الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

YAHIA FARES UNIVERSITY OF MEDEA

FACULTY OF SCIENCES

Department of Mathematics and Computer Science



A Thesis submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Science

Specialty: System Engineering and Web Technology

De novo Molecular Design based on Deep Learning

Proposed and Directed by

Dr. *Hentabli Hamza*

Prepared and Presented by

Ms. *Benfridja Ines*

Ms. *Mazari Noudjoud*

2021/2022

Acknowledgment

As a preamble to this thesis, we thank ALLAH the most gracious, the most merciful who guides us and gives us patience and courage during these long years of study.

We address our sincere thanks, recognitions and gratitude to all of our teachers, and in particular our advisor of this thesis Dr.Hentabli Hamza, at Yahia fares University for sharing knowledge and passion with us along this journey.

We thank all the members of the jury for agreeing to review and judge our graduation work and enrich it with their suggestions and proposals.

We also extend our gratitude to our families for their endless support, patience and inspiration during the long journey, They have always been and will continue to be our supporters and comfort zone during our academic journey.

Also thanks go to the ones who have contributed in accomplishing this work, either from near or far, may Allah grant them all the best.

We also show our sincere thanks to our professors who provided us their knowledge and experience during our all these years.

Dedication

With the help of God Almighty, who has set us on the path of our life:

This thesis is wholeheartedly dedicated to my beloved

family.

To my parents ***Fatiha and Sedik***, my source of inspiration.

Especially to my mother *Benali Fatiha* , the woman who dedicated herwhole life to raise me and kept affording me all her best and prayers.

To my beloved grandmother,my hero, may God bless

her.

To my cousin Fella who kept helping me during this

period.

To my bestfriends Wafa,Rafika and

maroua.To my friend Ryn, for her

emotional support.

To my partner and friend Noudjoud.

may God protect you and give you a long life full of health and ofhappiness.

Benfridja Ines

Dedication

In the name of merciful God, thank you very much to him, for his help and the will he gave us to overcome all the obstacles and difficulties during our years of study and for having enlightened our path in order to do this work.

To my mother and my father, who was always standing with me, encouraging me to pursue the path of knowledge, pushing me forward and inspiring me all the time .no dedication can express my respect, my eternal love and my consideration for the sacrifices you have made for my education and my well being. May God, the Most High, grant you health, happiness and long life .

To all my sisters and brothers one by one, who remember me by their prayers and pushing me to do my best and have always been by my side.

A special thanks to my sisters sons akrem ,ranim, loai and racim.

Likewise ,i extend my respectful thanks to the members of the jury, who have done the honor to participate in this jury and to examine this work.

To all my friends and my partner Ines, thanks for all the good times we spent together and all those who we forgot to mention, the mind forgets but the heart never does.

Noudjoud Mazari

Abstract

Chemoinformatics research involves the application of computer and information systems to problems in the field of chemistry. Molecular design strategies are integral to therapeutic progress in drug discovery. Computational approaches for de novo molecular design have been developed over the past three decades and, recently, thanks in part to advances in machine learning(ML) and artificial intelligence(AI), the drug discovery field has gained practical experience. Here, we explore the capability of deep learning to generate synthetically similar molecules with drug-like properties.

Keywords: de novo design, Generative models, molecular representation, artificial intelligence, molecular design, generative adversarial networks, deep learning.

ملخص

تتضمن أبحاث الكيمياء المعلوماتية تطبيق أنظمة الكمبيوتر والمعلومات على مشاكل في مجال الكيمياء.
استراتيجيات التصميم الجزيئي جزء لا يتجزأ من التقدم العلاجي في اكتشاف الأدوية. تم تطوير المناهج الحسابية للتصميم دي نوفو الجزيئي خلال العقود الثلاثة الماضية ومؤخراً بفضل التقدم في التعلم الآلي والذكاء الاصطناعي جزئياً ، اكتسب مجال اكتشاف الأدوية خبرة عملية . هنا ، نستكشف قدرة الذكاء الآلي للتعلم العميق لتوليد جزيئات متشابهة اصطناعيًا بخصائص تشبه الأدوية.

الكلمات الرئيسية نماذج مولدة ، تمثيل جزيئي ، ذكاء اصطناعي ، تصميم جزيئي ، شبكات خصومة مولدة ، تعلم عميق.

Résumé

La recherche en chimoinformatique implique l'application de systèmes informatiques et d'information à des problèmes dans le domaine de la chimie. Les stratégies de conception moléculaire font partie intégrante des progrès thérapeutiques dans la découverte de médicaments. Des approches informatiques pour la conception moléculaire de novo ont été développées au cours des trois dernières décennies et, récemment, en partie grâce aux progrès de l'apprentissage automatique (ML) et de l'intelligence artificielle (IA), le domaine de la découverte de médicaments a acquis une expérience pratique. Ici, nous explorons la capacité d'un d'apprentissage en profondeur à générer des molécules synthétiquement similaires avec des propriétés de type médicament.

Mots clés : conception de novo, modèles génératifs, représentation moléculaire, intelligence artificielle, conception moléculaire, réseaux génératifs antagonistes, apprentissage profond.

Bibliography

General Introduction

Chapter 1: Introduction

Introduction..... 1

Problem Background..... 2

1.1. Problem Statement..... 5

1.2. The Research Question.....6

1.3. Objectives of the Research..... 6

1.4. Scope of the Study.....7

1.5. Thesis Outline… 7

1.6. Summary…..... 8

Chapter 2: Literature Review

Introduction.....10

2.1. Chemoinformatics… 11

2.2. Molecular Representations… 12

2.2.1. String Representations…..... 13

2.2.2. Chemical Table Representations.....14

2.2.2.1. MDL Molfile… 14

2.2.2.2. Graph Representation 15

2.2.2.3. Extended-connectivity fingerprints… 17

2.2.2.4. Molecule generation in 3D 18

2.3. Molecular similarity measurement 19

2.3.1. Introduction..... 19

2.3.2. Similarity and distance…..... 19

2.3.3. Analysis of Similarity Measurement for Molecular Structure…..... 20

2.3.4. Conclusion..... 21

2.4 Molecular Biological Activity 22

2.5. Molecular Generation 23

2.5.1. Molecular Generation Using Computing 24

2.5.1.1. RNN-Based Generative Models 24

2.5.1.2. Graph-Based Generative Models… 25

2.5.1.3. SMILES-Based Generative Models 26

2.6. Deep learning 26

2.6.1. Artificial Intelligence (AI)..... 26

2.6.2. Machine Learning (ML)… 27

2.6.3. Deep Learning (DL)…..... 27

2.6.4. Convolutional Neural Network (CNN)…..... 28

2.6.4.1. Neural Networks… 28

2.6.4.2. Convolutional Neural Network… 30

2.6.5. Recurrent Neural Network…..... 31

2.6.6. AutoEncoder… 32

2.6.6.1. Variational AutoEncoder..... 33

2.6.7. Graph Neural Networks 33

2.7.7 .1. What is a Graph? 33

2.7.4 .2. Why is it hard to analyze a graph? 34

2.7.4 .3. Graph Neural Networks (GNNs)… 35

2.7.4 .4. Applications of GNNs…..... 35

2.6.8. Challenges… ..... …..38

2.7. Previous work..... 39

2.8. Discussion 39

Summary 40

Chapter 3: Research Methodology

Introduction..... 42

3.1. Research Framework… 42

3.2. Dataset 45

3.3. Steps and Design 46

Introduction 46

3.3.1. Data Preprocessing.....46

3.3.1.1. MDDR Database Contents… 46

3.3.1.2. Splitting the Dataset 47

3.3.1.3. Reading Data…..47

3.3.2. Graph Based Models… 48

3.3.3 . Generative Adversarial Network Model.....50

3.3.3.1. Generator.....50

3.3.3.2. Discriminator and Reward Network… 51

3.3.4. Model Layers and Functions… 51

3.3.4.1. MLP layers…..51

3.3.4.2. Tanh Function 52

3.3.4.3. Batch Normalization Layer…..52

3.3.4.4. Linear Layer.....52

3.3.4.5. Dense Layer.....52

3.3.5. Model Architecture… 52

3.4. Conclusion 54

Chapter 4: Result and Implementation

Introduction..... 56

4.1. Development Environment and Used Materials 56

4.1.1. Programming Language… 56

4.1.2. Environment 57

4.1.3. Libraries 57

4.2. Proposed Experimental Protocol.....59

4.2.1. Results and Discussion 59

4.2.1.1. Test Parameters..... 59

4.2.1.2. Evaluation measurements 60

4.2.1.3. Training 60

4.2.1.4. Model Results..... 61

4.2.1.5. Results Analysis 62

4.2.1.6. Discussion.....72

4.3. Obstacles.....72

4.4. Conclusion.....73

General Conclusion.....75

References

List of Figures

Figure1.1. The Application of GNN In Drug Discovery..... **04**

Figure2.1. Chemoinformatics Range of Applications in Real Life..... **11**

Figure2.2. Different Types of Molecular Representations **13**

Figure2.3. Example of a Connection Table and End Line Within an MDL Molfile V2000 for the Molecule Leucine Generated by ChemDraw.....**15**

