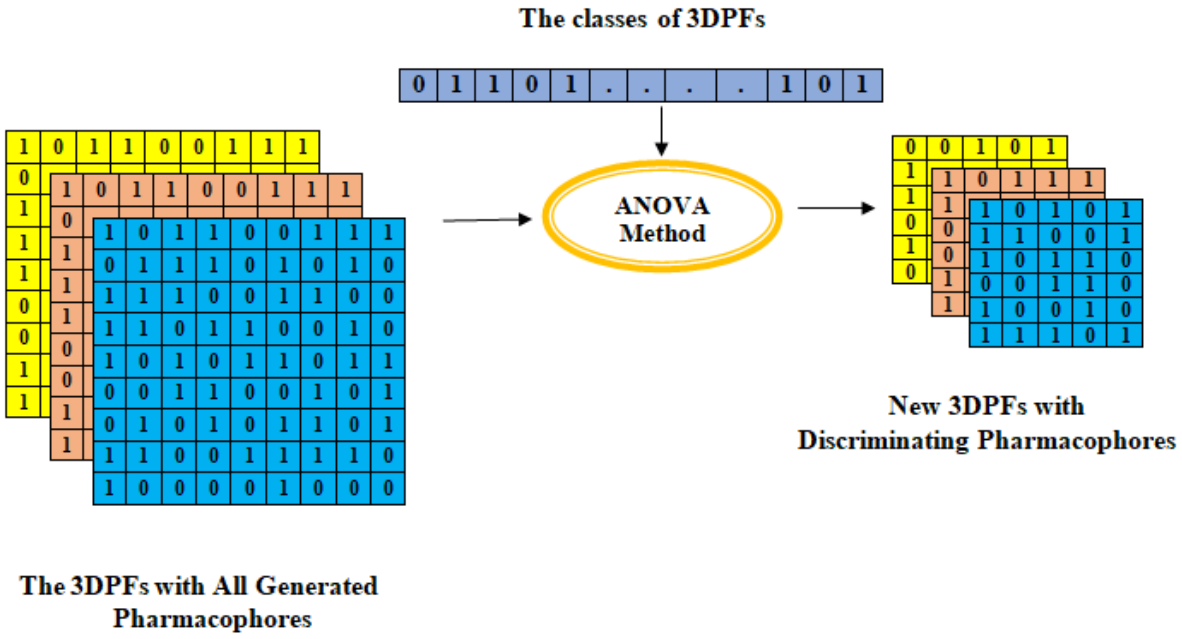


Figure 5.5. Selection of discriminating pharmacophores with ANOVA method.

B. Calculate the variance of each feature for the active and inactive classes

o calculate the variance of each feature for the active and inactive classes separately, we can use the following formula:

$$Var = (\Sigma(xi - \mu)^2) / (n - 1) \quad \text{Eq (5.2)}$$

Where xi is the value of the feature for a particular molecule, μ is the mean value of the feature for the class, and n is the number of molecules in the class. To calculate the variance of Feature1 for the active and inactive classes for 4 molecules:

Active class: Var = $(1-0.5)^2 + (0-0.5)^2 / (2-1) = 0.5$

Inactive class: Var = $(1-1)^2 + (1-1)^2 / (2-1) = 0$

We repeat this process for each of the features to obtain the variance of each feature for the active and inactive classes.

Table 5.2. Presence/Absence of Pharmacophores in Molecules.

Molecule	Pharm 1	Pharm 2	Pharm 3	.	.	Pharm n	Class
1	1	1	1	.	.	.	Active
2	0	1	1	.	.	.	Active
3	1	0	1	.	.	.	Inactive
4	1	1	0	.	.	.	Inactive
.	.	.	.	.	.	.	.
n	.	.	.	.	.	.	.

D. Calculate the F-value for each feature using the formula:

F-value = (variance between classes) / (variance within classes) Eq (5.3)

Where: The variance between groups is the variance of the means of each group. The variance within groups is the average of the variances of each classes.

F-value of Feature 1 with 4 molecules is calculated as follows:

Grand mean = (1 + 0 + 1 + 1) / 4 = 0.75

Variance between classes = $[(0.5 - 0.75)^2 + (1.0 - 0.75)^2] / (2 - 1) = 0.125$

Variance within classes = (0.5 + 0) / 2 = 0.25

F-value = 0.125 / 0.25 = 0.5

E. Determine the significance level (p-value) associated with each F-value using the F-distribution table or statistical software.

The null hypothesis is that the means of the feature in the active and inactive groups are equal for determining the F-value for each feature. They are not equal, according to the alternative theory. This hypothesis is tested using the F-value, and the p-value is the likelihood that, if the null hypothesis is correct, an F-value as extreme as or more extreme than the computed one

will be observed. Using the F-distribution table (Figure 5.6), one may find the p-value connected to each F-value. It displays the F-distribution's crucial values for a certain significance level and degree of freedom. To decide whether to accept or reject the null hypothesis, the estimated F-value can be compared to the critical value from the table. Given the F-value and degrees of freedom, statistical tools can also be used to determine the p-value directly. The null hypothesis can be disregarded if the p-value is smaller than the significance level (). The null hypothesis cannot be ruled out if the p-value is larger than or equal to.

	F-table of Critical Values of α = 0.10 for F(df1, df2)																		
	DF1=1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	120	∞
DF2=1	39.86	49.50	53.59	55.83	57.24	58.20	58.91	59.44	59.86	60.19	60.71	61.22	61.74	62.00	62.26	62.53	62.79	63.06	63.33
2	8.53	9.00	9.16	9.24	9.29	9.33	9.35	9.37	9.38	9.39	9.41	9.42	9.44	9.45	9.46	9.47	9.47	9.48	9.49
3	5.54	5.46	5.39	5.34	5.31	5.28	5.27	5.25	5.24	5.23	5.22	5.20	5.18	5.18	5.17	5.16	5.15	5.14	5.13
4	4.54	4.32	4.19	4.11	4.05	4.01	3.98	3.95	3.94	3.92	3.90	3.87	3.84	3.83	3.82	3.80	3.79	3.78	3.76
5	4.06	3.78	3.62	3.52	3.45	3.40	3.37	3.34	3.32	3.30	3.27	3.24	3.21	3.19	3.17	3.16	3.14	3.12	3.11
6	3.78	3.46	3.29	3.18	3.11	3.05	3.01	2.98	2.96	2.94	2.90	2.87	2.84	2.82	2.80	2.78	2.76	2.74	2.72
7	3.59	3.26	3.07	2.96	2.88	2.83	2.78	2.75	2.72	2.70	2.67	2.63	2.59	2.58	2.56	2.54	2.51	2.49	2.47
8	3.46	3.11	2.92	2.81	2.73	2.67	2.62	2.59	2.56	2.54	2.50	2.46	2.42	2.40	2.38	2.36	2.34	2.32	2.29
9	3.36	3.01	2.81	2.69	2.61	2.55	2.51	2.47	2.44	2.42	2.38	2.34	2.30	2.28	2.25	2.23	2.21	2.18	2.16
10	3.29	2.92	2.73	2.61	2.52	2.46	2.41	2.38	2.35	2.32	2.28	2.24	2.20	2.18	2.16	2.13	2.11	2.08	2.06
11	3.23	2.86	2.66	2.54	2.45	2.39	2.34	2.30	2.27	2.25	2.21	2.17	2.12	2.10	2.08	2.05	2.03	2.00	1.97
12	3.18	2.81	2.61	2.48	2.39	2.33	2.28	2.24	2.21	2.19	2.15	2.10	2.06	2.04	2.01	1.99	1.96	1.93	1.90
13	3.14	2.76	2.56	2.43	2.35	2.28	2.23	2.20	2.16	2.14	2.10	2.05	2.01	1.98	1.96	1.93	1.90	1.88	1.85
14	3.10	2.73	2.52	2.39	2.31	2.24	2.19	2.15	2.12	2.10	2.05	2.01	1.96	1.94	1.91	1.89	1.86	1.83	1.80
15	3.07	2.70	2.49	2.36	2.27	2.21	2.16	2.12	2.09	2.06	2.02	1.97	1.92	1.90	1.87	1.85	1.82	1.79	1.76
16	3.05	2.67	2.46	2.33	2.24	2.18	2.13	2.09	2.06	2.03	1.99	1.94	1.89	1.87	1.84	1.81	1.78	1.75	1.72
17	3.03	2.64	2.44	2.31	2.22	2.15	2.10	2.06	2.03	2.00	1.96	1.91	1.86	1.84	1.81	1.78	1.75	1.72	1.69
18	3.01	2.62	2.42	2.29	2.20	2.13	2.08	2.04	2.00	1.98	1.93	1.89	1.84	1.81	1.78	1.75	1.72	1.69	1.66
19	2.99	2.61	2.40	2.27	2.18	2.11	2.06	2.02	1.98	1.96	1.91	1.86	1.81	1.79	1.76	1.73	1.70	1.67	1.63
20	2.97	2.59	2.38	2.25	2.16	2.09	2.04	2.00	1.96	1.94	1.89	1.84	1.79	1.77	1.74	1.71	1.68	1.64	1.61
21	2.96	2.57	2.36	2.23	2.14	2.08	2.02	1.98	1.95	1.92	1.87	1.83	1.78	1.75	1.72	1.69	1.66	1.62	1.59
22	2.95	2.56	2.35	2.22	2.13	2.06	2.01	1.97	1.93	1.90	1.86	1.81	1.76	1.73	1.70	1.67	1.64	1.60	1.57
23	2.94	2.55	2.34	2.21	2.11	2.05	1.99	1.95	1.92	1.89	1.84	1.80	1.74	1.72	1.69	1.66	1.62	1.59	1.55
24	2.93	2.54	2.33	2.19	2.10	2.04	1.98	1.94	1.91	1.88	1.83	1.78	1.73	1.70	1.67	1.64	1.61	1.57	1.53
25	2.92	2.53	2.32	2.18	2.09	2.02	1.97	1.93	1.89	1.87	1.82	1.77	1.72	1.69	1.66	1.63	1.59	1.56	1.52
26	2.91	2.52	2.31	2.17	2.08	2.01	1.96	1.92	1.88	1.86	1.81	1.76	1.71	1.68	1.65	1.61	1.58	1.54	1.50
27	2.90	2.51	2.30	2.17	2.07	2.00	1.95	1.91	1.87	1.85	1.80	1.75	1.70	1.67	1.64	1.60	1.57	1.53	1.49
28	2.89	2.50	2.29	2.16	2.06	2.00	1.94	1.90	1.87	1.84	1.79	1.74	1.69	1.66	1.63	1.59	1.56	1.52	1.48
29	2.89	2.50	2.28	2.15	2.06	1.99	1.93	1.89	1.86	1.83	1.78	1.73	1.68	1.65	1.62	1.58	1.55	1.51	1.47
30	2.88	2.49	2.28	2.14	2.05	1.98	1.93	1.88	1.85	1.82	1.77	1.72	1.67	1.64	1.61	1.57	1.54	1.50	1.46
40	2.84	2.44	2.23	2.09	2.00	1.93	1.87	1.83	1.79	1.76	1.71	1.66	1.61	1.57	1.54	1.51	1.47	1.42	1.38
60	2.79	2.39	2.18	2.04	1.95	1.87	1.82	1.77	1.74	1.71	1.66	1.60	1.54	1.51	1.48	1.44	1.40	1.35	1.29
120	2.75	2.35	2.13	1.99	1.90	1.82	1.77	1.72	1.68	1.65	1.60	1.55	1.48	1.45	1.41	1.37	1.32	1.26	1.19
∞	2.71	2.30	2.08	1.94	1.85	1.77	1.72	1.67	1.63	1.60	1.55	1.49	1.42	1.38	1.34	1.30	1.24	1.17	1.00

Figure 5.6. F Distribution table [276].

F. Select the features with the lowest p-values as the most significant discriminating pharmacophores.

A specific number of features has already been predetermined. We can arrange the features in ascending order according to their p-values and select the top k features, where k is the predetermined number of features we wish to choose. The significance threshold for the p-value will be determined by the rank of the k-th feature. Any feature with a p-value less than or equal to this threshold will be deemed significant and chosen.

5.3.3. CNN architecture

In our work we use 10 conformations of molecules and we extract the 3D pharmacophore fingerprint of each conformation with size of 2048 bits. So, the chemical compound is represented by a binary matrix [2048*10] .We use ANOVA method the select the best

pharmacophores in order to minimize the size of fingerprint .So the new size is 4096 bits that reshape in matrix with size of [64*64] which used such an input data of the constructed model in order to predict the activity or inactivity of compound.

The suggested CNN2 model consisted of only one convolution layer with 512 filters of size (21*21) and one max-pooling layer with a pooling size of (21*21), as shown in Figure 5.7. The fully linked network consists of one layer with 512 neurons and one neuron in the output layer. The Sigmoid activation is used in the output layer, and RELU is used as the activation function in the other layers. With a learning rate of 0.001, MAE (mean absolute error) serves as the loss function, and the batch size is 64 samples.

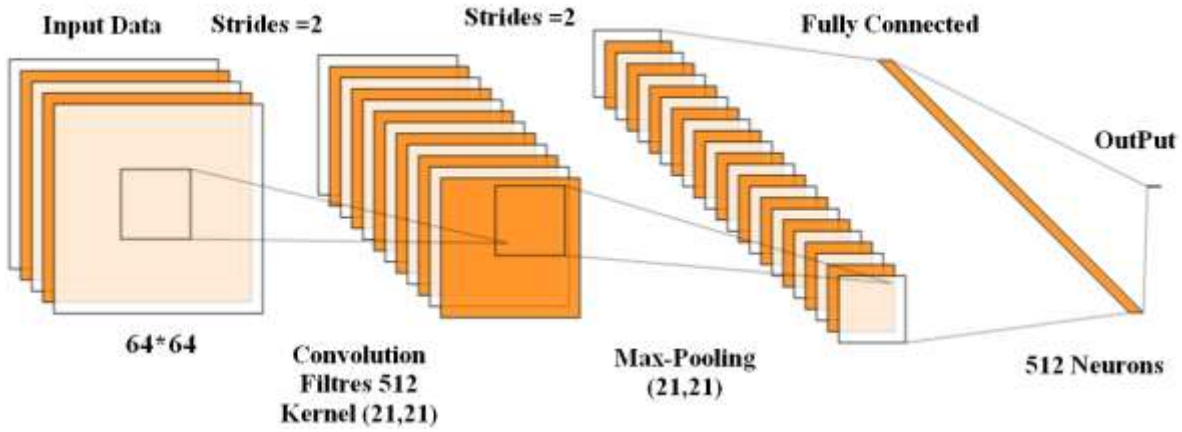


Figure 5.7. CNN2 Architecture for activity prediction of molecules with CDK1 using 3DPF.

5.3.4. Experimental results

5.3.4.1. Data sets

1360 samples total (680 active and 680 inactive) were used in our experiment, and they were split into two subsets: a set for training the model and a set for testing so that we could assess our models and adjust the hyperparameters.. The dataset was randomly divided, where we used 20% of it was used to test our model and the rest are used for training. The CHEMBL Database is used in this work to extract the active and inactive ligands with CDK1. The three-dimensional structure of all compounds is obtained from the PubChem database.

5.3.4.2. Overall performance

We have used the generated 3DPF which contains a set of pharmacophores of three features. Python was used to build the CNN2 model, which used Tensorflow and Keras libraries. We used the Pmapper and Rdkit libraries to determine each compound's 3DPF. We used six performance metrics which are: Accuracy (ACC), SEN, SPE, F1-Score, MCC, and AUC.

Figure 5.8 represents the performance evolution of the proposed CNN2 model. As mentioned before, we used a subset of active and inactive ligands containing 20% of the overall set to validate and test the model. The best performance implies a low value of the loss function. According to the evolution of the loss function, we notice that at epoch 76 there is a common

stability of the error value of the training set at the value 0.0055 and of the test set at the value 0.0880. At epoch 87, with values of 0.9118 for the test set and 0.9945 for the training set, the two sets achieved the highest accuracy convergence. For both sets during the same time period, the values of the other measures (SEN, SPE, F1-Score, and MCC) are very good (see Table 5.3).

Figure 5.9 shows the evolution of the performances of the AUC which measures the capacity of the model to distinguish the two classes. The training set reached an excellent AUC value at epoch 72 with a value of 0.9945 and it remained at this value until the end but the best convergence between the two sets at epoch 87 with the value of 0.9286 for the test set. The evolution of the AUC of the test set shows a very good distinction between active and inactive molecules.