Figure2.4. Mapping a Molecule to a Bit FingerPrint by Taking Different Molecular Substructures **17**

Figure2.5. The Tanimoto Similarity Measures **21**

Figure2.6. Butyrolactone scaffold, Molecular structure and Biological Activity of Compounds Containing **23**

Figure2.7. Structure of a Simple RNN..... **25**

Figure2.8. Simple Graph Neural Network **25**

Figure2.9. Smiles-Based Generative Model using Scaffold Model..... **26**

Figure2.10. Artificial Intelligence Circle.....**26**

Figure2.11. Represents the Relation between AI and Machine Learning..... **27**

Figure2.12. Represents the Relation between AI, ML and Deep Learning..... **27**

Figure2.13. A simple Neural Networks..... **28**

Figure2.14. A simple Architecture of CNN **30**

Figure2.15. RNN Architecture **31**

Figure2.16. RNN loop **31**

Figure2.17. Simplified Illustration of the Architecture AutoEncoder.....**32**

Figure2.18. Variational AutoEncoder Architecture with Molecule Input and Output **33**

Figure2.19. Directed Graph..... **34**

Figure2.20. A social Media Graph **34**

Figure2.21. Applications of GNN **36**

Figure3.1. The General Research Operational Framework.....**44**

Figure3.2. Loading and Creating data frame in Pandas Python **47**

Figure3.3. Displaying Loaded Dataset **48**

Figure3.4. Extracting SMILES.....**48**

Figure3.5. Python code to convert molecules into the graph representation required by the Data object provided by PyTorch-Geometric **49**

Figure3.6. Schema of GAN.....**51**

List of Tables

Table2.1. Various Representations of the Dicycloverine Hydrochloride Molecule **14**

Table2.2. Common Choice of Per-Atom Parameters **16**

Table2.3. Common Similarity/Distance Measure in Chemical Informatics..... **20**

Table2.4. Applications of Graph Neural Networks.....**36**

Table3.1. MDDR Activity Classes for DS1 Data Set..... **45**

Table3.2. MDDR Activity Classes for DS2 Data Set..... **45**

Table3.3. Represent Activity Classes with Number of Molecules.....**47**

Table4.1. Test Parameters **59**

Table4.2. Rusult of the Model For 10 Iterations **61**

Table4.3. Calculation of The Uniqueness Novelty Values **61**

Table 4.4: Presents the samples of activity 1 from the MDDR dataset **62**

Table 4.5: Presents the generated samples from GAN model in iteration 1.....**63**

Table 4.6: Presents the samples of activity 2 from the MDDR dataset.....**64**

Table 4.7: Presents the generated samples from GAN model in iteration 2.....**65**

Table 4.8: Presents the samples of activity 5 from the MDDR dataset.....**66**

Table 4.9: Presents the generated samples from GAN model in iteration 6.....**67**

Table 4.10: Presents the samples of activity 6 from the MDDR dataset.....**68**

Table 4.11: Presents the generated samples from GAN model in iteration 9. ……**69**

Table 4.12: The unique molecules ……....**70**

List of Abbreviations

2D: **T**wo **D**imensional.

3D: **T**hree **D**imensional.

AE: **A**uto**E**ncoder.

AI: **A**rtificial **I**ntelligence.

BP: **B**ack **P**ropagation

BN: **B**atch **N**ormalisation

CNN: **C**onvolutional **N**eural **N**etworks.

CT: **C**hemical **T**able.

DL: **D**eep **L**earning.

ECFP: **E**xtended-**C**onnectivity **F**ingerprints.

GAN: **G**enerative **A**dversarial **N**etwork.

GCN: **G**raph **C**onvolutional **N**etworks.

GNN: **G**raph **N**eural **N**etwork.

GGNN: **G**raph **G**enerative **N**eural **N**etwork.

MDDR: **M**DL **D**rug **D**atabase **R**eport.

MDL: **M**olecular **D**esign **L**imited.

ML: **M**achine **L**earning.

MLP: **M**ulti-**L**ayered **P**erceptron.

MSA: **M**olecular **S**imilarity **A**nalysis.

NN: **N**eural **N**etwork.

RL: **R**einforcement **L**earning.

RNN: **R**ecurrent **N**eural **N**etworks.

SDF: **S**tructure **D**ata **F**ormat.

SLN: **S**YBYL **L**ine **N**otation.

SMILES: **S**implified **M**olecular **I**nput **L**ine **E**ntry **S**ystem.

QSAR: **Q**uantitative **S**tructure-**A**ctivity **R**elationship.

VAE: **V**ariational **A**utoEncoder.

VS: **V**irtual **S**creening.

List of Terminologies

Chemical Database: Chemical database is a database specifically designed to store chemical compound attributed data.

Molecule: The simplest structural unit of a substance that retains the properties of the substance, and is composed of one or more atoms.

Molecular Descriptors: Vector of number based on predefined attributes. It is used to represent a chemical structure.

Similarity Coefficient: Measure used to quantify the degree of structural resemblance between target structure and each structure in the database.

Generative Models: consist of a generator, which learns to generate a molecule from Gaussian noise, and a discriminator, which learns to identify molecules as either real (belonging to a training dataset) or fake (constructed by the generator).

Generative Adversarial Network: A generative adversarial network, or GAN, is a deep neural network which is able to learn from a set of training data and generate new data with the same characteristics as the training data.

Atom: An atom is the smallest unit of ordinary matter that forms a chemical element. Every solid, liquid, gas, and plasma is composed of neutral or ionized atoms. Atoms are extremely small.

De novo design : is a process where scientist try to automatically develop new chemical compounds in the field of drug discovery.

General Introduction

General Introduction

Drug Design involves an iterative refinement process in which researchers perform modifications starting from an initial molecule and trying to optimize a property of interest which can be solubility, toxicity or more often the ability to generate novel molecules. The process traditionally requires long times and high costs in order to synthesize different molecules that are supposedly active in order to be tested experimentally with the ultimate goal of finding a molecule with the desired property. A mathematical model which is able to predict correctly the property of a molecule without the need of experimental testing, would then significantly lower the overall cost of developing a new drug and is of incredible interest in Medicinal Chemistry. These models are generally called Quantitative Structure-Activity Relationships (QSAR) which refers to any regression or classification model able to predict a physicochemical property from a molecular representation.

In practice, predictive QSAR models are often difficult to build and the literature is full of different attempts to represent properly a given molecule and build a mode out of common machine-learning techniques such as linear/logistic regression or random forests. The molecular parameterization constitutes a challenge as most of the currently used representations such as general molecular descriptors or fingerprints reduce a molecule to a set of numbers that describe many of its aspects but do not describe it fully. Different representations may then be suitable for different types of problems depending on which aspects of the molecule we are most interested in, but one representation that is especially suitable for molecules is by considering them as graphs.

The type of machine learning algorithms used for QSAR models, imposes a limit to the shape of the input as a vector of real values, but a different class of architectures that could take a full graph as its input, would be ideal for our purposes. For this reason, Graph Neural Networks got special attention in the context of developing QSAR models as they allow to use what is the most natural representation of molecules.

Graph Convolutional Networks (GCNs) are a common type of GNNs which implement a convolutional operation that helps to keep the description of the molecule in general.

Another limit of most traditional QSAR models is the inability to be directly used to generate maps to be overlapped to molecules in order to know which parts contribute the most to the predicted property and use this knowledge to design new ones. Among the QSAR methods reported so far, "Comparative Molecular Field Analysis" (CoMFA) is the first published one used to build 3D-QSAR models and constitutes an exception which has been considered through the investigation over the years. This, would constitute an important information to proceed with the development of a drug and it is also the kind of knowledge that input-attribution methods attempt to compute. Input-attributions is an umbrella term in the field of Explainable AI (XAI) which refers to any method capable of computing a score for each element of the input. This limit is a consequence of the compressed representation as, once we have trained a model and computed attributions, the scores could not be associated to the molecular structure (which is ultimately a graph). GCNs have again an advantage because of the input representation they use which allows us to directly compute an attribution for each node of the graph, i.e. for each atom. A proper attribution method must still be selected in order to get a useful result.

The main focus of this work is on how to use all available information —a complete molecular representation, an architecture that can leverage that representation and the input-attribution scores computed over the resulting model —in a semiautomatic workflow with the goal of developing tools that can aid researchers in their endeavours, specifically by generating suggestions for adding components to a lead compound and improve the property or activity of interest.

Chapter 1: Introduction

Chapter 1 Introduction

Introduction

This chapter is an entry to a whole new field in chemoinformatics where researches are taking time and effort to get results in the domain of Drug design.

Often in drug discovery, the objective molecular profile is to provoke a desirable biological response, while maintaining properties.

Recently, de novo design has also been referred to as generative chemistry, arising from the increased popularity of generative models in DL.

De novo design has a rich history in chemoinformatics and has received recent attention as ML methodologies continue to open new possibilities for navigating and sampling large search spaces. In this thesis we try to build a new model based on a Graph Neural Network to generate novel valid molecules.

This chapter will clarify the problem of generating novel molecules and show deeper information about the struggles and goals step by step.

Problem Background

The discovery and development of new drugs is a long, costly, and risky process that now takes 10 to 15 years with an average cost of >US$2.5 billion . Within this process, one-third of the time and money are spent during the early drug discovery stages **[1]**.

Therefore, it is attractive for the pharmaceutical industry to develop automated and effective techniques to quickly discover diverse and plausible candidate molecules in vast and discrete chemical spaces. This de novo drug design method can be generally divided into three tasks: (i) molecule generation; (ii) molecule scoring; and (iii) molecule optimization. **[2]**

Furthermore, how to synthesize the designed molecule is also a crucial issue that must be taken into account during de novo drug design. Therefore, we also incorporate synthesis planning in the workflow of automated de novo drug design in this review. Early efforts have been devoted to produce and evaluate new molecules, such as atom-based elongation, **[3]** fragment-based combination, **[4]** and molecular docking. **[5]**

In recent years, with the accumulation of biomedical data and progress in computing power, AI, an emerging computer technology, has developed rapidly in the field of drug design and shows tremendous potential for practical application. **[6,7]**

The discovery of promising drug-like compounds using AI was ranked as one of the top-ten technological breakthroughs of 2020 by MIT Technology Review. **[8]**

AI refers to a computer system that can learn from experience, adjust to new inputs, and perform human-like tasks. AI systems powered by machine learning (ML) can leverage massive amounts of data to uncover new insights and patterns, without dependency on any theoretical or practical knowledge discovered by humans. **[9]**

Thus, it is suitable to deal with problems encountered during drug design, where there is rich data available but less well-defined rules. In particular, deep learning, as a branch of ML, shows exceptionally fascinating prospects in drug design because of its powerful capability to learn complicated relationships between features and outcomes from large-scale data. **[10]**

Among various deep-learning models, GNNs **[11]** have attracted increasing attention from the field of drug molecules design, because the resulting graph is the most intuitive and concise way to represent molecules. **[12]**

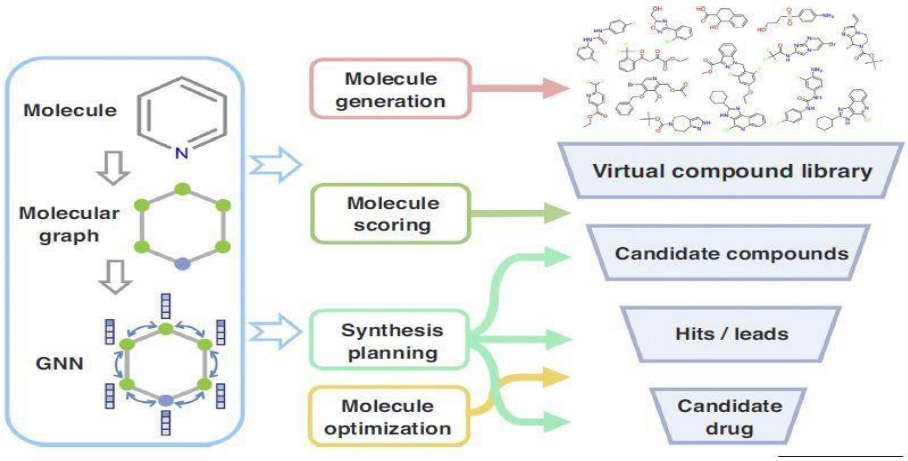


Figure.1.1. The Application of GNN In Drug Discovery

Chemical graph theory describes a molecule as an undirected graph in which nodes and edges correspond to atoms and chemical bonds, respectively. The neural networks operating on such molecular graphs can all be regarded as a kind of generalized GNN. According to different mission objectives, those GNNs for molecule learning can be further summarized into four categories, which have different roles in de novo drug design. The first category aims to predict the labels of the whole graphs, which can be used in the prediction of various molecular properties. The second category aims to predict the labels of nodes or edges, which can be used in the prediction of reaction and retro synthesis. The third category aims to learning the implicit representation of the graphs, which can be used in the one-step generation and optimization of molecules, whereas the fourth category aims to learn the transforming rules of the graph, which can be used in the iterative generation and optimization of molecules as well as the prediction of reaction and retro synthesis. There have been many excellent reviews on the application of AI technology in drug discovery. **[13]**

Here, we specifically focus on the application of GNNs in de novo drug design and briefly introduce the recent progress and outlook of this field.

Chapter 1 Introduction

1.1. Problem Statement

De novo molecular design is the process of automatically proposing novel chemical structures that optimally satisfy a desired molecular profile.

Often in drug discovery, the objective molecular profile is to provoke a desirable biological response, while maintaining acceptable pharmacokinetic properties. Recently, de novo design has also been referred to as generative chemistry, arising from the increased popularity of generative models in AI.

Traditionally, virtual screening (VS) is undertaken to identify molecules likely to exhibit desirable experimental outcomes. A key difference, compared with de novo design, is the source of the molecules considered: where structures are known a priori in VS, in de novo design we seek to generate the structures to be evaluated.

Chemical space, the expanse spanning all possible molecules, is vast. Although enumerated VS libraries are becoming enormous by drug discovery standards, with many containing upward of a billion molecules, these libraries correspond to a miniscule proportion of chemical space. Additionally, when considering compound libraries of such size, evaluation methods might necessarily sacrifice predictive validity. **[14]**

By generating compounds in a directed manner using de novo design, computational practitioners hope to traverse chemical space more effectively, reaching optimal chemical solutions while considering fewer molecules than allowed by brute-force screening of large chemical libraries.

Furthermore, there are likely many acceptable regions of chemical space for a given objective and, to this end, molecular design methods are tasked with balancing the exploration of global solutions with the exploitation of local minima.

De novo design has a rich history in chemoinformatics and has received recent attention as ML methodologies continue to open new possibilities for navigating and sampling large search spaces. **[15, 16]**

In this study, we consider de novo design methodologies from the perspective of the coarseness of molecular representation; specifically, we distinguish graph-based approaches for generating novel structures.

1.2. The Research Question

The main issue in this research is:

How we can generate novo molecular with a graph representation using graph neuron network based on activity structure?

Thus, the following issues will need to be addressed in order to answer the main research question stated above:

1- What is the drug discovery and where is deep learning there?

2- What are representation specific–methods (How to generate a graph?)

3- What are approaches to generating useful molecules? Is the graph neuron network (GNN) model gets a useful result?

4- Is it possible to founding a molecule with the desired activity? Or are the predictive molecules active?

5- Is it possible to have the same structure and activity?

1.3. Research Objectives

The main focus of this work is on how to use all available information to develop a novel molecule, using graph neuron network we generate novo molecule based on activity structure with the goal of developing tools that can aid researchers in their endeavors, specifically by generating a valid molecule suggestions for founding drugs and improve the property or activity of interest.

To achieve this goal, the following objectives have been set:

1- To develop a GNN model using graph representation to create novo molecular design

2- To apply GNN based on activity structure

3- To investigate the performance of using GNN in molecular generation

4- Compare the predicted molecules with the input molecules if they have the desired activity and the same class of activity

1.4. Scope of the Study

The scope of this study is confined to the following in order to fulfill the above-mentioned objectives: This study will focus on

The validation and evaluation of the quality of the prediction model proposed in this research will be done on different datasets that have been used to validate the generation of molecules based on structure-activity relationship.

The databases aimed to be used in this study are only limited to chemical data from MDDR, and among various deep-learning models we use GNN to generate novo molecules, because the resulting graph is the most intuitive and concise way to represent molecules.

1.5. Thesis Outline

This report is mainly divided into four chapters.

Chapter One provides an introduction and brief overview of the project including problem statement, objective of the research, scope of the study, thesis outline, and summary.