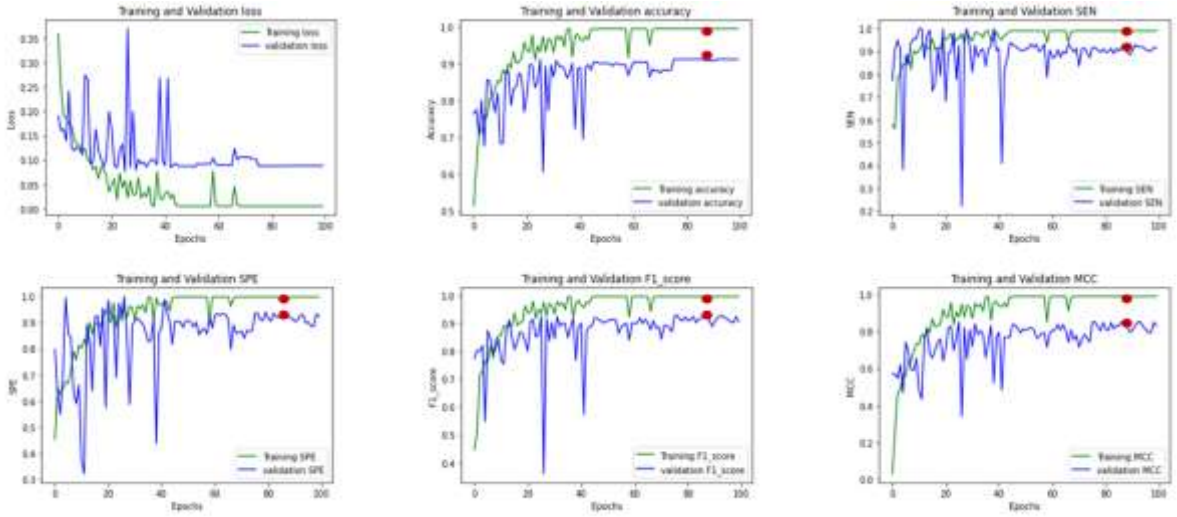


Figure 5.8. The Performance Evolution of the CNN2 Model using 3DPF.

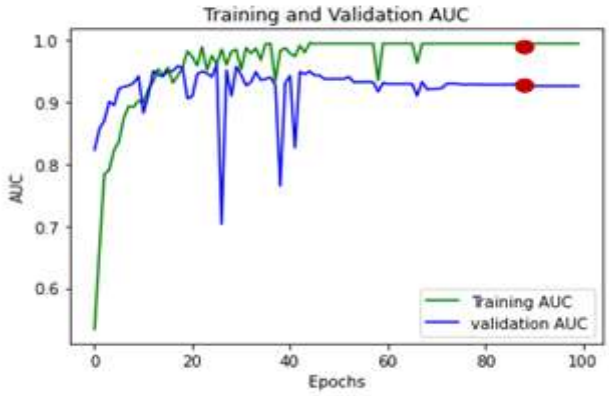


Figure 5.9. The Performance evolution of AUC.

To prove the performance of generated fingerprints and the proposed predictive model the results have been compared with the most methods of ML and some famous methods of deep learning which are: SVM, KNN, RF, NB, DT, Linear Discriminant Analysis (LDA), Quadratic discriminant analysis (QDA), Nearest shrunken centroids(NSC), Partial Least

Squares Regression (PLSR), Neural networks (NNET), RNN, GRU, and LSTM. Table 5.4 represents the obtained results of each method. We notice that the proposed CNN2 model with the generated fingerprints gives the best accuracy of 0.9118 and the best values with all metrics. KNN with K=3 and SVM give good accuracy with equal values of 0.8860 and convergent values of all metrics. The worst accuracy is given by PLSR and QDA with values of 0.4043 and 0.4485 respectively. The three methods of deep learning: RNN, GRU, and LSTM give reasonable values with all metrics, neither the best nor the worst despite their famous effectiveness.

Table 5.3. The performance CNN2 model using 3DPF.

Data Set	LOSS	ACC	AUC	SEN	SPE	F1-Score	MCC
Training Set	0.0055	0.9945	0.9945	0.9908	0.9976	0.9945	0.9890
Test Set	0.0881	0.9118	0.9286	0.9202	0.9255	0.9256	0.8528

Table 5.4. The comparison of the CNN2 model's performance to that of other methods using 3DPF.

	ACC	SEN	SPE	F1-Score	MCC
CNN2	**0.9118**	**0.9163**	**0.9234**	**0.9045**	**0.8289**
SVM	0.8860	0.9044	0.8676	0.8880	0.7725
KNN	0.8860	0.9191	0.8529	0.8896	0.7737
RF	0.8639	0.9338	0.7941	0.8728	0.7351
NB	0.7242	0.7647	0.6838	0.7349	0.4500
DT	0.7500	0.7794	0.7205	0.7571	0.5008
LDA	0.8014	0.75735	0.8455	0.79230	0.6053
QDA	0.4485	0.2867	0.6102	0.3421	0.1087
NSC	0.7316	0.2867	0.6102	0.3421	-0.1087
PLSR	0.4043	0.8529	0.7500	0.8111	0.6061
NNET	0.8198	0.7867	0.8529	0.8136	0.6411
RNN	0.7463	0.7424	0.7913	0.7614	0.5377
GRU	0.6801	0.5359	0.8424	0.6231	0.3995
LSTM	0.6618	0.5512	0.7481	0.5901	0.3072

5.4. Second proposed approach for activity prediction with BRCA1

5.4.1. Breast cancer gene (BRCA1)

Millions of people throughout the world are impacted by the dangerous and perhaps fatal disease known as breast cancer. With an estimated 2.3 million new cases expected to be detected in 2020 alone, breast cancer is the most prevalent malignancy in women overall, according to the World Health Organization. Researchers may be able to improve outcomes for breast cancer patients by continuing to examine the function of BRCA1 in this disease and creating novel medications that target its activity.

5.4.1.1. BRCA1 gene

BRCA1 is a tumor suppressor gene that plays a crucial role in maintaining the stability of the cell's genetic material (DNA). BRCA1 mutations are thought to contribute to cancer development by disrupting the cell's ability to repair DNA damage (Figure 5.10) which can lead to the accumulation of mutations and the formation of abnormal cells. In particular, BRCA1 is involved in the repair of double-strand breaks in DNA, a type of DNA damage that is particularly difficult to repair. BRCA1 mutations can lead to defects in this repair pathway, which in turn can increase the risk of cancer, particularly breast and ovarian cancer. So BRCA1 inhibition is an important area of research and drug development. Inhibiting its activity could potentially stop the growth and spread of breast cancer cells [277].

.

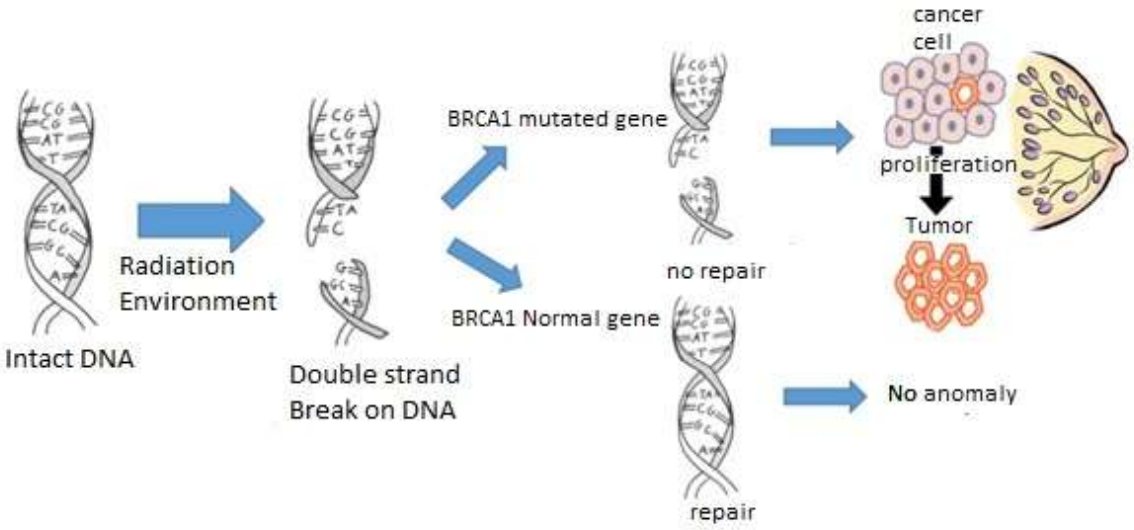


Figure 5.10. BCRA1 contribute to cancer [277].

5.4.1.2. Targeting BRCA1 gene for drug discovery

Targeting BRCA1 for drug discovery is an active area of research in the field of cancer therapeutics. Several approaches have been explored to develop drugs that target BRCA1 and its associated pathways. Here are some examples:

• **Inhibiting protein-protein interactions involving BRCA1:** Small molecules have been developed that inhibit the interaction between BRCA1 and BARD1, a protein that plays a role

in DNA repair. Inhibition of this interaction disrupts the function of BRCA1 and sensitizes cancer cells to DNA damage, which can lead to cell death [286].

• **Gene therapy:** Methods have been developed for delivering functional copies of the BRCA1 gene to cells with mutated BRCA1. Viral vectors can be used to deliver the functional BRCA1 gene to cells. This approach aims to restore the function of the mutated BRCA1 gene and has shown promise in preclinical studies [287].

• **Targeting other proteins and pathways involved in BRCA1-associated cancers:** Drugs that target the PI3K/AKT/mTOR pathway, which is frequently activated in BRCA1-mutated cancers, are being developed [288]. Other proteins and pathways that are being explored as potential targets include PARP (poly (ADP-ribose) polymerase) [289], and ATR (ataxia-telangiectasia and Rad3-related protein).

5.4.2. Proposed approach for activity prediction of molecules with BCRA1

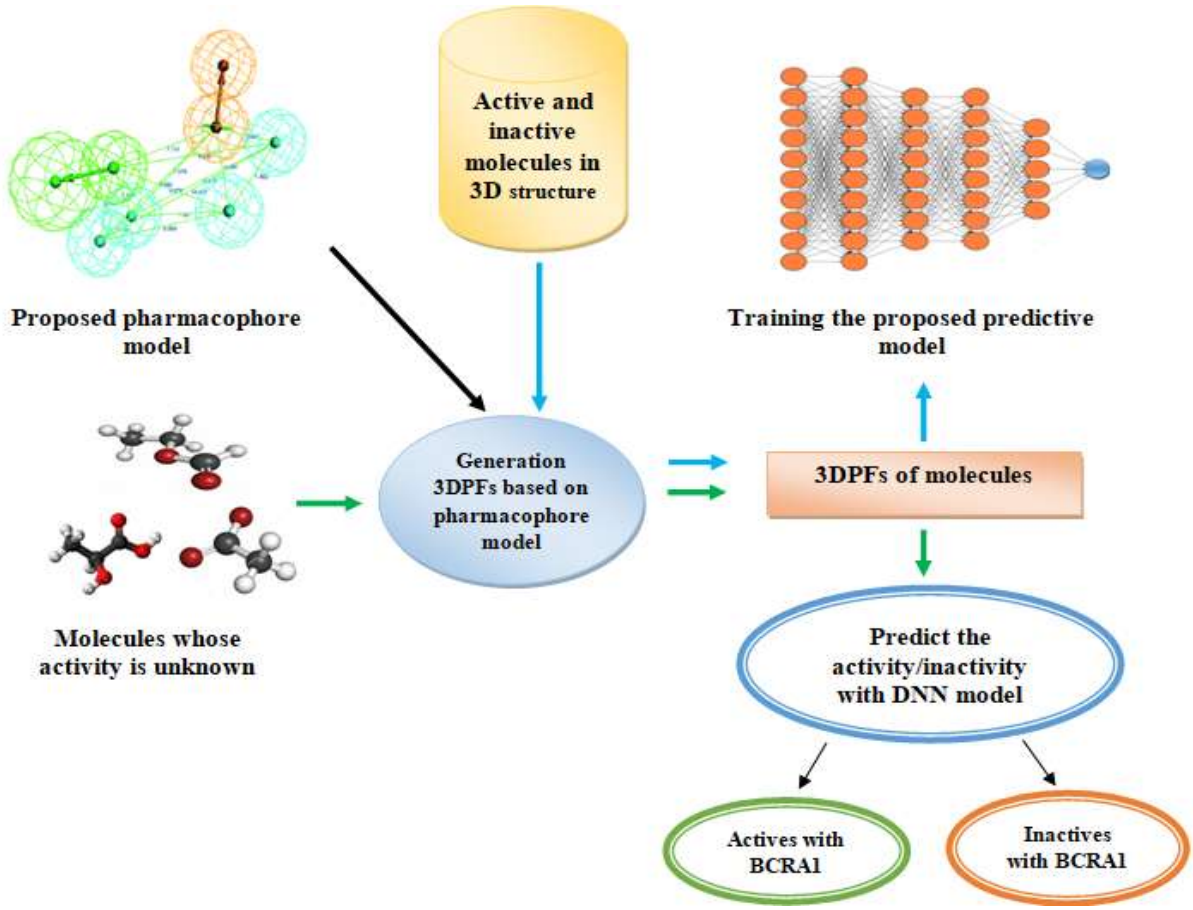


Figure 5.11. The proposed approach using 3DPF for activity prediction of molecules with BCRA1.

This approach has been used to predict the activity/inactivity of chemical compounds with the BRCA1 receptor (Figure 5.11). A molecule as active or inactive means whether the molecule

has a biological effect on the activity or function of BRCA1 gene, where : an active molecule is one that can interact with BRCA1 in a specific way that leads to a biological effect. For example, an active molecule might bind to BRCA1 and inhibit its activity, or it might enhance its activity. The specific biological effect will depend on the exact nature of the interaction between the molecule and BRCA1. In contrast, an inactive molecule is one that does not have a biological effect on the activity or function of BRCA1 gene. This might be because the molecule does not bind to BRCA1, or because it binds to BRCA1 in a way that does not affect its activity. Determining whether a molecule is active or inactive with respect to BRCA1 is an important step in drug discovery, as it can help researchers identify potential drug candidates that could be developed into therapies for BRCA1-associated cancers.

The first step in our approach is to propose a pharmacophore model that describes the distance ranges of certain features from a reference point. Once the pharmacophore model has been created, 3D pharmacophore fingerprints are generated for each molecule of the used dataset. These fingerprints are then used as input for a proposed deep neural network, which is trained to predict the biological activity of the molecule. The neural network learns to recognize patterns in the fingerprints that are associated with high or low activity, allowing it to make accurate predictions about the activity of new compounds. Finally, we used the final model to screen a set of molecules with unknown activity from the ChEMBL Database [176] and predicted whether they are potentially active or inactive against BRCA1.

5.4.2.1. Pharmacophore model

In our work, we proposed a pharmacophore model comprising four key features, Hydrogen Bond Donors (HBD), Hydrogen Bond Acceptors (HBA) Hydrophobic (Hyd) and Aromatic (Aro) that are critical for ligand-target binding. These features were identified through a series of experiments investigating the binding affinity and potency of various compounds. The distance ranges of these features from a reference point were determined to be (0-10), (5-10), (10-21), and (5-17) angstroms, respectively. Using this proposed pharmacophore model, we generated a 3D pharmacophore fingerprint that can be used to predict the activity or inactivity of compounds with BRCA1.

We relied to determine the optimal distance ranges of our model by generating a variety of fingerprints using different distance ranges and evaluating their performance in virtual screening experiments. Ultimately, the distance ranges that result in the most accurate and efficient predictions

5.4.2.2. Generation 3D pharmacophore fingerprint based on proposed model

The two stages for Generation 3D Pharmacophore Fingerprint based on proposed model are presented in the following:

A. Definition of pharmacophores spaces

The pharmacophores space is a set of all possible combinations between each feature of the model with the other features based on distance ranges. The combinations can be between two features, three, four, or more. In our work we have taken only a pharmacophores of three features to create the 3DPF. The creation of pharmacophores space is as follows:

Note: X and Y are two chemical features within the six pharmacophoric features.

• Definition of Space 1 of pharamcophore of three points with HBD (0-10):

Space 1 is a set of all possible combinations of hydrogen bond donor with two others features within the six features where the distance between a HBD and any other chemical feature is within the range of 0 to 10 angstroms (Figure 5.12.(a)) . This means that any chemical feature within 10 angstroms of the HBD is considered as a potential interaction partner. We define Space 1 of a pharmacophores of three features as follows:

Space 1 = {(HBD, X, Y) | dist (HBD, X) ≤ 10, dist (HBD, Y) ≤ 10 }

• Definition of Space 2 of pharamcophore of three points with HBA (5-10):

Space 2 is a set of all possible combinations of hydrogen bond acceptor with two others features within the six features where the distance between a HBA and any other chemical feature is within the range of 5 to 10 angstroms (Figure 5.12.(b)). This means that any chemical feature within 5 to 10 angstroms of the HBA is considered as a potential interaction partner. We define Space 2 of a pharmacophores of three features as follows:

Space 2 = {(HBA, X, Y) | 5 ≤ dist (HBA, X) ≤ 10, 5 ≤ dist (HBA, Y) ≤ 10}

• Definition of Space 3 of pharamcophore of three points with Hydrophobic group (10-21):

Space 3 is a set of all possible combinations of Hydrophobic group with two others features within the six features where the distance between a Hydrophobic group and any other chemical feature is within the range of 10 to 21 angstroms(Figure 5.12.(c)). This means that any chemical feature within 10 to 21 angstroms of the hydrophobic group is considered as a potential interaction partner. We define Space 3 of a pharmacophores of three features as follows:

Space 3 = {(Hyd, X, Y) | 10 ≤ dist (Hyd, X) ≤ 21, 10 ≤ dist (Hyd, Y) ≤ 21}

• Definition of Space 4 of pharamcophore of three points with Aromatic Ring (5-17):

Space 4 is a set of all possible combinations of Aromatic Ring with two others features within the six features where the distance between a HBD and any other chemical feature is within the range of 5 to 17 angstroms (Figure 5.12.(d)).This means that any chemical feature within

5 to 17 angstroms of the Arom is considered as a potential interaction partner. We define Space 4 of a pharmacophores of three features as follows:

Space 4 = {(Arom, X, Y) | 5 ≤ dist (Arom, X) ≤ 17, 5 ≤ dist (Arom, Y) ≤ 17}

B. Calculation of fingerprint

To generate the final 3D pharmacophore fingerprint, you would combine the three spaces into a single space and count the number of chemical features that fall within each space for a given molecule. The final space can be defined as the union of the three spaces as follows:

Space = Space1 ∪ Space2 ∪ Space3 ∪ Space4

Then, a binary vector is created where each combination in the pharmacophore space is assigned a place in the vector (Figure 5.12). If a molecule contains a specific pharmacophore in the space, that position in the binary vector will have a value of 1, and if the molecule doesn't contain that pharmacophore in that space, that position in the binary vector will have a value of 0. The resulting binary vector represents the 3D pharmacophore fingerprint of the molecule.

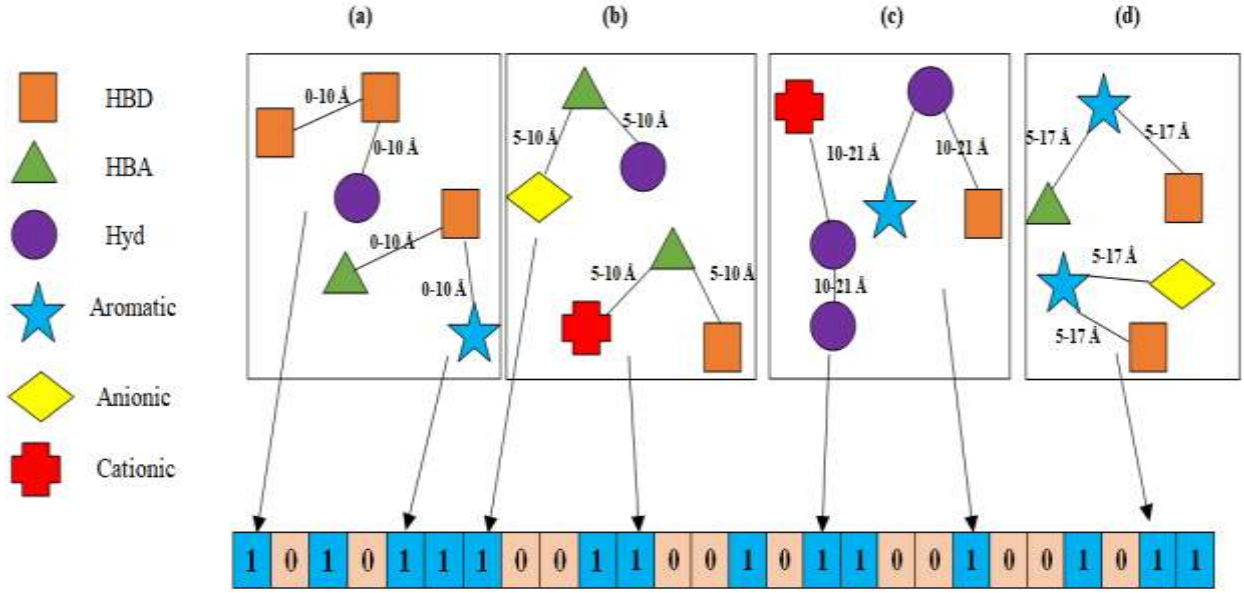


Figure 5.12. Generation of 3DPF using pharmacophore model.

5.4.2.3. Proposed predictive model

Based on Figure 5.13, the utilized DNN3 model comprises of six interconnected layers. The initial layer functions as the input layer, consisting of 512 bits that receive input data and transmit it to the first hidden layer. The first hidden layer contains 512 neurons and multiplies the received input data with weights before forwarding it to the subsequent layer comprising of 256 neurons. The subsequent layer is connected to another layer that also has 256 neurons.

Lastly, the DNN3 model incorporates a final hidden layer consisting of 64 neurons, which is connected to an output layer containing a single neuron. The activation function used in all neurons, except for the output neuron is the ReLU function. For the activation function in the output layer, we use the sigmoid function.

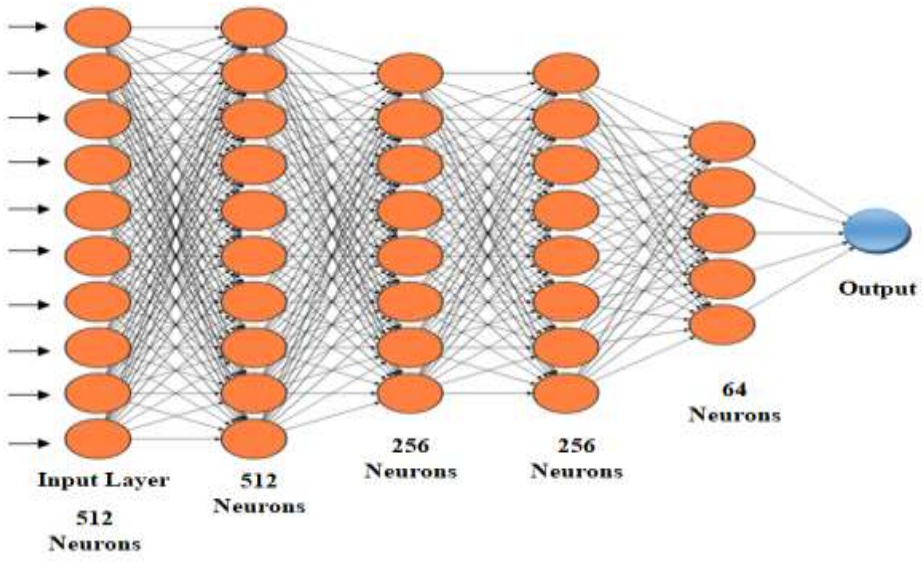


Figure 5.13. DNN3 model for activity prediction of molecules with BCRA1.

Table 5.5. Hyperparameters setting of DNN3 model.

Optimizer	Loss function	Learning rate	Dropout	Data Split	batch size
RMSprop	MSE	0.002	(0.25)	0.2	64

RMSprop (Root Mean Square Propagation) is a stochastic gradient descent optimization algorithm that is commonly used for training neural networks. The RMSprop algorithm computes a running average of the squared magnitudes of the gradients for each weight parameter. The running average is denoted by v_t, which is updated at each iteration t as:

$$v_t = beta * v_{t-1} + (1 - beta) * (grad_t)^2 \quad \text{Eq (5.4)}$$

Where beta is a hyperparameter that controls the weight given to the past gradient magnitudes relative to the current gradient magnitude. A typical value for beta is 0.9. The RMSprop update rule for each weight parameter w_t is then:

$$w_{t+1} = w_t - (learning\_rate / sqrt(v_t + epsilon)) * grad_t \quad \text{Eq (5.5)}$$

Where epsilon is a small constant added for numerical stability to avoid division by zero.

5.4.2.4. Experimental results

5.4.2.4.1. Data sets

Our experiment used a dataset of 3090 samples, where 1600 of them are active with BRCA1 and 1490 are inactive. The dataset was divided into three subsets: a training set, a validation set, and a test set. The validation set was used to tune the hyperparameters of the model, while the test set was used to evaluate the final model. The dataset was randomly split, with 20% used for validation and test the model, and the remaining 80% used for training. We extracted the active and inactive ligands with BRCA1 from the CHEMBL Database [271], and obtained the three-dimensional structure of all compounds from the PubChem database [272].

5.4.2.4.2. Overall performance

When the validation error starts to increase, it means that the model is starting to overfit, which can lead to poor generalization performance on new data. By stopping the training process at an earlier stage, we can prevent the model from overfitting and improve its generalization performance. The early stopping criterion of 20 consecutive epochs means that if the validation loss does not decrease for 20 consecutive epochs, the training process will be stopped to prevent overfitting. This helps to find the optimal hyperparameters for the model while ensuring that it does not overfit on the training data.

The models were implemented using Python and the TensorFlow library. The RDKIT and PMAPPER libraries were used to calculate the 3DPF of each compound. The constructed models were evaluated using six performance metrics**:** ACC, SEN, SPE, F1-Score, MCC, and AUC.

Table 5.6. The performance DNN3 model using 3DPF.

Data Set	**Loss**	**ACC**	**AUC**	**SEN**	**SPE**	**F1-Score**	**MCC**
Training Set	0.0142	0.9846	0.9902	0.9832	0.9847	0.9845	0.9690
Validation Set	0.1774	0.8074	0.8390	0.8361	0.7683	0.8271	0.6086

Figure 5.14 depicts the progression of the proposed DNN3 model's performance during training. To validate and test the model, we employed a subset of active and inactive ligands comprising 20% of the total set. Always, the best performance is indicated by a low loss function value. The validation set's error value achieved its lowest value of 0.1774 at epoch 10, while the training set had a value of 0.0142 at this epoch. The two sets achieved optimal accuracy convergence during this period, with the training set achieving 0.9846 and the validation set achieving 0.8074. Furthermore, the training set demonstrated excellent performance in the other measures (AUC, SEN, SPE, F1-Score, and MCC) during the same period, while the test set gave o good values with all metrics, as evidenced by Table 5.6.

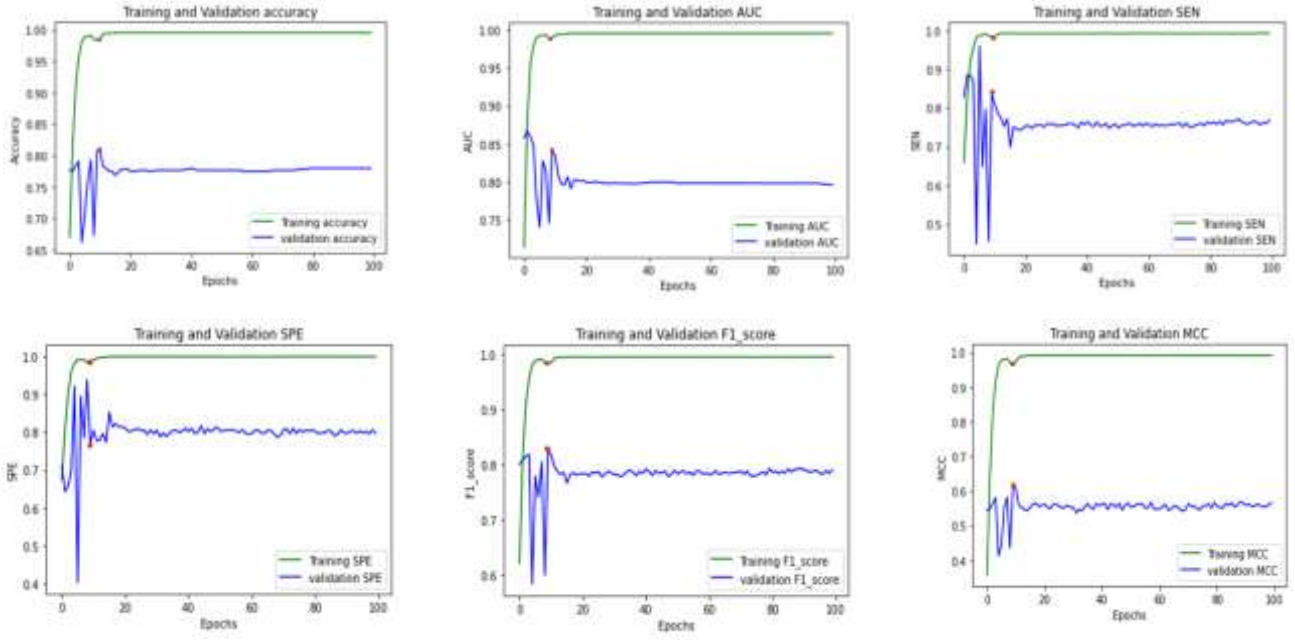
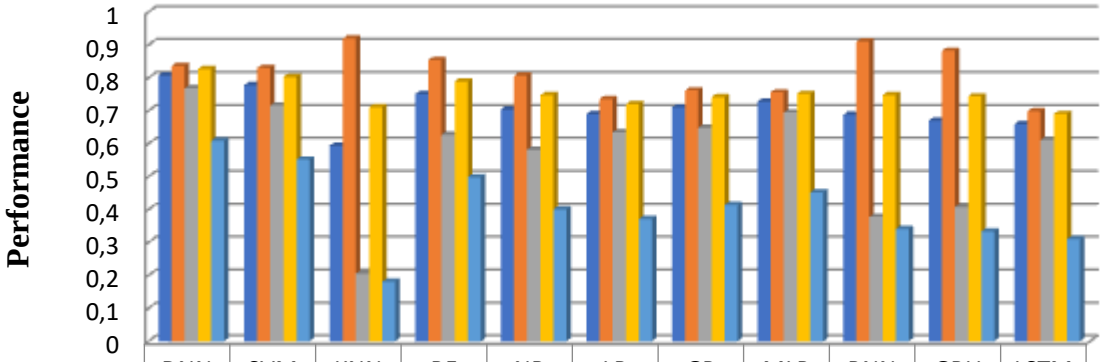


Figure 5.14. The performance evolution of the DNN3 model using 3DPF.



	DNN	SVM	KNN	RF	NB	LR	GP	MLP	RNN	GRU	LSTM
ACC	0,8074	0,7783	0,5938	0,7508	0,7038	0,6893	0,7103	0,7281	0,6875	0,6699	0,6586
SEN	0,8361	0,8303	0,9196	0,8541	0,8065	0,7351	0,7619	0,7559	0,9103	0,8824	0,6983
SPE	0,7683	0,7163	0,2056	0,6276	0,5815	0,6347	0,6489	0,695	0,3781	0,408	0,6115
F1-Score	0,8271	0,8028	0,7111	0,7884	0,7475	0,7201	0,7409	0,7514	0,7479	0,7438	0,6903
MCC	0,6086	0,5518	0,1812	0,4982	0,4002	0,3717	0,4138	0,4515	0,3412	0,3336	0,3103

Figure 5.15. The comparison of the DNN3 model's performance to that of other methods using 3DPF.

Figure 5.15 illustrates a comparison of the performance of the DNN3 model with other methods to demonstrate the efficacy of the generated fingerprints and the proposed predictive model. Specifically, the results have been compared with several ML methods, including SVM, KNN, RF, NB, LR, and GP, as well as several deep learning methods, such as MLP, RNN, GRU, and LSTM. Notably, the proposed DNN3 model using the generated fingerprints outperformed all other methods, achieving an accuracy of 0.8074 and the best values for all metrics except sensitivity, which was excellent at 0.9196 with KNN. SVM came in second

place with a good accuracy value of 0.7783, followed by RF in third place with an accuracy of 0.7508. NB, GP, and MLP achieved accuracy with convergent values. Among the deep learning methods, RNN, GRU, and LSTM gave reasonable values for all metrics except for SPE with RNN and MCC with RNN and LSTM, where they produced poor results.