Chapter Two , literature reviews , begins with an overview of chemoinformatics , then we present various type of molecular representations, Molecular similarity measurement , molecular biological activity and Molecular

Generation, In the second section we talk about Deep learning and its background, and explain some of the relevant DL architectures like Convolutional neural network. This includes the background knowledge on the terms that are involved in the project mainly like Autoencoder , variational AutoEncoder, also describe in details Graph Neural Networks and its implementation ,In addition, we also present variational graph autoEncoder and Challenges , This chapter is concluded by a discussion and summary about the applicability of the mentioned methods of deep learning ,there architectures and the best ways to improve the performance of these methods based on activities structures .

Chapter Three describes the overall methodology adopted in this research to achieve the objectives of this work. Here, we try to give a general picture about each phase in our research framework. The databases aimed to be used in this study which is chemical data from MDDR, and explaining the functionality of the model and the best parameters, including some detailed explanation of the important functions of the model and a summary in the end.

Chapter Four presents the implementation details, and presents the results from applying Graph Adversarial network to generate novel molecules based on activities structure. Finely a conclusion and perspectives are proposed.

1.6. Summary

In this chapter, we try to give a broad overview of the problems involved in the molecular generation and applied GNN to generate molecules based on structure-activity relationship which is the subject of this study.

This chapter serves as an introductory to the research problem addressed in this study.

The goal, objectives, the scope, and the outline of this study were presented.

Chapter 2 : Literature Review

Introduction

Artificial intelligence (AI) (named by John McCarthy) first appeared in the article of Turing, 1950, who asked the question "Can machines think? **[17]** The goal of AI is to design systems capable of reproducing the behavior of humans in their reasoning activities in other words, "Systems that think like humans" and go as far as beating him in challenges (Turning Test) that according to him a computer must possess the learning functionality to be able to improve autonomously, and as the boss of artificial intelligence in Facebook Yan leCun also stated: "There is no intelligence without learning", which is where the machine learning (ML) approach comes from. Machine learning is the discipline that aims to make a machine capable of learning by itself without being explicitly programmed. A subset of ML called deep learning (DL) aims at handling and solving more complex tasks such as pattern recognition, signal processing, memorization, generalization with deeper and more accurate learning, based on the biological neural mechanism translated into artificial neural networks (multilayer). **[18]**

Deep generative models have become a widely used tool to generate new data from limited amounts. They have been applied successfully to generate text, images, video and music. Additionally, they have been applied to drug discovery and have enabled many new ways to explore the chemical space.

For instance, Graph Neural Networks (GNNs) are able to generate billions of drug-like molecules, and one specific feature of a large number of architectures is the use of the SMILES molecular representation, The use of the full molecular graph has been reported in literature **[19]**. Up to now SMILES generative model architectures do not allow molecular generation from scaffolds mainly due to the restrictions of the SMILES syntax. Some approaches have been reported that use graph generative neural networks (GGNN) which are even able to decorate a scaffold without the need of specifying attachment points.

Making (GGNN) the hot topic of this study.

2.1. Chemoinformatics

Chemoinformatics is a relatively new principle of chemistry and is based upon the processing of data concerning chemical and molecular structures through the use of computational analysis. **[20]**

The analysis of these data allows the relationship between chemical structure, chemical properties, and molecular activity to be studied. it is an in silico technique, which means it is a form of scientific study which is performed virtually on a computer via software and simulations, chemoinformatics has emerged from several older disciplines such as computational chemistry, computer chemistry, chemometrics, QSAR, chemical information.It poses very challenging problems such as converting 3D molecule structures into inputs to the ML model, tackling the scarcity of data, or trying to get a better understanding of which molecule's features are important for predictions. **[21]**

Chemists were already aware of the potential of computer applications in advanced molecular problems however also realized that such problems as drug discovery (molecular) design will not proceed as smooth as previously expected.

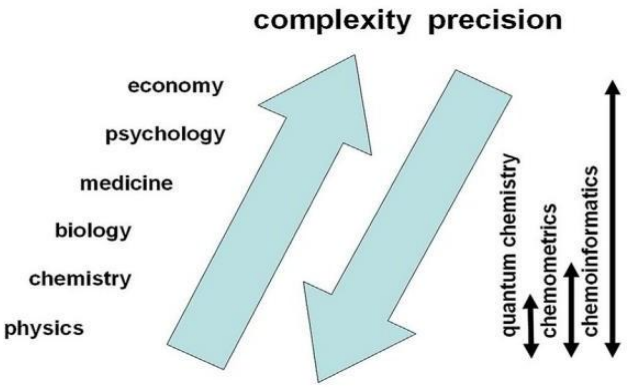


Figure2.1. The extent of Computer-Assisted Chemical Investigations in Association with other Disciplines is Shown Above**. [22]**

Finally a novel medication is not only biophysics, but also the surprising behavior of the commercial market. As a result, chemoinformatics has a wide range of applications, ranging from physics to economics. The difficulty of problems is determined by their broad breadth. **[22]**

2.2. Molecular Representations

Representing chemical data in a concise and unambiguous way that both humans and machines can comprehend, is not an easy task; this is particularly true for the description of molecules. While there are numerous methods of adequately representing small and "simple" organic molecules, significant complexity may arise when considering molecules with features such as ring structures, nonstandard valency /bonding, inorganic components, or symmetry. These complexities may lead to issues such as representations being noncanonical (i.e. many representations of the same molecule), being nonunique /clashing (i.e., multiple different molecules that are encoded into the same representation), assuming the wrong number of implicit hydrogen atoms, or failing to capture tautomerism. This can make (sub) structure searching in databases difficult, and even result in representations that refer to the wrong molecules. One way of elucidating the robustness of a representation is with a so-called "round-trip conversion experiment," which tracks whether the conversion from representation to structure and back is correct for a given molecule. As an example, one could draw a molecule in ChemDraw, read it into ChemDoodle, and then check whether the same structure is obtained when reading the ChemDoodle file back into ChemDraw. Broadly speaking, molecules can be represented in a machine-readable format in four ways: as a string; with a connection table; as a collection of features, for example, a fingerprint or series of physical descriptors; or most recently, with a computer-learned representation using machine learning (ML) and deep learning (DL).

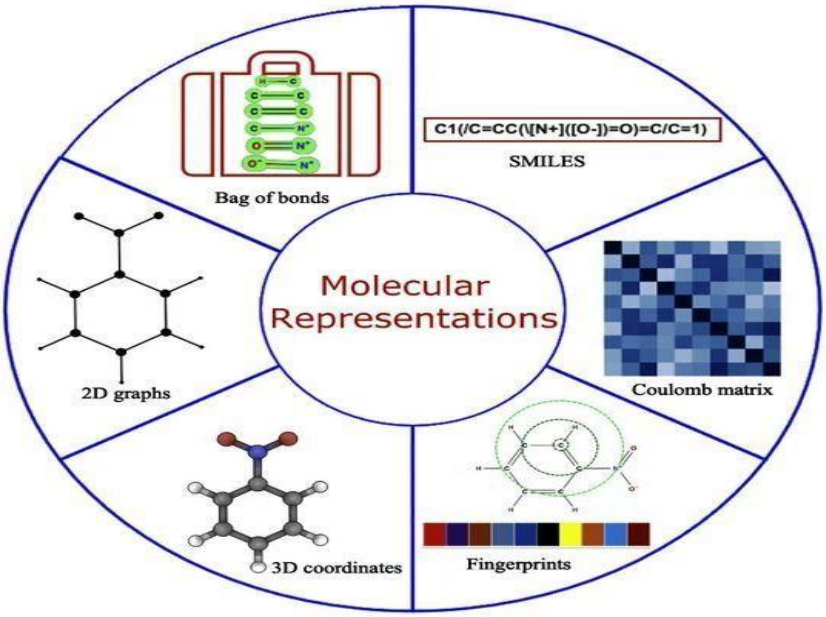


Figure2.2. Different Types of Molecular Representations**.**

2.2.1. String representations

String representations generally consist of characters from the American Standard Code for Information Interchange (ASCII) character encoding standard, and are more compact and easier for humans to read and write than other representations. The most popular string representations are:

- Registry systems.
- Wiswesser Line Notation.
- Fragmentation codes.
- IUPAC.
- Simplified molecular.
- International Chemical Identifier.

  Also, exist less popular line notations than the aforementioned, such as SYBYL Line Notation. **[23]**

An example showing various string representations for the molecule dicycloverine hydrochloride can be seen in **Table 2.1**.

Generic names	Dicycloverine HCl, benacol, bentyl, dibent, Dyspas
Mol. Formula	C19H36ClNO2C19H36ClNO2
IUPAC name	2-(Diethylamino)ethyl 1-cyclohexylcyclohexane-1-carboxylate hydrochloride
CAS RN	67−92−567−92−5
Canonical SMILES	CCN(CC)CCOC(═O)C1(CCCCC1)C2CCCCC2.Cl
InChI	InChI = 1S/C19H35NO2.ClH/c1-3-20(4-2)15-16-22-18(21)19 (13-9-6-10-14-19)17-11-7-5-8-12-17;/h17H,3-16H2,1-2H3;1H InChIKey:GUBNMFJOJGDCEL-UHFFFAOYSA-N
WLN	L6TJA-AL6TJAVO2N2&2&GH

Table 2.1.Various Representations of the Dicycloverine Hydrochloride Molecule.

2.2.2. Chemical Table Representations

A chemical table (CT) lists the (*x*, *y*, *z*) coordinates of each atom in a connection table (CTab), and how they are bonded to each other in a molecule. This makes the generation of 2D/3D graphic representations from a CT quite easy, and they are typically used for representing molecules within databases/programs. The most widely used is the MDL molfile, which exists in two versions (V2000 and V3000). MDL molfiles can be "bundled" into a structure–data file. One drawback of the CT is that translation or rotation of a molecule will lead to a new set of atom coordinates, despite the molecule being unchanged.

2.2.2.1. MDL Molfile

MDL Molfiles consist of three main sections:

- **A header block:** Containing title, Timestamp and an optional comment,
- **A CTab:** The CTab consists of a number of sections.

- **M END:** which define the end line of the molecule.

Best understood by considering **Figure 2.3** down bellow:

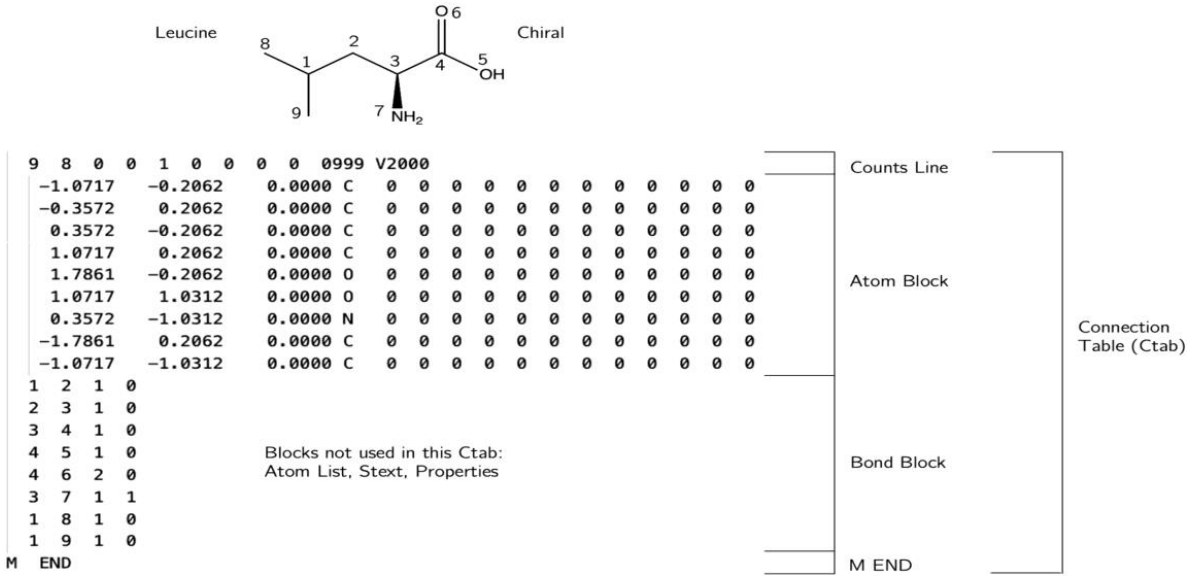


Figure2.3. Example of a Connection Table and End Line Within an MDL Molfile

One of the primary drawbacks of CTs is related to their handling of complex chemistries. MDL Molfiles only support single, double, and triple bonds, which does not work well for molecules containing bonds where simple sharing of electrons in a covalent bond is an inadequate description. It has been suggested that introducing a zero-bond order could (Partially) alleviate this issue.

Furthermore, when the number of hydrogen atoms is not explicitly stated, nontrivial valency could lead to the wrong number of implied hydrogen atoms; an important issue not easily solved (simply requiring the number of H-atoms to be specified might compromise back-compatibility).

2.2.2.2. Graph Representation

The most intuitive, complete representation for molecules is a graph in which nodes describe atoms and edges describe bonds.

Every node/atom will have its own properties and it will be connected by edges/bonds of possibly different types (single-, double-, triple- bonds).

Many choices could be made but the simplest representation would be to consider no special property for bonds and decide a set of atomic properties for each node.

There is no default choice for what properties to consider but a common one is provided.

Feature	Description	Type
Atom type	atom type	one-hot
Degree	the number of heavy atom neighbors (0 to 6)	one-hot
Number of hydrogens	the number of neighboring hydrogens (0 to 4)	one-hot
Implicit valence	the number of implicit hydrogens (0 to 6)	one-hot
Hybridization	sp, sp^2, sp^3, sp^3d, or sp^3d^2.	one-hot
Formal charge	atomic formal charge (-3 to +3)	one-hot
Ring size	whether this atom belongs to a ring (ring size: 3 to 8)	binary
Aromaticity	whether this atom is part of an aromatic system.	binary
Chirality	*R*, *S*, or nonchiral	one-hot
Acid/base	whether this atom is acidic or basic	binary
Hydrogen bonding	whether this atom is a hydrogen bond donor or acceptor	binary

Table2.2. Common Choice of Per-Atom Parameters.

One of the first attempts to use a representation different from fingerprints in favor of graphs, is represented by the work of Kearnes where “Molecular Graph Convolutions” are presented. The paper develops a custom architecture to work on molecules by considering them as graphs with atoms having their own properties, trying to including the 3D information of the molecule as well.

They conclude that their work cannot outperform state-of-the-art fingerprints molecules on all tasks but their early work shows the potential to build architectures with more parameters that can learn more specific functions. This is different from Graph Convolutional Networks which became more common in later work.

2.2.2.3. Extended-Connectivity FingerPrints

Molecular fingerprints are intended to represent the presence (or absence) of substructures within molecules often in a sparse vector, and they generally fall into one of two categories: matching substructures in a molecule to substructures in an expert-defined set, or algorithmic enumeration and hashing of substructures in a molecule.

The ECFP is one of the most widely used chemical fingerprinting techniques due in large part to the popular open-source python package RDK it having an implementation of it, called “Morgan fingerprint.”

However, chemical fingerprinting techniques existed before the ECFP; see for example HOSE and FREL. Comparing and contrasting fingerprinting methods can be found in the literature.

Morgan fingerprints are lightweight, quick to compute, and represent salient features of molecules well, and have thus found many uses beyond computing similarity. Being already formatted as a vector with numerical entries, they are well suited to be used as features in ML models for chemistry.

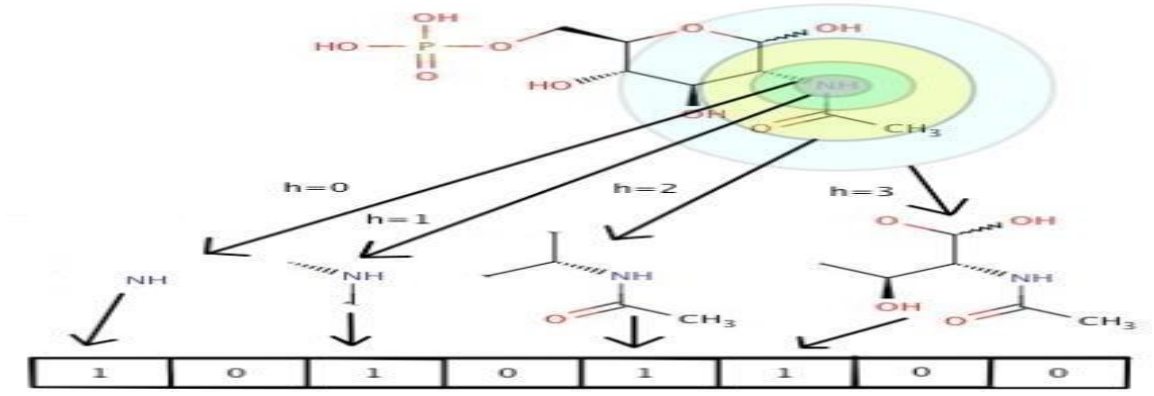


Figure2.4. Mapping a Molecule to a Bit Fingerprint by Taking Different Molecular Substructures**.**

which are defined by a varying radius range (h=0, h=1, …) from the NH atom pair.