5.5. Predict activity/inactivity of unknown molecules with BRCA1

In this stage, we utilized a ChEMBL database with inconclusive activity data for the BCRA1 gene. To predict the activity/inactivity of 150 molecules, we applied our final predictive deep learning model that includes fixed parameters and weights. Furthermore, we selected two other models, namely SVM and RF, which exhibited good performance in our experiments where SVM achieved an accuracy value of 0.7783, while RF attained a value of 0.7508. To further validate the accuracy of our DNN3 model, we employed SVM and RF models. The results are presented in a Table.5.7 that lists the molecule ID, their SMILES format, and the predictions of all three models.

Table 5.7. The predicted activity of 150 molecules with the BRCA1 gene by DNN3, SVM, and RF.

ID Molecule	SMILES Representation	DNN3	SVM	RF
CHEMBL1500037	O=C(NCc1ccccc1Cl)c1ccc(-c2ccccc2)cc1	Active	Active	Active
CHEMBL1500051	Cc1cc(N=Nc2ccc3ccccc3c2)ccc1O	Inactive	Active	Active
CHEMBL1500060	CC(=O)Oc1ccc(C(=O)Nc2ccc3c(c2)OCCO3)cc1	Active	Active	Active
CHEMBL1500066	Cc1cc(-c2nnc(SCc3ccccn3)o2)no1	Active	Active	Active
CHEMBL1500106	CC(=O)c1sc(NC(=O)c2ccccc2F)nc1C	Inactive	Active	Active
CHEMBL1500140	COc1ccc(/C=C/C(=O)O/N=C(\N)c2ccccc2)cc1OC	Inactive	Active	Active
CHEMBL1500193	Cc1cc(NC(=O)c2ccccc2)ncc1NC(=O)c1ccco1	Active	Active	Active
CHEMBL1500220	c1ccc2[nH]c(SCc3ccc(-c4nnco4)cc3)nc2c1	Active	Active	Active
CHEMBL1500290	O=C(CSc1nc2ccccc2s1)N1CCCCC1	Active	Active	Active
CHEMBL1500304	O=c1c2c3c(sc2nc(NCCO)n1Cc1ccco1)CCCCC3	Inactive	Inactive	Inactive
CHEMBL1500348	COc1ccc(Cc2nnc(NC(=O)c3ccc(OC)cc3OC)s2)cc1	Inactive	Inactive	Inactive
CHEMBL1500367	COc1ccc(N2CCN(c3ncccn3)CC2)cc1	Inactive	Active	Active
CHEMBL1500395	O=C(NC1CCCCC1)c1ccc2c(c1)OCCO2	Active	Active	Active
CHEMBL1500427	N#Cc1ccccc1NC(=O)c1ccc2c(c1)OCCO2	Active	Active	Active
CHEMBL1500438	Cc1cccn2c(=O)c3cc(C(=O)NCCCN4CCCC4=O)n(C)c3nc12	Inactive	Inactive	Inactive
CHEMBL1500499	CC(=O)Nc1ccccc1C(=O)OCc1cccc(F)c1	Inactive	Inactive	Active
CHEMBL1500515	O=C(O)c1ccc(COc2ccc(Cl)cc2Cl)o1	Inactive	Inactive	Inactive
CHEMBL1500540	Cc1ccc(OCc2ccc(C(=O)NN)o2)cc1C	Inactive	Inactive	Active

CHEMBL1500643	Cc1ccc(N/C(=C\C(=O)c2ccccc2)C(=O)O)cc1	Active	Active	Active
CHEMBL1500819	O=C(NCc1ccc(F)cc1)c1cnn2ccccc12	Active	Active	Active
CHEMBL1500876	O=C(Nc1ccc2nc(S)sc2c1)c1ccccc1	Active	Active	Active
CHEMBL1500941	Cc1c(C(=O)NC2CCOC2)oc2c(F)cccc12	Active	Active	Active
CHEMBL1500961	CCCn1nnc(NC(=O)c2ccc(OC)cc2)n1	Active	Active	Active
CHEMBL1500997	CC(C)c1ccc(/C=C2\SC(=O)N(CC(=O)NC3CCS(=O)(=O)C3)C2=O)cc1	Active	Active	Active
CHEMBL1501051	COc1ccc(-c2ncc3ccccn23)cc1	Inactive	Inactive	Active
CHEMBL1501064	O=C(NCCOc1ccccc1)c1ccccc1Cl	Active	Active	Active
CHEMBL1501113	CN(CC(=O)Nc1ccc(Cl)c(S(=O)(=O)N(C)C)c1)Cc1ccccc1	Inactive	Inactive	Inactive
CHEMBL1501121	CC(C)c1cc(C(N)=O)c(NC(=O)c2cccc(COc3ccc(Br)cc3)c2)s1	Inactive	Inactive	Inactive
CHEMBL1501168	Cc1c(C(=O)Nc2cc(F)cc(F)c2)cnn1-c1ccccc1	Active	Active	Active
CHEMBL1501169	Nc1ccc2c(c1)nnn2-c1ccccc1	Active	Active	Active
CHEMBL1501234	CC(=O)c1ccc(NC(=S)N2CCCC(CO)C2)cc1	Inactive	Inactive	Inactive
CHEMBL1501255	O=C(Nc1nc(-c2ccc3c(c2)OCCO3)cs1)C1CCCC1	Active	Active	Active
CHEMBL1501282	Cc1cc(C(=O)CSc2nc3ccccc3o2)c(C)n1C1CC1	Active	Active	Active
CHEMBL1501345	N#Cc1ccccc1NC(=O)COc1ccc2c(c1)OCO2	Inactive	Inactive	Inactive
CHEMBL1501377	CN(C)c1oc(-c2ccc(OCc3ccccc3)cc2)nc1C#N	Active	Active	Active
CHEMBL1501419	O=C1c2ccccc2C(=O)C1(CO)c1ccccc1	Inactive	Active	Active
CHEMBL1501423	OC1(c2cccc(C(F)(F)F)c2)CCN(c2ccc(-c3ccccc3)nn2)CC1	Active	Inactive	Inactive
CHEMBL1501469	COc1ccccc1C(=O)OCc1nc(N)nc(Nc2ccccc2C)n1	Inactive	Inactive	Active
CHEMBL1501515	Cc1ccc(-c2nc3ccccc3n2C)o1	Inactive	Inactive	Inactive
CHEMBL1501528	COc1ccc(Nc2cc(Nc3ccccc3)ncn2)cc1	Active	Active	Active
CHEMBL1501587	Cc1ccc(CNc2ccc3c(c2)nc(C)n3C)cc1	Active	Active	Active
CHEMBL1501596	Cc1ccc2nc(NC(=O)C3CCCN(S(=O)(=O)c4c[nH]cn4)C3)sc2c1	Active	Active	Active
CHEMBL1501618	CC(C)c1ccc(NC(S)=Nc2ccc3c[nH]nc3c2)cc1	Inactive	Inactive	Inactive
CHEMBL1501634	CN(Cc1cccs1)C(=O)CSc1nc2ccccc2[nH]1	Inactive	Inactive	Inactive
CHEMBL1501692	COc1ccc(-n2nc(C(=O)NCC(=O)N3CCC4(CC3)OCCO4)c3ccccc3c2=O)cc1	Active	Active	Active
CHEMBL1501724	CC1=CC(=C2C(=O)c3ccccc3C2=O)C=C(C)N1Cc1ccco1	Inactive	Inactive	Inactive
CHEMBL1501825	Cc1csc2nc(CNC(=O)c3ccco3)cn12	Active	Active	Active
CHEMBL1501845	CC(=O)c1sc(NCc2ccc(C)cc2)nc1C	Active	Active	Active
CHEMBL1501846	Cc1nc(NCc2ccccc2)sc1C(=O)Nc1ccccc1	Active	Active	Active
CHEMBL1501851	Cc1nc2c(C(=O)N3CCCCC3)cnn2c(C)c1Cc1ccccc1F	Active	Active	Active
CHEMBL1501864	C1=C(c2ccc(-c3ccccc3)cc2)Nc2ncnn2C1c1ccco1	Active	Inactive	Inactive
CHEMBL1501910	O=C(Nc1cccc(Cl)c1)N1CCc2ccccc2C1	Inactive	Active	Inactive

CHEMBL1501923	CC(=O)c1ccc(NC(=O)c2noc3c2CCCCC3)cc1	Active	Active	Active
CHEMBL1501924	CCOC(=O)c1sc(=N)n(-c2ccc(OC)cc2)c1C	Active	Active	Active
CHEMBL1501925	Cc1cc(CNC(=O)c2cccc(OC3CCN(S(=O)(=O)N(C)C)CC3)c2)on1	Inactive	Active	Inactive
CHEMBL1501987	CCCCn1c(=O)[nH]c2nc3cc(N)ccc3nc2c1=O	Active	Active	Active
CHEMBL1502016	O=C(O)c1cncc(C#CCC2CCCCC2)c1	Active	Active	Active
CHEMBL1502124	COc1cccc(C2=NOC(C(=O)N3CCOCC3)C2)c1	Active	Active	Active
CHEMBL1502184	Brc1ccc(-c2nnc(-c3ccccc3Br)o2)cc1	Inactive	Active	Active
CHEMBL1502234	S=C(Nc1ccc2nsnc2c1)N1CCCCCC1	Active	Active	Active
CHEMBL1502261	COc1ccc(-n2nc([N+](=O)[O-])c(NCc3ccco3)[n+]2[O-])cc1	Active	Active	Active
CHEMBL1502306	CN(C)c1ccc(C(=O)c2ccccc2)cc1	Inactive	Inactive	Inactive
CHEMBL1506977	Cc1ccc2c(CN(C)Cc3cccc(F)c3)cc(=O)oc2c1	Active	Active	Active
CHEMBL1502503	Cc1ccnc(NC(=O)c2ccc(Cl)s2)c1	Active	Inactive	Inactive
CHEMBL1502535	COc1ccc(C2=NN(S(C)(=O)=O)C(c3cccs3)C2)cc1OC	Active	Active	Active
CHEMBL1502567	O=c1c2ccccc2nc2ccccn12	Active	Active	Inactive
CHEMBL1502620	O=C(NC1CC1)c1sc2ccccc2c1Cl	Active	Active	Active
CHEMBL1502635	Cc1ccc(CNC2=Nc3ccccc3CS2)cc1	Active	Active	Active
CHEMBL1502692	C=CCn1c(=O)c(C(=O)Nc2ccccc2S(N)(=O)=O)c(O)c2ccccc21	Active	Active	Active
CHEMBL1502693	O=C(CNS(=O)(=O)c1cccs1)N1CCc2ccccc2C1	Active	Inactive	Inactive
CHEMBL1502698	COc1cccc(Nc2cc(C)c3ccccc3n2)c1	Inactive	Active	Active
CHEMBL1502706	O=S(=O)(Cc1cc(-c2ccccc2Cl)no1)c1ccccc1	Active	Active	Active
CHEMBL1502746	COc1cccc(CC(=O)Nc2ccc3nc(C)sc3c2)c1	Active	Active	Active
CHEMBL1502801	CC(Cc1cccs1)NC(=O)c1cccs1	Active	Active	Active
CHEMBL1502802	Cc1ccc(NC(=O)NC(=O)CSc2nnc(-c3ccccc3F)n2N)c(C)c1	Inactive	Active	Active
CHEMBL1502863	CC(C)(C)C(=O)c1cnc(Nc2ccccc2Cl)s1	Inactive	Inactive	Inactive
CHEMBL1502866	COc1cc(C(=S)N2CCCC2)ccc1OCc1ccccc1	Active	Active	Active
CHEMBL1502977	COc1ccc(C2CC(c3ccc4ccccc4c3)=NN2C(=O)COC(=O)C2CCC(=O)N2)cc1	Active	Active	Active
CHEMBL1503002	COc1ccc(C(=O)Nc2cc(OC)nc(OC)n2)cc1	Active	Inactive	Inactive
CHEMBL1503016	COc1ccc(NC(=O)Nc2ccc3nccnc3c2)cc1	Active	Active	Active
CHEMBL1503076	Nc1nc(-c2cc3ccccc3o2)cs1	Active	Active	Active
CHEMBL1503081	Cc1ccc(C(=O)NCCCc2ccccc2)s1	Active	Active	Active
CHEMBL1503085	Cc1noc(N)c1C(=O)Nc1ccc(C(C)(C)C)cc1	Active	Active	Active
CHEMBL1503175	O=C(/C=C/c1ccco1)NC(=S)Nc1cccc(C(=O)O)c1	Active	Active	Active
CHEMBL1503238	Cc1nc(Nc2ccccc2)sc1-c1ccccc1	Inactive	Inactive	Active
CHEMBL1503239	O=C(/C=C/c1cc2ccccc2o1)c1ccccc1O	Active	Active	Active

CHEMBL1503246	CCOc1ccc2nc(NC(=O)Cc3cccs3)sc2c1	Active	Active	Active
CHEMBL1503339	O=C(CSc1nnc(-c2ccc(O)cc2)o1)Nc1ccccc1Br	Active	Active	Active
CHEMBL1503383	O=C(CSc1nnc(-c2ccccc2F)o1)N1CCCC1	Inactive	Inactive	Inactive
CHEMBL1503389	CCOC(=O)COc1ccc(Br)cc1/C=C\c1nc(O)nc(O)c1[N+](=O)[O-]	Active	Active	Active
CHEMBL1503390	COc1nc(-c2ccccc2)ccc1-c1noc(C)n1	Inactive	Inactive	Inactive
CHEMBL1503406	COc1ccc(CCNC(=O)C2CC(c3ccccc3[N+](=O)[O-])=NO2)cc1	Active	Active	Active
CHEMBL1503484	CCOc1ccc(Nc2oc(Cc3ccccc3)nc2C#N)cc1	Active	Inactive	Inactive
CHEMBL1503486	ClC1(Cl)CC1CSc1nnc(-c2ccncc2)o1	Active	Active	Active
CHEMBL1503742	Cc1ccc(/C=C/C(=O)N2CCCC(C)C2)cc1	Active	Active	Active
CHEMBL1503754	CN(C(=O)c1ccc2c(c1)OCO2)C1CCCCC1	Active	Active	Active
CHEMBL1503947	O=C1CCCC(=O)C1=NNc1ccc2c(c1)OCO2	Active	Active	Active
CHEMBL1504045	Nc1nnc(SCCCC(=O)c2ccccc2)s1	Active	Active	Active
CHEMBL1504121	Cc1sc(NC(=O)c2ccccc2)c(C#N)c1-c1ccccc1	Active	Active	Active
CHEMBL1504147	Cc1ccccc1NC(=O)C(=O)Nc1ccccc1C	Active	Active	Active
CHEMBL1504152	COc1ccc2nc(NC(=O)NC3CCCCC3)sc2c1	Active	Active	Active
CHEMBL1504221	N#C/C(=C/c1ccc(N2CCCCCC2)o1)c1cccc(F)c1	Active	Active	Active
CHEMBL1504256	COc1cc(NC(=O)c2cc3nc(C)cc(C)n3n2)c(OC)cc1Cl	Inactive	Active	Active
CHEMBL1504269	Cc1cc(C(=O)CSc2ccc(F)cc2)cc(C)c1O	Inactive	Inactive	Inactive
CHEMBL1504289	Cc1ccc(OCc2nc(-c3cccs3)no2)c([N+](=O)[O-])c1	Active	Active	Active
CHEMBL1504452	Cc1ccc(NC(=O)CSc2nccn2C)cc1F	Inactive	Active	Active
CHEMBL1504527	O=C(CSc1nnc(-c2ccccn2)o1)Nc1ccccc1C(F)(F)F	Active	Active	Active
CHEMBL1504529	Cc1cc(C)n2c(Br)c(CSc3nc4ccccc4s3)nc2n1	Active	Active	Active
CHEMBL1504572	Cc1c(Cl)cnc(NC(=O)CSc2nnc3ccccn23)c1Cl	Inactive	Active	Inactive
CHEMBL1504601	O=S(=O)(Cc1nc2ccccc2s1)c1ccccc1	Inactive	Inactive	Inactive
CHEMBL1504609	Cc1nc2ccccc2n1CCCOc1ccccc1	Inactive	Active	Active
CHEMBL1504690	C=CCNC(=O)/C=C/c1ccc(C(C)C)cc1	Active	Active	Active
CHEMBL1504718	O=C(Nc1ccc(OC(F)(F)F)cc1)Nc1cccnc1	Active	Active	Active
CHEMBL1504728	Cc1ccc(NC(=S)NC(=O)C2CCC2)cc1C	Active	Active	Active
CHEMBL1504840	C=CCNC(=O)c1ccc2c3n(nc2c1)-c1ccccc1CO3	Active	Active	Active
CHEMBL1504847	Cc1cc(C)n2cc(CSc3ccc(Br)cc3)nc2n1	Active	Active	Active
CHEMBL1505056	O=C(NN=C1CCCCCC1)c1cccs1	Active	Active	Active
CHEMBL1505090	COc1ccc(NC(=O)Nc2ccc(Br)cn2)cc1	Inactive	Active	Active
CHEMBL1505147	Cc1[nH]c2ccccc2c1/C=C(\C#N)C(=O)NC1CC1	Active	Active	Active
CHEMBL1505212	COc1cc(NC(=O)c2csc3c2CCCC3)ncn1	Inactive	Active	Active