This molecule is represented by a vector of length 20 consisting of binary numbers.

2.2.2.4. Molecule Generation in 3D

Real molecules exist in three dimensions, so a representation in fewer dimensions must necessarily incur a loss of information, which may or may not be relevant to the task at hand. Molecules with multiple low-energy conformations also cannot be adequately represented simply with a single static 2D/3D representation.

A molecule represented in 3D space would have different coordinates following translation and/or rotation, despite still being the same molecule, and answering even the simple question of identicality of two molecules can be computationally expensive.

Tensor field NNs, which are locally equivariant to rotations, translations, and permutations in 3D, have recently been introduced, and they have been shown to be capable of handling molecular structures (when molecules are treated as 3D point-clouds).

There are also a number of examples of 3D molecular representations being developed to be compatible with CNNs. Deep learning has been shown to make accurate predictions about biological function from electron density fields and electrostatic potential fields. Representing molecules in 3D adds additional degrees of freedom, and while this may be a good thing, because it allows the representation to more closely align with the real world, it also would require more data to yield a well-trained model. We believe access to high-quality standardized data is one of the most significant bottle-necks in computational chemistry and drug discovery today, and while this issue is indeed receiving much attention (e.g., with the Open Reaction Database initiative), we are cautious about methods which would require more data to work well, rather than less. **[24]**

2.3. Molecular similarity measurement

2.3.1. Introduction of molecular similarity measurement

Molecular similarity is a pervasive concept in chemistry. It is essential to many aspects of chemical reasoning and analysis and is perhaps the fundamental assumption underlying medical chemistry.

Therefore How molecular information is represented, called the representation problem, and it is important to the type of molecular similarity analysis (MSA) that can be carried out in any given situation. In this side, Sets, graphs, vectors, and functions are mathematical structures that are utilized to encode molecular information.

Similarity measures, also called similarity coefficients or indices, are functions that map pairs of compatible molecular representations that are of the same mathematical form into real numbers usually.

Determining the similarity of one "molecular object" to another is basically an exercise in pattern matching – generally called the matching problem.

The outcome of the exercise is a value, the similarity measure that characterizes the degree of matching, association, proximity, resemblance or similarity of pairs of molecules as manifested by their "molecular patterns," which are made up of sets of features.

2.3.2. Similarity and Distance

A similarity measure can be defined as a function n that can compute the degree of similarity between a pair of object, from mathematical point of view the distance also known as dissimilarity is used to quantify the degree of difference between two objects. The smaller value of distance, the larger the degree of similarity and vice versa.

Engineers have proposed four different algorithms of pattern matching to determine the presence or absence of an atom and associated interatomic distance in 3D chemical structure and the results show that, out of four algorithms proposed, the Ullman algorithm is more efficient on small molecular structure searching compared to the others.**[25]**

Nowadays, the similarity searching of 3D molecular structure had rapidly developed with various applications and methods which are: Manhattan distance, Tanimoto coefficient, and cosine coefficient…etc.

2.3.3. Analysis of Similarity Measurement for Molecular Structure

According to Willett peter, a similarity searching will be effective if, and only if, an appropriate similarity measure is employed. **[26]**

Similarity/Distance Coefficient	Formula	
Tanimoto Coefficient	$Tani = \frac{c}{a+b-c}$	(1)
Dice Coefficient	$Dice = \frac{2a}{2a+b+c}$	(2)
Cosine Coefficient	$Cos = \frac{a}{\sqrt{(a+b)(a+c)}}$	(3)
Euclidean Distance	$ED(a,b) = \sqrt{\sum_{i=1}^{n}(b_i - a_i)^2}$	(4)
Hamming Distance	$HM(a,b) = a+b-2c$	(5)

Table.2.3. Common Similarity/Distance Measure in Chemical Informatics.

This coefficient is specifically designed for use with binary data, which used 0 and 1 to indicate the absence and the presence of molecular structure .The association coefficient also can be used with non-binary data, in which the different formulation and range of value may apply.

Most of the previous research done by most of famous universities indicate that the Tanimoto coefficient is the most used one, Tanimoto coefficient was preferred based on two reasons : firstly, on the basis of a subjective evaluation of the similarity search ranking, and secondly, referred to its calculation that does not involve any square root, making it faster in calculation.

Equation (1) refers to the Tanimoto coefficient (also known as the Jaccard coefficient) and it said as the extension from the Jaccard coefficient. The Tanimoto coefficient is defined as the sum of the common feature occurs in both structures (c) over the total feature that occur only in (a) and (b), minus with the common feature, (c). **[25]**

For example, a fingerprint is a binary sequence of Boolean values where 1 indicates a presence of a feature and a 0 indicates absence. Feature examples are “200 < molecular weight <= 250” or “aromatic ring count = 1”. When comparing these fingerprints similarity scores are used. A much-applied method is the Tanimoto similarity (Figure 5).

Which for molecule A and B is defined as SAB= a / (a + b +c) where a bit are set common to both strings, b bits are set in the comparison string, c bit are set in the reference-structure string, and d bits are set in neither string.

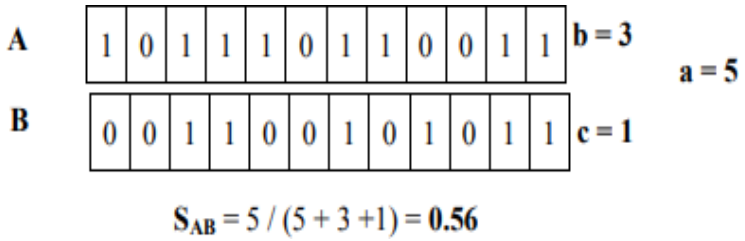


Figure.2.5. The Tanimoto Similarity Measures.

2.3.4. Conclusion

This part reviewed the common similarity/distance measurer in molecular structure, In comparison to other similarity measures, the Tanimoto coefficient is determined to be the most extensively utilized in similarity searching in chemo informatics field. Some researchers believed that the similarity and distance measurer can be used for non-binary data. In conclusion, the studies of similarity searching in chemical informatics offer the advantages to the chemist and also in the field of drug discovery itself.

2.4. Molecular Biological Activity

The present state of knowledge on the structure of a biological molecule gives the latter a typical general structure:

- A chemofunctional part contributing to the bond of the molecule with the corresponding receptor;

- A biofunctional part responsible for the biological activity.

Each molecule endowed with a biological activity takes a specific structure of this action, the determination of which makes it possible to obtain a model, or pharmacophore, which is found in all molecules with similar properties.

The extension to pharmacomolecules of the hypotheses on the complementarity of the enzymatic sites and the substrate (key-lock) made it possible to affirm that certain privileged sites exist at the cellular level, called receptors, having a complementary structure of the pharmacophores and on which the latter come to nest. These receptors are located in the proteins that make up the cell's plasma membrane. It then forms a drug-receptor complex responsible for different biological actions. The intensity of action of a drug is most likely proportional to the number of receptors occupied.

The forces of interaction between the drug and the receptor may, depending on the case, be polar, ionic, or due to hydrogen bonds or Van der Waals forces.

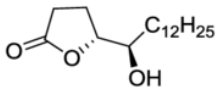
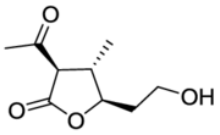
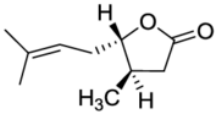
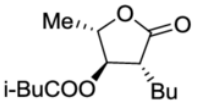
Name	Molecular Structure	Biological activities
(-) Muricatacin		Antitumor[1e]
11-Deoxyprostaglandins		Anti-inflammatory, anti-apoptotic, anti asmthmatic agents
Eldanolide		Phermone
(+) Blastmycinone		Antifungal agents
Pilocarpine		Commonly used in the treatment of glaucoma.[1f]

Figure.2.6.Butyrolactone Scaffold, Molecular Structure and Biological Activity of Compounds Containing.

2.5. Molecular Generation

The ultimate goal of molecular design for new materials and drugs is to directly generate molecules with the desired properties. This is apparently challenging work because a molecular space is extraordinarily vast, discrete, and disorganized with diverse types of molecules. For instance, 10^8 molecules have been synthesized, whereas it is estimated that there are 10^{23}–10^{60} drug-like molecules, Despite advances in experimental techniques, it is too demanding to find molecules suitable for specific applications only through experiments. **[27]**

2.5.1. Molecular Generation Using Computing

Computer-aided molecular design has attracted much attention as a promising solution to overcome the experimental limitation; High-throughput virtual screening to find molecules with target attributes is possible thanks to fast calculation methods, reasonable accuracy, and inexpensive cost. A common strategy is to select computationally top molecules out of millions of molecules in a virtual library and then verify them experimentally.

As a result time and effort are saved significantly. Molecules in the library may not meet the given criteria. In this case, traditional optimization methods such as a genetic algorithm can be used to further improve molecular properties beyond the criteria by structural modifications.

Recently emerging generative models based on deep learning techniques may offer a viable solution for more efficient molecular design. Scientists adopted a variational autoEncoder, to improve molecular properties in a latent space where molecules are represented by a real vector; The key advantage of this method is that a gradient-based optimization becomes feasible because the latent space is continuous and differentiable. **[28]**

2.5.1.1. RNN-based Generative Models

The RNN-based models attempt to learn the underlying probability distribution over a large chemical set. And fine-tuning with maximum likelihood estimation (MLE) would cause a risk of forgetting what was initially learned by the RNN. To reduce this risk, the concept of RL has been integrated to molecular generation.

To reduce this risk, the concept of RL (Reinforcement Learning) has been integrated to molecular generation. RL is usually suitable for solving dynamic decision problems. RL generative method aims to fine-tune a pertained RNN-based generator toward creating desirable molecules through learning an imposed episodic likelihood considering both prior likelihood and a defined reward function. **[29]**

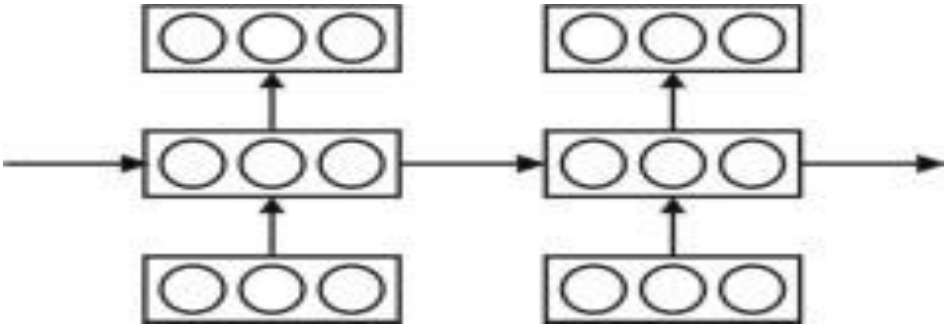


Figure2.7. Structure of the simple RNN.

2.5.1.2. Graph-Based Generative Models

Graphs are commonly thought of as basic structural data. which are widely adopted in the field of biological and chemical network systems. Given a chemical compound (molecule) as an example, a 2D structure is a graph with nodes as its atoms and edges between two nodes as its bonds. There has been a surge of great interest in studying DL models on graphs, with useful applications across fields such as computer vision or chemistry, Fueled by the success of RNNs, DL generative models on molecular graphs have recently allowed the direct



Figure2.8. Simple Graph Neural Network.

2.5.1.3. SMILES-Based Generative Models

The SMILES string, like natural language, is a popular molecular representation method among molecular generative models. For the SMILES-based generative models, one-hot encoding of a string is fed into a model for learning, and then generates a string char-by-char based on the learned distribution. **[30]**

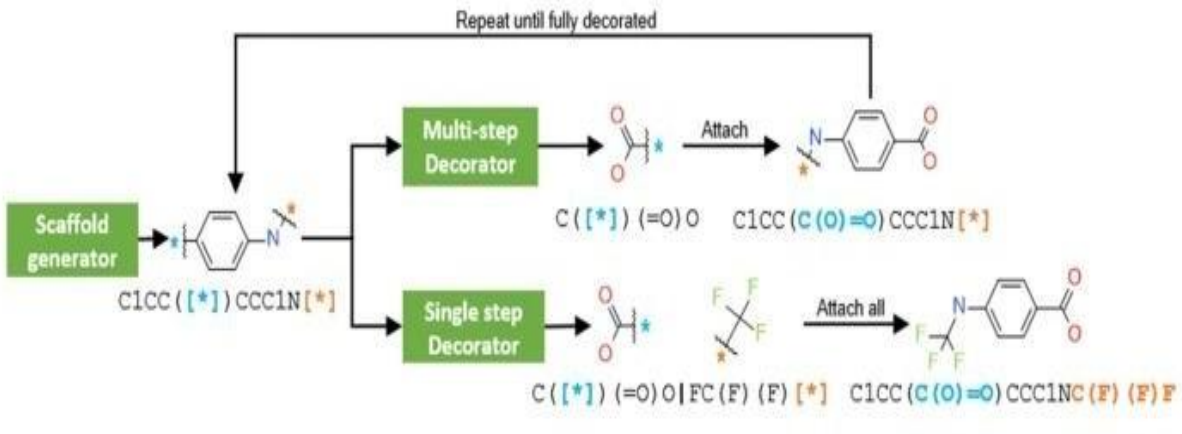


Figure.2.9. Smiles-based Generative Model using Scaffold Model.

2.6. Deep learning

2.6.1. Artificial Intelligence (AI)

AI is the theory and development of computer systems able to perform tasks that usually require human intelligence, including: (Visual perception, speech recognition, decision-making).

Examples: expert systems, machine learning, used in CAD



Figure2.10. Represent the Artificial Intelligence.

2.6.2. Machine Learning (ML)

Machine Learning is a type of artificial intelligence technology that allows machines to learn without having previously been specifically programmed for this purpose. Machine Learning is explicitly linked to Big Data, since to learn and develop, computers need streams of data to analyze, on which to train with minimal human intervention.

As a result, Machine Learning, which is essentially based on Big Data, precisely needs Big Data to function. Machine Learning and Big Data are therefore interdependent. **[26]**

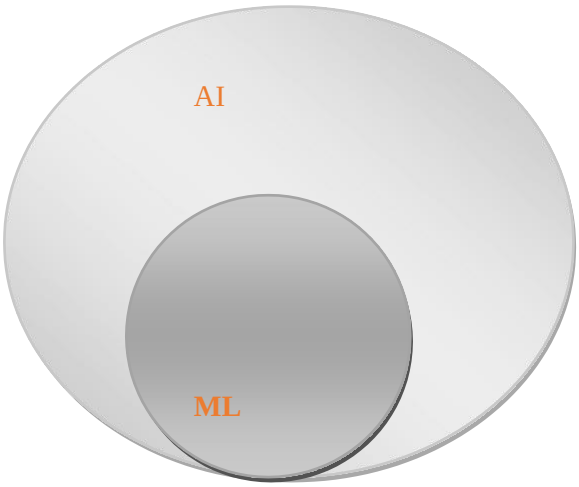


Figure2.11. Represents the Relation Between AI and Machine Learning.

2.6.3. Deep Learning (DL)

Part of the broader family of machine learning, individualized by specific configuration of neural network organized in multiples layers, emulating the human learning approach and increasing the ability to address complex problems. Deep learning networks are iterative methods that propagate information, training their features automatically through gradient-based optimization methods and back propagation. **[31]**

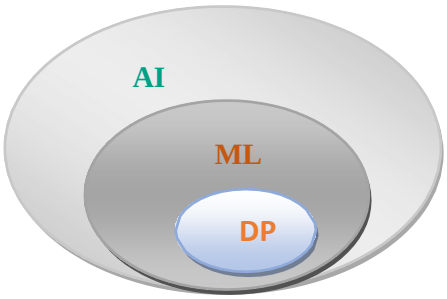


Figure 2.12. Represents The Relation Between AI, ML and Deep Learning.

2.6.4. Convolutional Neural Network (CNN)

2.6.4.1. Neural Networks

A Neural Network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. **[32]**

Neural Networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems. **[33]**

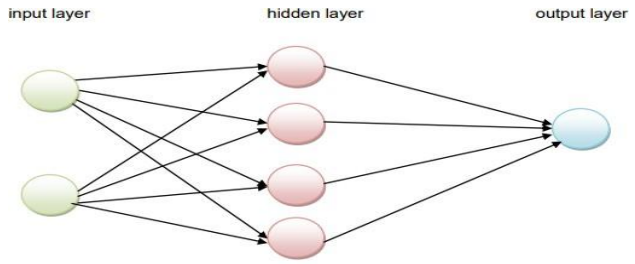


Figure 2.13. A Simple Neural Networks.

-Basics of Neural Networks : Neural networks, in the world of finance, assist in the development of such processes as time-series forecasting, algorithmic trading, securities classification, credit risk modeling, and constructing proprietary indicators and price derivatives.

A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

A neural network contains layers of interconnected nodes. Each node is a known as perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

-**Multi-Layered Perceptron:** In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map. For instance, the patterns may comprise a list of quantities for technical indicators about a security; potential outputs could be “buy,” “hold” or “sell.”

-Hidden layers: fine-tune the input weightings until the neural network’s margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs.