CHEMBL1505303	CC(=O)c1sc(NC(=O)c2cccc(C)c2)nc1C	Active	Active	Active
CHEMBL1505428	COc1cc2sc3ccc(-c4ccc5[nH]ccc5c4)cc3c(=O)c2cc1OC	Active	Active	Active
CHEMBL1505469	CCc1ccc(NC(=O)Nc2cccc(OC)c2)cc1	Inactive	Inactive	Inactive
CHEMBL1505482	Cn1ncc([N+](=O)[O-])c1C(=O)Nc1ccc(-c2nc3ccccc3[nH]2)cc1	Active	Active	Active
CHEMBL1505529	O=C(c1cccc([N+](=O)[O-])c1)c1ccc2[nH]c(O)nc2c1	Inactive	Inactive	Inactive
CHEMBL1505598	Cc1cc(Cl)ccc1OCCCC(=O)N1CCOCC1	Inactive	Inactive	Inactive
CHEMBL1505651	Cc1cc(NS(=O)(=O)c2ccc(NC(=O)Nc3ccc4c(c3)OCO4)cc2)no1	Inactive	Active	Active
CHEMBL1505703	COc1cccc(-c2nc(-c3ccncc3)no2)c1	Inactive	Inactive	Inactive
CHEMBL1505717	Cc1ccc(C(=O)Nc2cc3c(cc2Cl)OCCCO3)cc1	Active	Active	Active
CHEMBL1505940	Cc1c(C(=O)NCCc2ccccc2)oc2ccccc12	Active	Active	Active
CHEMBL1506034	Cc1cccc(OCCn2nnc3ccccc32)c1C	Active	Active	Active
CHEMBL1506130	COC(=O)c1[nH]c2ccc(Br)cc2c1NC(=O)Oc1ccccc1	Active	Active	Active
CHEMBL1506175	COc1cccc(-c2nnc(-c3ccc(C)o3)o2)c1	Inactive	Inactive	Inactive
CHEMBL1506203	Cc1cc(-c2nnc(S)o2)c(C)n1-c1ccccc1	Active	Active	Active
CHEMBL1506254	CN(C)c1cccc(C(=O)NN=C(S)NC2CC3CCC2C3)c1	Inactive	Active	Active
CHEMBL1506291	OC(COc1ccc2ccccc2c1Cl)CN1CCOCC1	Inactive	Active	Active
CHEMBL1506327	C=CCNC(=O)c1ccc(-c2cccc([N+](=O)[O-])c2)o1	Inactive	Inactive	Inactive
CHEMBL1506330	O=C(Nc1ccc2c(c1)CCC2)Nc1ccc(F)cc1F	Active	Active	Active
CHEMBL1506396	c1ccc2sc(NC3=NCCN3)nc2c1	Active	Active	Active
CHEMBL1506407	Cc1ccc(C(=O)Nc2ccc3c(c2)ncn3C)cc1	Active	Active	Active
CHEMBL1506416	CCCCOC(=O)Nc1ccc(C)c(Cl)c1	Active	Active	Active
CHEMBL1506461	Nc1ccc(C(=O)OCc2ccncc2)cc1[N+](=O)[O-]	Active	Active	Active
CHEMBL1506607	O=C(Nc1cccc(-c2nc3cc(Cl)ccc3o2)c1)C1CCCO1	Inactive	Active	Active
CHEMBL1506622	Cc1ccc(C(=O)/C=C/c2c[nH]c3ccccc23)cc1	Active	Active	Active
CHEMBL1506647	O=C(CSc1nnc(-c2ccc(F)cc2)o1)N1CCCC1	Active	Active	Active
CHEMBL1506677	Clc1ccccc1SCc1nc2ccccc2[nH]1	Active	Active	Active
CHEMBL1506690	COc1ccccc1Oc1ncnc2c1cnn2-c1ccccc1	Inactive	Active	Active
CHEMBL1506712	CC(=O)Nc1ccc(-c2csc(NCc3ccccc3)n2)cc1	Active	Active	Active
CHEMBL1506750	O=C(/C=C/c1ccc(-c2cccc([N+](=O)[O-])c2)o1)N1CCOCC1	Active	Active	Active
CHEMBL1506816	O=C(NCCc1ccccc1)c1nnc(O)c2ccccc12	Inactive	Active	Active

Figure 5.16 depicts the percentage of predicted active and inactive molecules by each model. Notably, RF predicted the highest number of active molecules, followed by SVM and DNN3, while DNN3 predicted the highest number of inactive molecules. These results can be explained by the fact that RF had the highest SEN value, while DNN3 had the highest SPE

value in the precedent results. To obtain a more reliable final decision, we recommend considering the results agreed upon by all three methods.

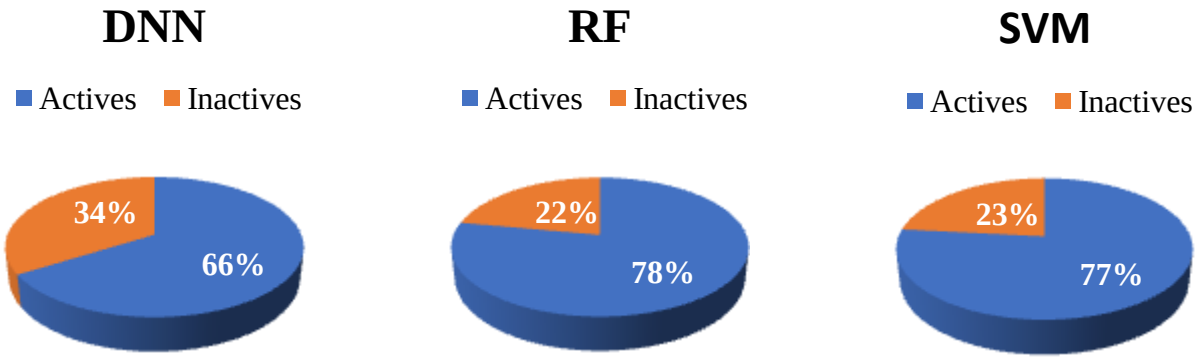


Figure 5.16. The percentage of predicted active and inactive molecules.

Figure 5.17 shows the proportion of molecules that were classified similarly and differently by the three models, i.e., molecules that were predicted in the same way by the three models and molecules that were classified differently. In the preceding table, we highlighted in blue the classification that was identical across the three models, where the models have classified 92 molecules as active and 24 as molecules inactive. In summary, these results demonstrate the effectiveness of the generated 3D fingerprints.

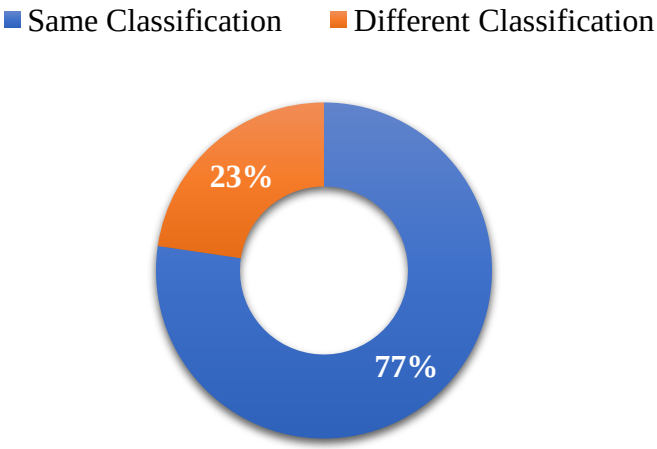


Figure 5.17. The percentage of the same and different classification.

5.6. Conclusion

In this chapter we have explored the use of deep learning for virtual screening based on 3D pharmacophore fingerprints for activity pradiction, so we proposed two approaches. The first approach based on using multiple conformations of the molecules and training a proposed

CNN model on a large dataset of known active and inactive compounds with CDK1, this approach can improve the accuracy of activity predictions and help identify promising drug candidates. The second approach has been used to predict the activity of potential drug candidates with the BRCA1 receptor, which is associated with breast cancer by combining 3D pharmacophore fingerprints with a deep neural network, this approach can help identify potential drug candidates that are likely to be effective against the BRCA1 receptor. Moreover, by creating a pharmacophore model based on the 3D structure of known active compounds, this approach can help identify new compounds that are structurally similar to known active compounds but may have improved activity or other properties.

General Conclusion And Future Prospects

Predicting the activity of molecules against a specific target was a challenging task that often relied on experimental methods, such as high-throughput screening. These methods were time-consuming, expensive, and often yielded limited results due to the vast chemical space that needed to be explored. Another challenge was the lack of understanding of the complex molecular interactions that underlie biological activity. Additionally, traditional approaches often failed to account for the conformational flexibility of molecules, which can greatly impact their activity. As a result, there was a need for a more efficient and accurate method for predicting the activity of molecules against a specific target.

The development of technology in recent decades has led to the creation of large chemical and biological numerical libraries containing vast amounts of molecular data. This wealth of data has fueled the emergence of virtual screening as a new line of research in drug development. Virtual screening involves the use of computational methods to search large databases of molecules and identify potential drug candidates based on their predicted activity against a specific target and the development of predictive models based on known molecular interactions. The advantages of virtual screening include its speed and cost-effectiveness compared to traditional experimental methods, as well as its ability to explore much larger chemical space. The use of virtual screening has contributed to the discovery of several important drugs, such as raltegravir for the treatment of HIV and vemurafenib for the treatment of melanoma. Virtual screening has also enabled the identification of novel drug targets and the optimization of existing drugs for improved efficacy and reduced toxicity. However, virtual screening is not without its own challenges, such as the need for accurate molecular descriptors and the potential for false positives or negatives due to the limitations of computational methods. Nonetheless, virtual screening has become an indispensable tool for predicting the activity of molecules against specific targets.

The availability of data on two important targets, CDK1 and BRCA1, which play crucial roles in the development of some of the most dangerous types of cancer, has motivated us to develop new virtual screening methods focused on these targets. Our goal in those studies was to develop predictive models that can accurately predict the activity of molecules with CDK1 and BRCA1. 2D and 3D pharmacophore fingerprints are essential in molecular description as they provide a simplified representation of a molecule, capturing its essential features responsible for binding to a target protein. In our study of the CDK1 receptor, we proposed two approaches based on deep learning, which utilized 2D pharmacophore fingerprints to classify molecules as active or inactive. First, we suggested two different ranges of distance in each approach, for efficient fingerprint generation. Then, we proposed deep learning architectures that used the generated fingerprints as input data.

In the two other approaches, we utilized 3D pharmacophore fingerprints along with deep learning. In the third approach, we focused on CDK1 and generated multiple conformations. For each conformation, we calculated the corresponding fingerprint, and finally, we combined these fingerprints to create a hybrid fingerprint, which was used as input in our predictive model. In the last approach, our focus was on the BCRA1 gene. We proposed a

pharmacophore model specific to this gene, which allowed us to generate fingerprints based on this model. These fingerprints were used in our analysis and prediction tasks.

The proposed approaches yielded excellent results, comparable to the most widely used ML methods in the field. Overall, our proposed approaches demonstrate the potential of ML methods in predicting the activity of molecules on the CDK1 and BCRA1. We believe our findings will contribute to the development of new drugs and therapies for various diseases.In summary, these methods provided valuable insights and demonstrated the potential of pharmacophore-based fingerprinting in predictive modeling for drug discovery and gene analysis.

The experimental study we conducted not only yielded promising results but also opened up a new and exciting field for exploration. We are deeply motivated to continue our research in this area and make meaningful contributions to its development. Our passion for this field stems from a strong belief in its humanitarian potential to significantly improve people's lives through the discovery of new drugs and treatments for various diseases. With access to vast libraries of millions of chemical compounds, we are optimistic about the possibilities that lie ahead.

In our current study, we focused primarily on the ligands' role in drug discovery. However, we recognize the importance of investigating the structure of disease receptors (proteins) more comprehensively. By exploring their potential interactions with available ligands, we aim to achieve more accurate and efficient results. This entails considering both the ligands and the proteins' three-dimensional structures in our analysis, as they play crucial roles in determining their binding affinities and activity profiles.

Looking forward, our research trajectory involves integrating information about ligands and proteins' structures to achieve a more holistic approach. By combining ligand-based and structure-based virtual screening methodologies, we expect to enhance the accuracy and reliability of our predictions. We believe that leveraging the strengths of both approaches will enable us to uncover novel insights and identify promising drug candidates with greater precision.

Moreover, our study highlights the effectiveness of both 2D and 3D pharmacophore fingerprints in virtual screening. These powerful tools capture essential molecular features that contribute to drug-target interactions. Moving forward, we aim to harness the potential of these fingerprints by combining them synergistically. By integrating the strengths of 2D and 3D pharmacophore fingerprints, we can create a more comprehensive representation of the molecular landscape, thus enhancing our ability to identify potential drug candidates and optimize their efficacy.

In conclusion, our study serves as a stepping stone towards the convergence of ligand-based and structure-based virtual screening approaches. By further exploring the complexities of protein structures and integrating them with ligand information, we aspire to unlock new dimensions in drug discovery. Our ultimate goal is to contribute to the development of innovative therapies and make a meaningful impact on human health and well-being.

Bibliographic References

[**1**] J. B. Hagen, "The origins of bioinformatics," Nat. Rev. Genet., vol. 1, pp. 231–236, 2000.

[**2**] M. D. Adams et al., "The genome sequence of Drosophila melanogaster," Science, vol. 287, pp. 2185–2195, 2000.

[**3**] R. Staden, "A strategy of DNA sequencing employing computer programs," Nucleic Acids Res., vol. 6, pp. 2601–2610, 1979.

[**4**] R. D. Fleischmann et al., "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd," Science, vol. 269, pp. 496–512, 1995.

[**17**] G. Jebah Nouairia, M. Ben Larbi, M.H. Yahyaoui, and M. Ben Naceur, "Comparaison de methodes d'extraction de l'ADN de lapin à partir du sang: Fiabilité et coût," in Congrès International de Biotechnologie et Valorisation des bio-ressources, Tunisia, pp. 7-8, 2013.

[**19**] M.D. Adams, J.M. Kelley, J.D. Gocayne, M. Dubnick, M.H. Polymeropoulos, H. Xiao, C.R. Merril, A. Wu, B. Olde, R.F. Moreno, and A.R. Kerlavage, "Complementary DNA sequencing: expressed sequence tags and human genome project," Science, vol. 252, no. 5013, pp. 1651-1656, Jun. 1991.

[**21**] H. Lodish et al., "Molecular Cell Biology," 4th ed., New York, NY, USA: Freeman & Co., 2000, pp. 1084.

[**22**] "Structure of RNA," Byju's, 2021. [Online]. Available: https://byjus.com/biology/structure-of-rna/. [Accessed: May 10, 2023].

[**24**] Proteins. (2020). Gsu.Edu. http://hyperphysics.phy-astr.gsu.edu/hbase/Organic/protein.html

[**25**] G. Wu, "Amino acids: metabolism, functions, and nutrition," Amino Acids, vol. 37, no. 1, pp. 1-17, 2009.

[**26**] J. T. Brosnan and M. E. Brosnan, "The sulfur-containing amino acids: an overview," J. Nutr., vol. 136, no. 6, pp. 1636S-1640S, 2006.

[**27**] "Amino acids - structure, optical activity, and classifications," Microbiology Notes, 2021. [Online]. Available: https://microbiologynotes.org/amino-acids-structure-optical-activity-and-classifications/.

[**31**] Nelson, D. L. and Cox, M. M., "Lehninger principles of biochemistry," W. H. Freeman, 2008.

[**32**] Rohl, C.A., Strauss, C.E., Misura, K.M. and Baker, D., "Protein structure prediction using Rosetta," Methods in Enzymology, vol. 383, pp. 66-93, 2004.

[**33**] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P., "Molecular Biology of the Cell," Garland Science, 2002.

[**34**] Voet, D. and Voet, J.G., "Biochemistry," John Wiley & Sons, 2011.

[**35**] Berg, J.M., Tymoczko, J.L. and Stryer, L., "Biochemistry (5th ed.)," WH Freeman, 2002.

[**29**] Al-Khami, A.A., Rodriguez, P.C. and Ochoa, A.C., "Energy metabolic pathways control the fate and function of myeloid immune cells," Journal of Leukocyte Biology, vol. 102, no. 2, pp. 369-380, 2017.

[**38**] J.L. Lahti, G.W. Tang, E. Capriotti, T. Liu, and R.B. Altman, "Bioinformatics and variability in drug response: a protein structural perspective," J R Soc Interface, vol. 9, pp. 1409-1437, Jul. 2012.

[**39**] M. Punta and Y. Ofran, "The Rough Guide to In Silico Function Prediction, or How To Use Sequence and Structure Information To Predict Protein Function," PLoS Comput Biol, vol. 4, no. 10, pp. e1000160, Oct. 2008.

[**40**] Relling, M.V. and Evans, W.E., "Pharmacogenomics in the clinic," Nature, vol. 526, no. 7573, pp. 343-350, 2015.

[**41**] R. K. Varshney, K. C. Bansal, P. K. Aggarwal, S. K. Datta, and P. Q. Craufurd, "Agricultural biotechnology for crop improvement in a variable climate: hope or hype?," Trends Plant Sci, vol. 16, no. 7, pp. 363-371, Jul. 2011.

[**43**] J. Handelsman, M. R. Rondon, S. F. Brady, J. Clardy, and R. M. Goodman, "Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products," Chemistry & Biology, vol. 5, no. 10, pp. R245-R249, Oct. 1998.

[44] P. Tarczy-Hornoch and M. Minie, "Bioinformatics Challenges and Opportunities," in Medical Informatics, vol. 3, pp. 63-94, Jan. 2005.

[45] M. Mathur, "Bioinformatics challenges: A review," Int. J. Adv. Sci. Res., vol. 3, no. 6, pp. 29-33, Nov. 2018.

[46] G. H. Fernald, E. Capriotti, R. Daneshjou, K. J. Karczewski, and R. B. Altman, "Bioinformatics challenges for personalized medicine," Bioinformatics, vol. 27, no. 13, pp. 1741-1748, 2011.

[**48**] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," AI magazine, vol. 17, no. 3, pp. 37-54, Fall 1996.

[**49**] P. N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining," Pearson, 2019.

[**50**] C. C. Aggarwal, "Data Mining: The Textbook," Springer, 2015.

[**52**] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Data Mining: Practical Machine Learning Tools and Techniques," Morgan Kaufmann, 2016.

[**53**] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Supervised Machine Learning: A Review of Classification Techniques," Informatica, vol. 31, no. 3, pp. 249-268, 2007.

[**54**] J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," Morgan Kaufmann, 2011.

[**57**] C. M. Bishop, "Pattern recognition and machine learning," Springer, 2006.

[58] A. Osojnik, P. Panov, and S. Džeroski, "Multi-label classification via multi-target regression on data streams," Mach. Learn., vol. 106, pp. 745-770, 2017.

[59] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," International Journal of Data Warehousing and Mining, vol. 3, no. 3, pp. 1-13, 2007.

[61] C. N. Silla Jr. and A. A. Freitas, "A survey of hierarchical classification across different application domains," Data Mining and Knowledge Discovery, vol. 22, pp. 31-72, 2011.

[64] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009.

[65] X. Su, X. Yan, and C.-L. Tsai, "Linear regression," WIREs Computational Statistics, first published on 10 February 2012.

[66] R. M. Heiberger and E. Neuwirth, "Polynomial Regression," in R Through Excel, 1st ed., ch. 5, pp. 55-68, Jan. 2009.

[68] O. Cokluk, "Logistic Regression: Concept and Application," Educational Sciences: Theory and Practice, vol. 10, no. 3, pp. 1397-1407, 2010.

[72] M. Thevaraja, A. Rahman, and M. Gabirial, "Recent Developments in Data Science: Comparing Linear, Ridge and Lasso Regressions," in Proceedings of the International Conference on Data Science and Information Technology (DISP), ISBN: 978-1-912532-09-4, pp. 1-6, 2019.

[76] T. A. Kumbhare, S. V. Chobe, et al., "An Overview of Association Rule Mining Algorithms," International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5, no. 1, pp. 927-930, 2014.

[77] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Record, vol. 22, no. 2, pp. 207-216, 1993.

[**80**] J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," Morgan Kaufmann, 2011.

[81] S. Shukla and N. S., "A Review on K-means Data Clustering Approach," International Journal of Information & Computation Technology, vol. 4, no. 17, pp. 1847-1860, 2014.

[**82**] F. Nielsen, "Hierarchical Clustering," in Introduction to HPC with MPI for Data Science, 1st ed., Boca Raton, FL, USA: CRC Press, ch. 4, pp. 1-20, 2016.

[**83**] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," WIREs Data Mining and Knowledge Discovery, vol. 1, no. 3, pp. 231-240, May/June 2011.

[**84**] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," Neural Computing and Applications, vol. 24, pp. 1477-1486, 2014.

[**85**] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90-105, June 2004.

[**86**] E. H. Ruspini, J. C. Bezdek, and J. M. Keller, "Fuzzy Clustering: A Historical Perspective," IEEE Computational Intelligence Magazine, vol. 14, no. 1, pp. 10-18, February 2019.

[**87**] A. Fatima, N. Nazir, and M. G. Kh, "Data Cleaning In Data Warehouse: A Survey of Data Pre-processing Techniques and Tools," I.J. Information Technology and Computer Science, vol. 3, pp. 50-61, March 2017.

[**88**] J. J. Tamilselvi and C. B. Gifta, "Handling Duplicate Data in Data Warehouse for Data Mining," International Journal of Computer Applications, vol. 15, no. 4, pp. 7-12, February 2011.

[**89**] J. V. den Broeck, S. A. Cunningham, R. Eeckels, and K. Herbst, "Data Cleaning: Detecting, Diagnosing, and Editing Data Abnormalities," PLOS Medicine, vol. 2, no. 10, p. e267, September 2005.

[**90**] S. K. Bansal, "Towards a Semantic Extract-Transform-Load (ETL) Framework for Big Data Integration," IEEE International Congress on Big Data, June 27 - July 2, pp. 82-89, 2014.

[**91**] A.-R. Bologa and R. Bologa, "A Perspective on the Benefits of Data Virtualization Technology," Informatica Economică, vol. 15, no. 4, pp. 95-104, 2011.

[**92**] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati, "Data Integration in Data Warehousing," International Journal of Cooperative Information Systems, vol. 10, no. 03, pp. 237-271, 2001.

[**93**] J. Ofoeda, R. Boateng, and J. Effah, "Application Programming Interface (API) Research: A Review of the Past to Inform the Future," International Journal of Enterprise Information Systems (IJEIS), vol. 15, no. 3, 2019.

[**95**] I. U. Haq, I. Gondal, P. Vamplew, and S. Brown, "Categorical Features Transformation with Compact One-Hot Encoder for Fraud Detection in Distributed Environment," in Data Mining, Conference paper, Feb. 16, 2019.

[**96**] J. W. Grzymala-Busse, "Data reduction: discretization of numerical attributes," Handbook of Data Mining and Knowledge Discovery, pp. 218-225, Jan. 2002.

[**97**] P. Jindal and D. Kumar, "A Review on Dimensionality Reduction Techniques," in International Journal of Computer Applications, vol. 179, no. 29, pp. 11-17, September 2017.

[**98**] K. Sayood, Introduction to Data Compression. Elsevier, 2017.

[**101**] H.-A. Park, "An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain," Journal of Korean Academy of Nursing, vol. 43, no. 2, pp. 154-164, Apr. 2013.

[**102**] O. Çokluk, "Logistic Regression: Concept and Application," Educational Sciences: Theory & Practice, vol. 10, no. 3, pp. 1397-1407, Summer 2010.

[**103**] S. Menard, "Logistic Regression: From Introductory to Advanced Concepts and Applications," SAGE Publications, Inc., 2010.

[**104**] Y.-Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," Shanghai Arch Psychiatry, vol. 27, no. 2, pp. 130-135, Apr. 2015.

[**105**] H. H. Patel and P. Prajapati, "Study and Analysis of Decision Tree Based Classification Algorithms," Int. J. Comput. Sci. Eng., vol. 6, no. 10, pp. 35-39, Oct. 2018.

[**106**] B. T. Jijo and A. M. Abdulazeez, "Classification Based on Decision Tree Algorithm for," J. Appl. Sci. Technol. Trends, vol. 02, no. 01, pp. 20-28, 2021.

[**107**] S. B. Kotsiantis, "Decision trees: a recent overview," Artificial Intelligence Review, vol. 39, pp. 261-283, 2013.

[**109**] G. Louppe, "Understanding Random Forests: From Theory to Practice," Machine Learning, arXiv preprint arXiv:1407.7502v3, 2015.

[**110**] Y. Liu, Y. Wang, and J. Zhang, "New Machine Learning Algorithm: Random Forest," in Information Computing and Applications, Conference Proceedings, pp. 246-252, 2012.

[**112**] M. Awad and R. Khanna, "Support Vector Machines for Classification," in Efficient Learning Machines, Cham: Springer, pp. 39-66, 2015.

[**113**] H. Xue, Q. Yang, and S. Chen, "SVM: Support Vector Machines," in The Top Ten Algorithms in Data Mining, 1st Edition, Chapman and Hall/CRC, pp. 24, 2009.

[**114**] J. M. Moguerza and A. Muñoz, "Support Vector Machines with Applications," Statist. Sci., vol. 21, no. 3, pp. 322-336, Aug. 2006.

[**116**] I. Rish, "An empirical study of the naive Bayes classifier," in IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 2001.

[**117**] K. P. Murphy, "Naive Bayes classifiers,", 2006.

[**118**] E. M. K. Reddy, A. Gurrala, V. B. Hasitha, and K. V. R. Kumar, "Introduction to Naive Bayes and a Review on Its Subtypes with Applications," in Bayesian Reasoning and Gaussian Processes for Machine Learning Applications, 2022.

[**119**] L. Wang, "Research and Implementation of Machine Learning Classifier Based on KNN," IOP Conference Series: Materials Science and Engineering, vol. 677, no. 5, p. 052038, 2019.

[**120**] M. Steinbach and P.-N. Tan, "kNN: k-Nearest Neighbors," in The Top Ten Algorithms in Data Mining, 1st ed. Chapman and Hall/CRC, 2009.

[**121**] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, "The Evolution of Boosting Algorithms: From Machine Learning to Statistical Modelling," Methods Inf Med, vol. 53, no. 06, pp. 419-427, 2014.

[**122**] A. Natekin and A. Knoll, "Gradient Boosting Machines: A Tutorial," Front. Neurorobot., vol. 7, 2013.

[**123**] Z. He, D. Lin, T. Lau, and M. Wu, "Gradient Boosting Machine: A Survey," arXiv:1908.06951, 2019.

[**125**] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M.A. Janati Idrissi, "Multilayer Perceptron: Architecture Optimization and Training," September 2016.

[**126**] A. V. Joshi, "Perceptron and Neural Networks," in Machine Learning and Artificial Intelligence, pp. 57-72, September 2022.

[**127**] J. Singh and R. Banerjee, "A Study on Single and Multi-layer Perceptron Neural Network," in Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC), IEEE, 2019.

[**130**] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks,"arXiv:1511.08458 [cs.NE], Nov. 2015.

[**131**] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," Pattern Recognition, vol. 77, pp. 354-377, May 2018.

[**132**] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of Training Convolutional Neural Networks," arXiv:1506.01195 [cs.CV], June 2015.

[**133**] J. Wu, "Introduction to Convolutional Neural Networks," LAMDA Group, National Key Lab for Novel Software Technology, Nanjing University, China, May 1, 2017.

[**135**] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview," arXiv:1912.05911 [cs.LG], 2019.

[**136**] L.R. Medsker and L.C. Jain, "Recurrent neural networks," Design and Applications, 2001.

[**137]** Ah Chung Tsoi, "Recurrent neural network architectures: An overview," in Adaptive Processing of Sequences and Data Structures, pp. 1-26, 2006.

[**139**] A. A. Ismail, T. Wood, and H. C. Bravo, "Improving Long-Horizon Forecasts with Expectation-Biased LSTM Networks," 2018.

[**140**] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," arXiv preprint arXiv:1909.09586, Sep. 2019.

[**141**] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," Neural Computation, vol. 31, no. 7, pp. 1235-1270, 2019.

[**142**] A. Graves, "Long Short-Term Memory," in Supervised Sequence Labelling with Recurrent Neural Networks, pp. 37-45, 2012.

[**143**] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," Physica D: Nonlinear Phenomena, vol. 404, pp. 132306, March 2020.

[**146**] Y. Hua, J. Guo, and H. Zhao, "Deep Belief Networks and deep learning," in Proceedings of 2015 International Conference on Machine Learning and Cybernetics, IEEE, 2015.

[**149**] J. Luo, N. L. Solimini, and S. J. Elledge, "Principles of cancer therapy: Oncogene and non-oncogene addiction," Cell, vol. 136, no. 5, pp. 823-837, Mar. 2009.

[**150**] A. L. Hopkins, "Network pharmacology: the next paradigm in drug discovery," Nat. Chem. Biol., vol. 4, no. 11, pp. 682-690, Nov. 2008.

[**151**] D. C. Swinney and J. Anthony, "How were new medicines discovered?," Nat Rev Drug Discov, vol. 10, no. 7, pp. 507-519, Jul. 2011.

[**152**] J. W. Jorgensen, "The many roles of computation in drug discovery," Science, vol. 303, no. 5665, pp. 1813-1818, Mar. 2004.

[**153**] Sneader W. Drug discovery: a history. John Wiley & Sons; 2005.

[**154**] Hansch C, Leo A. Exploring QSAR: Hydrophobic, Electronic, and Steric Constants. American Chemical Society; 1995.

[**155**] Kola I, Landis J. Can the pharmaceutical industry reduce attrition rates? Nat Rev Drug Discov.3(8):711-715 , 2004 .

[**156**] Pocock SJ. Clinical trials: a practical approach. John Wiley & Sons; 1984.

[**157**] Shoichet BK. "Virtual screening of chemical libraries." Nature. 2004 Mar 11; 432(7019):862-5 .

[**158]** I. D. Kuntz, "Structure-based strategies for drug design and discovery," Science, vol. 257, no. 5067, pp. 1078-1082, Jun. 1992.

[**159**] S. Kalyaanamoorthy, Y. P. P. Chen, and Structure-Based Drug Design Consortium, "Structure-based drug design to augment hit discovery," Drug Discovery Today, vol. 24, no. 3, pp. 676-684, 2019.

[**160**] A. Lavecchia, "Machine-learning approaches in drug discovery: Methods and applications," Drug Discovery Today, vol. 20, no. 3, pp. 318-331, 2015.

[**161**] P. Willett, "Similarity searching using 2D structural fingerprints," Methods in Molecular Biology, vol. 361, pp. 61-75, 2006.

[**162**] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," Drug Discovery Today, vol. 23, no. 6, pp. 1241-1250, 2018.

[**163**] A. Lavecchia, "Machine-learning approaches in drug discovery: methods and applications," Drug Discov Today, vol. 20, no. 3, pp. 318-331, Mar. 2015.

[**164**] S. Dara, S. Dhamercherla, S. S. Jadav, C. M. Babu, and M. J. Ahsan, "Machine Learning in Drug Discovery: A Review," Artif Intell Rev, vol. 55, no. 3, pp. 1947-1999, 2022.

[**165**] J. Verma, V. M. Khedkar, and E. C. Coutinho, "3D-QSAR in drug design--a review," Curr. Top. Med. Chem., vol. 10, no. 1, pp. 95-115, 2010.

[**166**] O. Korb, T. Stützle, and T. E. Exner, "Empirical scoring functions for advanced protein-ligand docking with PLANTS," Journal of Chemical Information and Modeling, vol. 49, no. 1, pp. 84-96, 2009.

[**167**] C. M. Venkatachalam, X. Jiang, T. Oldfield, and M. Waldman, "LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites," Journal of molecular graphics and modelling, vol. 21, no. 4, pp. 289-307, 2003.

[**168**] R. E. Amaro and A. J. Mulholland, "Multiscale methods in drug design bridge chemical and biological complexity in the search for new drugs," Nature Reviews Chemistry, vol. 2, no. 10, pp. 0159, 2018.

[**169**] C. Bissantz, G. Folkers, and D. Rognan, "Protein-based virtual screening of chemical databases," Journal of Medicinal Chemistry, vol. 43, no. 22, pp. 4759-4767, 2000.

[**170]** J. Jain, "Surflex-Dock 2.1: robust performance from ligand energetic modeling, ring flexibility, and knowledge-based search," Journal of computer-aided molecular design, vol. 18, no. 7, pp. 551-564, 2004.

[**171**] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell and A. J. Olson, "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility," Journal of Computational Chemistry, vol. 30, no. 16, pp. 2785-2791, Nov. 2009.

[**172**] R. A. Friesner et al., "Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy," Journal of medicinal chemistry, vol. 47, no. 7, pp. 1739-1749, 2004.

[**173**] R. D. Hoffmann, A. Gohier, and P. Pospisil, "Data Mining in Drug Discovery," in Data Mining, vol. 57, WILEY-VCH,2014.

[**174**] F. Grisoni, V. Consonni, and R. Todeschini, "Impact of Molecular Descriptors on Computational Models," Methods Mol Biol, vol. 1825, pp. 171-209, 2018.

[**175**] F. Grisoni, D. Ballabio, R. Todeschini, and V. Consonni, "Molecular Descriptors for Structure-Activity Applications: A Hands-On Approach," Methods Mol Biol, vol. 1800, pp. 3-53, 2018.

[**176**] A.T. Balaban, "Drug Design, Molecular Descriptors in," in Encyclopedia of Complexity and Systems Science, pp. 2196-2215.

[**177**] A. Cereto-Massagué, M.J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé, G. Pujadas, "Molecular fingerprint similarity search in virtual screening," Methods, vol. 71, no. C, pp. 58-63, 2015.

[**178**] G. Huang, F. Yan and D. Tan, "A Review of Computational Methods for Predicting Drug Targets," Curr. Protein Pept. Sci., vol. 19, no. 6, pp. 562-572, 2018

[**179**] J. Jiang and Y. Zhang, "A Survey of Computational Methods for Predicting Protein-Ligand Inhibition," Curr. Drug Metab., vol. 19, no. 6, pp. 470-479, 2018.

[**180**] L. Jia and H. Gao, "Machine Learning for In Silico ADMET Prediction," Methods Mol Biol, vol. 2390, pp. 447-460, 2022.

[**181**] T. Zhu, S. Cao, P.-C. Su, R. Patel, D. Shah, H. B. Chokshi, R. Szukala, M. E. Johnson, and K. E. Hevener, "Hit Identification and Optimization in Virtual Screening: Practical Recommendations Based on a Critical Literature Analysis," J. Med. Chem., vol. 56, no. 17, pp. 6560–6572, 2013.

[**182**] T. I. Oprea and H. Matter, "Integrating virtual screening in lead discovery," Current Opinion in Chemical Biology, vol. 8, no. 4, pp. 349-358, Aug. 2004.

[**183**] H. Afzaal, R. Altaf, U. Ilyas, S. U. Zaman, S. D. A. Hamdani, S. Khan, H. Zafar, M. M. Babar, and Y. Duan, "Virtual screening and drug repositioning of FDA-approved drugs from the ZINC database to identify the potential hTERT inhibitors," Front. Pharmacol., vol. 13, Nov. 2022, Art. no. 848249.

[**184**] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and QSAR modeling," J. Chem. Inf. Comput. Sci., vol. 43, no. 6, pp. 1947-1958, 2003.

[**185**] M. Song and Z. Jiang, "Inferring Association between Compound and Pathway with an Improved Ensemble Learning Method," Mol. Inform., vol. 34, no. 11-12, pp. 753-760, Nov. 2015.

[**186**] H. Singh, S. Singh, D. Singla, S.M. Agarwal, and G.P.S. Raghava, "QSAR based model for discriminating EGFR inhibitors and non-inhibitors using Random forest," Biol Direct, vol. 10, no. 1, pp. 1-12, 2015.

[**187**] P. Mistry, D. Neagu, P. R. Trundle, and J. D. Vessey, "Using random forest and decision tree models for a new vehicle prediction approach in computational toxicology," Soft Computing, vol. 20, no. 8, pp. 2967-2979, 2016.

[**188**] S. Ai, Y. Bai, and X. Liu, "Virtual Screening for COX-2 Inhibitors with Random Forest Algorithm and Feature Selection," in ICBRA '17: Proceedings of the 4th International Conference on Bioinformatics Research and Applications, Dec. 2017, pp. 9-14.

[**189**] A. P. Lind and P. C. Anderson, "Predicting drug activity against cancer cells by random forest models based on minimal genomic information and chemical properties," PloS one, vol. 14, no. 7, p. e0219774, 2019.

[**190**] R. Rahman, S. R. Dhruba, S. Ghosh, and R. Pal, "Functional random forest with applications in dose-response predictions," Scientific reports, vol. 9, no. 1, pp. 1-14, 2019.

[**191**] S. Ahn, S. Lee, and M. H. Kim, "Random-forest model for drug–target interaction prediction via Kullback–Leibler divergence," J Cheminform, vol. 14, no. 1, p. 67, 2022

[**192**] J. M. Kriegl, T. Arnhold, B. Beck, and T. Fox, "Prediction of human cytochrome P450 inhibition using support vector machines," QSAR Comb. Sci., vol. 24, no. 6, pp. 807-811, Jun. 2005, doi: 10.1002/qsar.200430925.

[**193**] E. P. Kondratovich, N. I. Zhokhova, I. I. Baskin, V. A. Palyulin, and N. S. Zefirov, "Fragmental descriptors in (Q)SAR: prediction of the assignment of organic compounds to pharmacological groups using the support vector machine approach," Russ. Chem. Bull., vol. 59, no. 4, pp. 738-746, Apr. 2010

[**194**] X. H. Liu, X. H. Ma, C. Y. Tan, Y. Y. Jiang, M. L. Go, B. C. Low⊥, and Y. Z. Chen, "Virtual Screening of Abl Inhibitors from Large Compound Libraries by Support Vector Machines," J. Chem. Inf. Model., vol. 49, no. 9, pp. 2101-2110, 2009.

[**195**] X. Dong, C. Jiang, H. Hu, J. Yan, J. Chen, Y. Hu, "QSAR study of Akt/protein kinase B (PKB) inhibitors using support vector machine," European Journal of Medicinal Chemistry, vol. 44, no. 11, pp. 4499-4505, 2009.

[**196**] X. H. Ma, R. Wang, C. Y. Tan, Y. Y. Jiang, T. Lu, H. B. Rao, X. Y. Li, M. L. Go, B. C. Low, and Y. Z. Chen, "Virtual Screening of Selective Multitarget Kinase Inhibitors by Combinatorial Support Vector Machines," Mol. Pharmaceutics, vol. 7, no. 5, pp. 1545-1560, 2010.

[**197**] A. M. Wassermann, H. Geppert, and J. Bajorath, "Application of support vector machine-based ranking strategies to search for target-selective compounds," in Methods Mol. Biol., vol. 672, Springer, pp. 517-530, 2011.

[**198**] G. Hinselmann, L. Rosenbaum, A. Jahn, N. Fechner, C. Ostermann, and A. Zell, "Large-scale learning of structure-activity relationships using a linear support vector machine and problem-specific metrics," J. Chem. Inf. Model., vol. 51, no. 2, pp. 203–213, Feb. 2011.

[**199**] Z. Shi, X.H. Ma, C. Qin, J. Jia, Y.Y. Jiang, C.Y. Tan, Y.Z. Chen, "Combinatorial support vector machines approach for virtual screening of selective multi-target serotonin reuptake inhibitors from large compound libraries", Journal of Molecular Graphics and Modelling, vol. 32, pp. 49-66, Feb. 2012.

[**200**] N. Poorinmohammad, H. Mohabatkar, M. Behbahani, and D. Biria, "Computational prediction of anti HIV-1 peptides and in vitro evaluation of anti HIV-1 activity of HIV-1 P24-derived peptides," J. Pept. Sci., vol. 21, no. 1, pp. 10-16, Jan. 2015.

[**201**] M. Fernandez, S. Ahmad, J.I. Abreu, and A. Sarai, "Large-scale recognition of high-affinity protease–inhibitor complexes using topological autocorrelation and support vector machines," BMC Bioinformatics, vol. 16, no. 1, pp. 420-433, Jul. 2015.

[**202**] X. Zhang and E. A. Amin, "Highly predictive support vector machine (SVM) models for anthrax toxin lethal factor (LF) inhibitors," J. Mol. Graph. Model., vol. 63, pp. 22-28, Jan. 2016 .

[**203**] A. Onay, M. Onay and O. Abul, "Classification of nervous system withdrawn and approved drugs with ToxPrint features via machine learning strategies," Comput Methods Programs Biomed, vol. 142, pp. 9-19, 2017.

[**204**] J. Che, L. Chen, Z.-H. Guo, S. Wang, and Aorigele, "Drug Target Group Prediction with Multiple Drug Networks," Comb. Chem. High Throughput Screen, vol. 23, no. 4, pp. 274-284, 2019.

[**205**] Y. Wei, W. Li, T. Du, Z. Hong, and J. Lin, "Targeting HIV/HCV coinfection using a machine learning-based multiple quantitative structure-Activity Relationships (Multiple QSAR) Method," Int. J. Mol. Sci., vol. 20, no. 14, 2019.

[**206**] J. Fang, R. Yang, L. Gao, D. Zhou, S. Yang, A. Liu, and G. Du, "Predictions of BuChE Inhibitors Using Support Vector Machine and Naive Bayesian Classification Techniques in Drug Discovery," J. Chem. Inf. Model., vol. 53, no. 11, pp. 3009-3020, Oct. 2013.

[**207**] L. Wang, L. Chen, Z. Liu, M. Zheng, Q. Gu, and J. Xu, "Predicting mTOR Inhibitors with a Classifier Using Recursive Partitioning and Naïve Bayesian Approaches," PLoS One, vol. 9, no. 5, May 2014.

[**208**] W. Lian, J. Fang, C. Li, X. Pang, A.-L. Liu and G.-H. Du, "Discovery of Influenza A virus neuraminidase inhibitors using support vector machine and Naïve Bayesian models," Mol. Divers., vol. 20, no. 2, pp. 439-451, May 2016.

[**209**] H. Zhang, P. Yu, J.-X. Ren, X.-B. Li, H.-L. Wang, L. Ding, and W.-B. Kong, "Development of novel prediction model for drug-induced mitochondrial toxicity by using naïve Bayes classifier method," Food Chem. Toxicol., vol. 110, pp. 122-129, Dec. 2017.

[**210**] D. Kang, X. Pang, W. Lian, L. Xu, J. Wang, H. Jia, B. Zhang, A.-L. Liu, and G.-H. Du, "Discovery of VEGFR2 inhibitors by integrating naïve Bayesian classification, molecular docking and drug screening approaches," RSC Advances, vol. 8, no. 8 2018.

[**211**] Madhukar NS, Khade PK, Huang L, Gayvert K, Galletti G, Stogniew M, et al. "A Bayesian machine learning approach for drug target identification using diverse data types." Nat Commun [Internet] 10(1):1–14. 2019.

[**212**] A. L. Perryman, J. S. Patel, R. Russo, E. Singleton, S. Ekins, J. S. Freundlich et al., "Naive Bayesian Models for Vero Cell Cytotoxicity," HHS Public Access, vol. 35, no. 9, pp. 170, 2020.

[**213**] H. Zhang, C.-T. Liu, J. Mao, C. Shen, R.-L. Xie, and B. Mu, "Development of novel in silico prediction model for drug-induced ototoxicity by using naïve Bayes classifier approach," Toxicology in Vitro, vol. 65, pp. 104812, Jun. 2020.

[**214**] I. B. Mustapha and F. Saeed, "Bioactive Molecule Prediction Using Extreme Gradient Boosting," Molecules, vol. 21, no. 8, p. 983, 2016.

[**215**] T. Lei, H. Sun, Y. Kang, F. Zhu, H. Liu, W. Zhou, Z. Wang, D. Li, Y. Li, and T. Hou, "ADMET Evaluation in Drug Discovery. 18. Reliable Prediction of Chemical-Induced Urinary Tract Toxicity by Boosting Machine Learning Approaches," Mol. Pharmaceutics, vol. 14, no. 11, pp. 3935-3953, Nov. 2017.

[**216**] P. Xuan, C. Sun, T. Zhang, Y. Ye, T. Shen and Y. Dong, "Gradient Boosting Decision Tree-Based Method for Predicting Interactions Between Target Genes and Drugs," Front. Genet., vol. 10, pp. 1-8, May 2019.

[**217**] D. Gavrilev, N. Amangeldiuly, S. Ivanov, and E. Burnaev, "High Performance of Gradient Boosting in Binding Affinity Prediction," arXiv:2205.07023, May 2022.

[**218**] E. N. Grafskaia, E. R. Pavlova, I. A. Latsis, M. V. Malakhova, D. V. Ivchenkov, P. V. Bashkirov, E. F. Kot, K. S. Mineev, A. S. Arseniev, D. V. Klinov, and V. N. Lazarev, "Non-toxic antimicrobial peptide Hm-AMP2 from leech metagenome proteins identified by the gradient-boosting approach," Materials & Design, vol. 224, pp. 111364, Dec. 2022.

[**219**] Y. Liang and X. Ma, "iACP-GE: accurate identification of anticancer peptides by using gradient boosting decision tree and extra tree," SAR and QSAR in Environmental Research, vol. 34, no. 1, pp. 1-19, Dec. 2022.

[**220**] M. Yarmolenko and B. J., "Extreme Gradient Boosting Algorithm Classification for Predicting Lifespan-Extending Chemical Compounds," Research Square, Oct. 2022.

[**221**] G.E. Dahl, N. Jaitly, and R. Salakhutdinov, "Multi-task neural networks for QSAR predictions," 2014.

[**222**]J. Ma, R.P. Sheridan, A. Liaw, G.E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships," J. Chem. Inf. Model., vol. 55, no. 2, pp. 263-274, Feb. 2015.

[**223**] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, "DeepTox (2016) Toxicity prediction using deep learning," Front. Environ. Sci., vol. 3, Feb. 2016.

[**224**] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov, "Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data," Mol. Pharm., vol. 13, no. 7, pp. 2524-2530, 2016.

[**225**] K. Wu and G.-W. Wei, "Quantitative Toxicity Prediction Using Topology Based Multitask Deep Neural Networks," J. Chem. Inf. Model., vol. 58, no. 2, pp. 520-531, 2018.

[**226**] K. Lee and D. Kim, "In-Silico Molecular Binding Prediction for Human Drug Targets Using Deep Neural Multi-Task Learning," Genes, vol. 10, no. 11, p. 906, Nov. 2019.

[**227**] Z. Sun, S. Zheng, H. Zhao, Z. Niu, Y. Lu, Y. Pan and Y. Yang, "To Improve Prediction of Binding Residues With DNA, RNA, Carbohydrate, and Peptide Via Multi-Task Deep Neural Networks," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 19, no. 6, pp. 3735-3743, Nov.-Dec. 2022.

[**228**] M. Hofmarcher, A. Mayr, E. Rumetshofer, P. Ruch, P. Renz, J. Schimunek, P. Seidl, A. Vall, M. Widrich, S. Hochreiter, and G. Klambauer, "Large-scale ligand-based virtual screening for SARS-CoV-2 inhibitors using deep neural networks," arXiv:2004.00979 , 2020.

[**229**] B. Sharma, V. Chenthamarakshan, A. Dhurandhar, S. Pereira, J. A. Hendler, J. S. Dordick and P. Das, "Accurate Clinical Toxicity Prediction using Multi-task Deep Neural Nets and Contrastive Molecular Explanations," arXiv:2204.06614 [q-bio.QM], 2022.

[**230**] R. Liu, S. Laxminarayan, J. Reifman and A. Wallqvist, "Enabling data-limited chemical bioactivity predictions through deep neural network transfer learning," J. Comput. Aided Mol. Des., vol. 36, pp. 867-878, 2022.

[**231**] D. K. Duvenaud, D. Maclaurin, J. A. Iparraguirre, R. G. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in Advances in Neural Information Processing Systems, pp. 2224-2232, , 2015.

[**232**] M. Defferrard et al., "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering," in Advances in Neural Information Processing Systems, vol. 29, pp. 3844-3852, 2016.

[**233**] C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola, and K. F. Jensen, "Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction," J. Chem. Inf. Model., vol. 57, no. 8, pp. 1757-1772, Aug. 2017 .

[**234**] I. Wallach, M. Dzamba, and A. Heifets, "AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery," arXiv preprint arXiv:1510.02855, 2015.

[**235**] T. Shi et al., "Molecular image-based convolutional neural network for the prediction of ADMET properties," Chemom. Intell. Lab. Syst., vol. 194, p. 103853, Sep. 2019 .

[**236**] M. Fernandez, F. Ban, G. Woo, M. Hsing, T. Yamazaki, E. Leblanc, et al., "Toxic Colors: The Use of Deep Learning for Predicting Toxicity of Compounds Merely from Their Graphic Images," J. Chem. Inf. Model, vol. 58, no. 8, pp. 1533-1543, Aug. 2018.

[**237**] B. Rao, L. Zhang, and G. Zhang, "ACP-GCN: The Identification of Anticancer Peptides Based on Graph Convolution Networks," in IEEE Access, vol. 8, pp. 176005-176011, 2020.

[**238**] M. Kumari and N. Subbarao, "Deep learning model for virtual screening of novel 3C-like protease enzyme inhibitors against SARS coronavirus diseases," in Computational Biology and Medicine, vol. 132, pp. 104317, May 2021.

[**239**] G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas and N. Baker, "Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models," arXiv preprint arXiv:1706.06689, 2017.

[**240**] M. Tsubaki, K. Tomii and J. Sese, "Compound-Protein Interaction Prediction with End-to-End Learning of Neural Networks for Graphs and Sequences," in Bioinformatics, vol. 35, no. 2, pp. 309-318, Jan. 2019.

[**241**] A. Fitriawan, I. Wasito, A. F. Syafiandini, M. Amien, and A. Yanuar, "Multi-label classification using deep belief networks for virtual screening of multi-target drug," 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), , pp. 57-61. Oct. 2016.

[**242**] A. Fitriawan, I. Wasito, A. F. Syafiandini, M. Amien, and A. Yanuar, "Deep belief networks using hybrid fingerprint feature for virtual screening of drug design," in 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 436-441, Oct. 2016.

[**243**] F. Ghasemi, A. Mehridehnavi, A. Fassihi and H. Pérez-Sánchez, "Deep neural network in QSAR studies using deep belief network," Applied Soft Computing, vol. 62, pp. 251-258, Jan. 2018.

[**244**] S. A. Hooshmand, S. A. Jamalkandi, S. M. Alavi, and A. Masoudi-Nejad, "Distinguishing drug/non-drug-like small molecules in drug discovery using deep belief network," Molecular Diversity, vol. 25, pp. 827-838, Mar. 2020.

[**245**] L. Yu, X. Shi, S. Tian, S. Gao and L. Li, "Classification of Cytochrome P450 1A2 Inhibitors and Noninhibitors Based on Deep Belief Network," in International Journal of Computational Intelligence and Applications, vol. 16, no. 1, pp. 1750002, 2017.

[**246**] A. Shakya, B. Joshi, U. K. Yadav, and O. P. Mahato, "Drug-target interaction prediction using deep belief network," International Journal of Bioinformatics Research and Applications, vol. 18, no. 5, pp. 479-495, Jan. 2022.

[**247**] A. Lusci, G. Pollastri, and P. Baldi, "Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules," J. Chem. Inf. Model, vol. 53, no. 7, pp. 1563-1575, Jul. 2013 .

[**248**] E.J. Bjerrum, "SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules," arXiv preprint arXiv:1703.07076, 2017.

[**249**] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, "Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks," ACS Cent. Sci., vol. 4, no. 1, pp. 120-131, Dec. 2017.

[**250**] P. Ruiz Puentes, N. Valderrama, C. González, L. Daza, C. Muñoz-Camargo, J. C. Cruz, and P. Arbeláez, "PharmaNet: Pharmaceutical discovery with deep recurrent neural networks," PLOS One, vol. 16, no. 4, p. e0241728, Apr. 2021.

[**251**] M. V. S. Santana and F. P. Silva-Jr, "De novo design and bioactivity prediction of SARS-CoV-2 main protease inhibitors using recurrent neural network-based transfer learning," BMC Chemistry, vol. 15, no. 1, p. 8, Jan. 2021.

[**252**] L. Benarous, K. Benarous, G. Muhammad, and Z. Ali, "Deep learning application detecting SARS-CoV-2 key enzymes inhibitors," Journal of Ambient Intelligence and Humanized Computing, pp. 1-10, Jul. 2022.

[**253**] B. Alberts, A. Johnson, J. Lewis, et al., "The Cell Cycle," in Molecular Biology of the Cell, 4th ed., New York: Garland Science, ch. 17. 2002.

[**254**] G. M. Cooper, "The Cell: A Molecular Approach," 2nd ed. Sunderland (MA): Sinauer Associates, 2000.

[**257**] M. Malumbres and M. Barbacid, "Cell cycle, CDKs and cancer: a changing paradigm," Nature Reviews Cancer, vol. 9, no. 3, pp. 153-166, Mar. 2009.

[**258**] A. Goga, D. Yang, A. D. Tward, D. O. Morgan, and J. M. Bishop, "Inhibition of CDK1 as a potential therapy for tumors over-expressing MYC," Nat Med, vol. 13, no. 7, pp. 820-827, Jul. 2007

[**259**] I. Mendolia, S. Contino, U. Perricone, R. Pirrone, and E. Ardizzone, "A Convolutional Neural Network for Virtual Screening of Molecular Fingerprints," in Image Analysis and Processing – ICIAP 2019, vol. 11752, pp. 399-409, Springer International Publishing, 2019.

[**260**] Y. Zhou, H. Jiang and Y. Zhang, "Liver Tumor Image Enhancement and CDK1 Gene Mutation Prediction Method," 2020 2nd International Conference on Intelligent Medicine and Image Processing (IMIP), pp. 1-7, April 2020.

[**261**] Y. H. Lee and G.-S. Yi, "Prediction of Novel Anoctamin1 (ANO1) Inhibitors Using 3D-QSAR Pharmacophore Modeling and Molecular Docking," Int. J. Mol. Sci., vol. 19, no. 10, p. 3204, 2018.

[**262**] Xiao-Yu Qing, Xiao Yin Lee, and Joren De Raeymaecker. "Pharmacophore modeling: Advances, Limitations, And current utility in drug discovery." Journal of Receptor, Ligand and Channel Research, vol. 7 , pp. 81-92, November 2014.

[**269**] G. Sliwoski, S. Kothiwale, J. Meiler, and E. W. Lowe Jr., "Computational Methods in Drug Discovery," Pharmacological Reviews, vol. 66, no. 1, pp. 334-95, Dec. 2013.

[**273**] M. M. Rodríguez-Hernández, R. E. Pruneda, and J. M. Rodríguez-Díaz, "Statistical Analysis of the Evolutive Effects of Language Development in the Resolution of Mathematical Problems in Primary School Education," May 2021.

[**274**] S. Hadiby and Y. M. Ben Ali, "Deep Learning Based-Virtual Screening Using 2D Pharmacophore Fingerprint in Drug Discovery," Neural Processing Letters, pp. 1-12, May 2022.

[**275**] S. Hadiby and Y. M. Ben Ali, "FNN Based-Virtual Screening Using 2D Pharmacophore Fingerprint for Activity Prediction in Drug Discovery," Int. J. Comput. Intell. Appl., vol. 21, no. 3, pp. 2250019, Sep. 2022.

[**281**] C. F. Karouzos and K. Φ. Καρούζος, "Unsupervised Domain Adaptation for Natural Language Processing," Computer Science, 2020.

[286] Z. Na, S. Pan, M. Uttamchandani, and S. Q. Yao, "Protein-Protein Interaction Inhibitors of BRCA1 Discovered Using Small Molecule Microarrays," Methods Mol Biol, vol. 1518, pp. 139-156, 2017.

[287] P.S. Obermiller, D.L. Tait, and J.T. Holt, "Gene therapy for carcinoma of the breast: Therapeutic genetic correction strategies," Breast Cancer Res, vol. 2, no. 1, pp. 28-31, 2000.

[288] E. Paplomata and R. O'Regan, "The PI3K/AKT/mTOR pathway in breast cancer: targets, trials and biomarkers," Adv Med Oncol, vol. 6, no. 4, pp. 154-166, Jul 2014.

[**289**] PARP Inhibition in BRCA-Mutant Breast Cancer. Cancer. 2018 Jun 15; 124(12): 2498–2506.

Webography

[5] https://www.sciencefacts.net/atom-2.html

[**6**] https://energywavetheory.com/atoms.

[**7**] https://www.cancer.gov/publications/dictionaries/cancer-terms/def/molecule

[**8**] Club Roots, "Caffeine," Club Roots, 2019. [Online]. Available: https://www.clubroots.com/fr/blogs/ingredients/caffeine. [Accessed: May 7, 2023].

[9] https://www.britannica.com/science/atom/Atomic-bonds

[**10**] https://www.inspiritvr.com/general-chemistry/ionic-and-metallic-bonding/ionic-bonding-study-guide.

[**11**] https://www.chemistrylearner.com/chemical-bonds/covalent-bond.

[**12**] https://byjus.com/question-answer/what-are-some-examples-of-metallic-compounds.

[**13**] https://www.geeksforgeeks.org/hydrogen-bonding.

[14] https://www.toppr.com/guides/biology/molecular/macromolecules-definition-types-examples/].

[15] https://www.genome.gov/genetics-glossary/DNA-Sequencing

[**16**] https://fr.wikipedia.org/wiki/Acide_d%C3%A9soxyribonucl%C3%A9ique.

[**18**] https://www.biochemithon.in/biology/dna-replication.

[**20**] https://www.mun.ca/biology/scarr/Expressed_Sequence_Tags.html.

[**23**] https://www.toppr.com/guides/biology/difference-between/dna-and-rna/]:

[**28**] https://atdbio.com/nucleic-acids-book/Transcription-Translation-and-Replication

[**29**] https://www.biology-questions-and-answers.com/protein-synthesis.html

[**30**] https://www.thoughtco.com/protein-function-373550

[**34**] https://byjus.com/chemistry/protein-structure-and-levels-of-protein

[**36**] https://study.com/academy/lesson/what-is-a-metabolic-pathway-definition-example.html

Webography

[42] https://www.heighpubs.org/jfsr/jfsr-aid1040.php

[47] https://www.talend.com/resources/what-is-data-mining.

[**52**] https://www.tutorialandexample.com/supervised-machine-learning.

[55] https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning.

[56] https://machinelearningmastery.com/types-of-classification-in-machine-learning.

[60] https://www.analyticsvidhya.com/blog/2021/07/demystifying-the-difference-between-multi-class-and-multi-label-classification-problem-statements-in-deep-learning/

[**62**] https://www.kdnuggets.com/2020/01/5-most-useful-techniques-handle-imbalanced-datasets.html.

[63] https://medium.com/eni-digitalks/imbalanced-data-an-extensive-guide-on-how-to-deal-with-imbalanced-classification-problems-6c8df0bc2cab.

[67] https://serokell.io/blog/polynomial-regression-analysis.

[**69**] https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression.

[**70**] https://analystprep.com/study-notes/cfa-level-2/quantitative-method/multiple-regression-equation.

[71] https://www.investopedia.com/terms/m/mlr.asp

[**73**] https://www.linkedin.com/pulse/lasso-regression-clearly-explained-bhabani-shankear-basak.

[**74**] https://machinelearningjourney.com/index.php/2020/02/13/ridge-regression.

[**75**] https://www.javatpoint.com/unsupervised-machine-learning.

[78] https://towardsdatascience.com/the-eclat-algorithm-8ae3276d2d17

[**79**] https://comidoc.net/udemy/association.

[**94**] https://www.simplilearn.com/what-is-data-standardization-article.

[**96**] https://www.c-sharpcorner.com/article/a-quick-overview-of-machine-learning-tasks/

Webography

[**100**] https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c.

[**108**] https://medium.com/analytics-vidhya/breast-cancer-risk-prediction-system-a575c2eb8130

[**111**] https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest.

[**115**] https://datatron.com/what-is-a-support-vector-machine

[**124**] https://anupkrsingh.medium.com/gradient-boosting-in-machine-learning-8446c8d31bb1

[**128**] https://www.geeksforgeeks.org/hidden-layer-perceptron-in-tensorflow/

[**129**] https://www.baeldung.com/cs/mlp-vs-dnn

[**134**] https://developersbreach.com/convolution-neural-network-deep-learning

[**138**] https://pub.towardsai.net/whirlwind-tour-of-rnns-a11effb7808f

[**144**] https://datascience.fm/deep-belief-network/

[**145**] https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-belief-network

[**147**] https://iq.opengenus.org/deep-belief-network

[**148**] https://www.toppr.com/guides/chemistry/chemistry-in-everyday-life/drugs-target-interaction/

[**255**] https://www.gettyimages.fr/illustrations/human-cell

[**256**] https://le.ac.uk/vgec/topics/cell-cycle/the-cell-cycle-schools-and-colleges.

[**263**] IUPAC Gold Book-Hydrogen Bond Donor: https://goldbook.iupac.org/terms/view/H02889

[**264**] IUPAC Gold Book- Hydrogen Bond Acceptor: https://goldbook.iupac.org/terms/view/H02890

[**265**] IUPAC Gold Book - Anion: https://goldbook.iupac.org/terms/view/A00245

[**266**] IUPAC Gold Book - Cation: https://goldbook.iupac.org/terms/view/C00907

[**267**] IUPAC Gold Book - hydrophobicity : https://goldbook.iupac.org/terms/view/HT06964

Webography

[**268**] https://blog.iglcoatings.com/the-science-of-hydrophobicity

[**270**] https://www.statology.org/how-to-read-chi-square-distribution-table

[**271**] https://www.ebi.ac.uk/chembl

[**272**] https://pubchem.ncbi.nlm.nih.gov

[**276**] https://statisticsbyjim.com/hypothesis-testing/f-table

[**277**] https://lemondeetnous.cafe-sciences.org/2015/05/comprendre-le-cancer-du-sein-partie-iii/

[**278**] https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

[**279**] https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964

[**280**] https://www.width.ai/post/neural-collaborative-filtering

[**282**] https://www.maxicours.com/se/cours/conformation-et-aspect-energetique

[**283**] https://www.britannica.com/science/colloid

[**284**] https://chem.libretexts.org/Courses/Sacramento_City_College/SCC%3A_Chem_420_-_Organic_Chemistry_I/Text/02%3A_Structure_and_Properties_of_Organic_Molecules/2.06%3A_Bond_Rotation

[**285**] https://www.cliffsnotes.com/study-guides/chemistry/organic-chemistry-i/structure-of-organic-molecules/free-rotation-around-single-bonds

Publications in international journals

Publication 1

S. Hadiby and Y. M. Ben Ali, "Deep Learning Based-Virtual Screening Using 2D Pharmacophore Fingerprint in Drug Discovery," **Neural Processing Letters**, pp. 1-12, May 2022, doi: 10.1007/s11063-022-10879-6.

Informations about Neural Processing Letters journal

ISSN: 13704621, 1573773X
Web Site: https://www.springer.com/journal/11063
H Index :54
Impact Factor: 2.565

Publication 2

S. Hadiby and Y. M. Ben Ali, "FNN Based-Virtual Screening Using 2D Pharmacophore Fingerprint for Activity Prediction in Drug Discovery," Int. J. Comput. Intell. Appl., vol. 21, no. 3, pp. 2250019, Sep. 2022, doi: 10.1142/S1469026822500195.

Informations about International Journal of Computational Intelligence and Applications

ISSN: 1469-0268 (print),1757-5885 (online)
Web Site: https://www.worldscientific.com/loi/ijcia
H Index : 21

papers submitted

Paper1

Title: CNN-based Virtual Screening using 3D Pharmacophore Fingerprint for Activity Prediction of Molecules with CDK1 Gene

Paper2

Title: Integrating Pharmacophore Model and Deep Learning for Activity Prediction of Molecules with BRCA1 Gene