This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis.

-Application of Neural Networks: Neural networks are broadly used, with applications for financial operations, enterprise planning, trading, business analytics, and product maintenance. Neural networks have also gained widespread adoption in business applications such as forecasting and marketing research solutions, fraud detection, and risk assessment.

A neural network evaluates price data and unearths opportunities for making trade decisions based on the data analysis. The networks can distinguish subtle nonlinear interdependencies and patterns other methods of technical analysis cannot. According to research, the accuracy of neural networks in making price predictions for stocks differs. Some models predict the correct stock prices 50 to 60 percent of the time, while others are accurate in 70 percent of all instances. Some have posited that a 10 percent improvement in efficiency is all an investor can ask for from a neural network. **[34]**

-The Components of a Neural Network: There are three main components: an input later, a processing layer, and an output layer. The inputs may be weighted based on various criteria. Within the processing layer, which is hidden from view, there are nodes and connections between these nodes, meant to be analogous to the neurons and synapses in an animal brain.

2.6.4.2. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of neural network used to recognize patterns in images or extract features from text. A CNN has an input and output layer, as well as numerous hidden layers. Convolutional, pooling, and completely linked layers are among the hidden layers. Simard has details on the numerous layers of CNN.

The introduction of convolution layers is an essential element of a CNN. Each input data is subjected to the identical convolution procedure in each layer, after which it is replaced with a linear combination of its neighbors. Filter or kernel refers to the linear combination's parameters, The number of neighbors considered is referred to as filter size or kernel size. A feature map is the result of applying a single filter to the input data. Nonlinearity, such as sigmoid or scaled exponential linear units (SELU), can be used to model nonlinear data on a feature map. In addition, layering a CNN on top of a feature map will allow. **[31]**

One to simulate aspects of inputs that are spatially distant the pooling layer is used to make feature maps smaller. The feature maps are concatenated into fully linked layers after going through many convolution and pooling layers, where every neuron in neighboring layers is connected to generate the final output value.

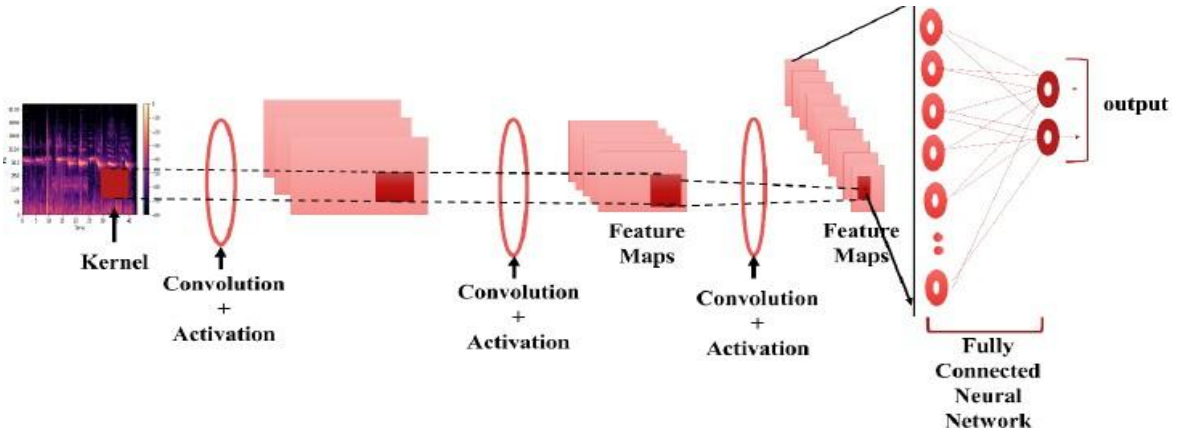


Figure2.14. A Simple Architecture of CNN. **[32]**

2.6.5. Recurrent Neural Network

Humans don't build their thinking from scratch. They don't throw it all away and start thinking from scratch again. The thoughts have persistence. Traditional neural networks can't do this, and that seems like a major flaw. For example, imagine you want to categorize the type of event that occurs at each stage of a movie. It's not obvious that a traditional neural network can use its reasoning about events from earlier in the movie to exploit them later.

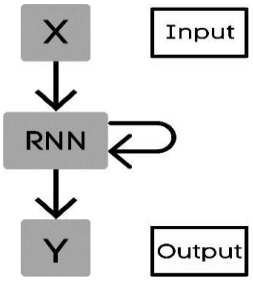


Figure2.15. RNN Architecture

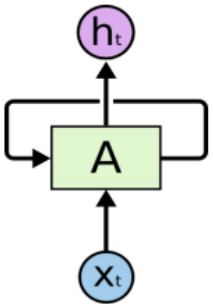


Figure2.16.RNN loops

Recurrent Neural Networks can solve this problem. They are networks with loops that allow information to persist. In the figure above, a neural network segment; "A" looks at an input Xt and provides a value ht. A loop allows information to be passed from one stage of the network to the next.

A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. This chain nature reveals that recurrent neural networks are intimately linked to sequences and lists.

They are the natural architecture of the neural network to use for such data. In recent years, there has been an increased success in applying RNNs to a variety of problems: speech recognition, language modeling, translation, image subtitling, etc. **[17]**

2.6.6. Auto-Encoder

An AutoEncoder is an unsupervised learning neural network. It is primarily used for learning data compression and inherently learns an identity function .First introduced in the 1980s. An AutoEncoder network aims to learn a generalized latent representation (encoding) of a dataset. AutoEncoder is helpful in various domains, such as for processing image, text, and audio.

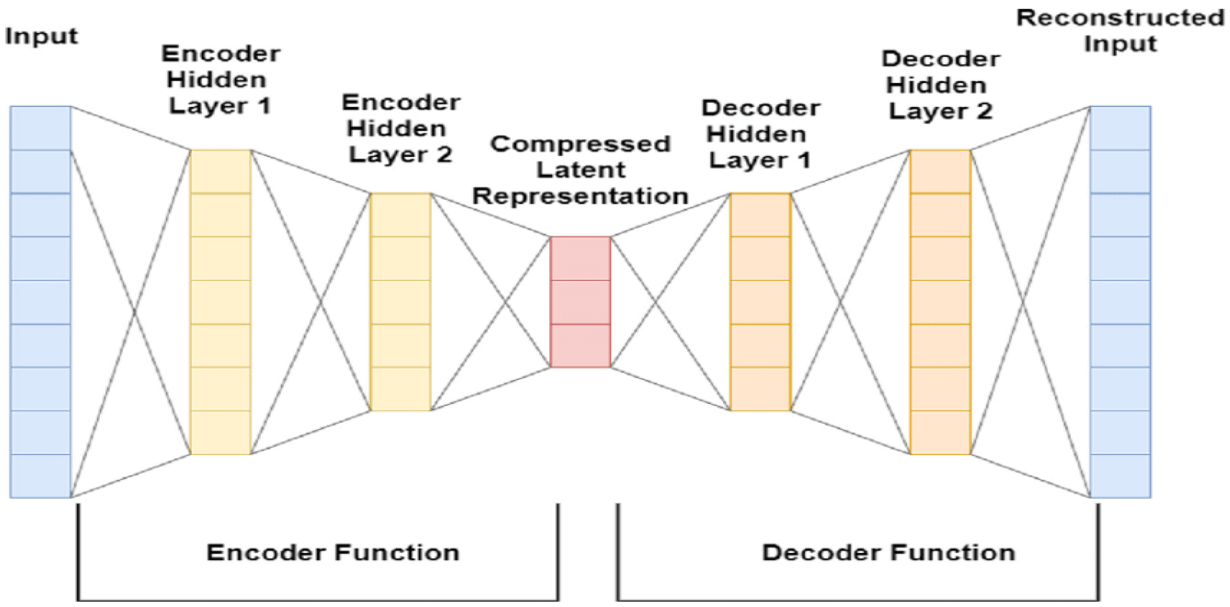


Figure 2.17. Simplified Illustration of The Architecture AutoEncoder.

AutoEncoders based on neural networks with special architecture can also be used to process complex discrete structures with variable number of elements, such as graphs and sequences of characters. In this case, the formation of vectors in the code layer can be considered as an embedding of such structures into the Euclidean space of fixed dimensionality. In chemoinformatics, the latent vectors d are viewed in this case as "revertible" data-driven molecular descriptors, which form a continuous representation of discrete chemical objects, e.g. molecules and chemical reactions. **[33]**

The AutoEncoder network has two blocks:

1. **Encoder**: This is the part of the network that compresses the input into a fewer number of bits known as **latent-space**, also sometimes called a bottleneck. This latent-space representation is called an "encoding" of the input
2. **Decoder**: It is the part of the network that reconstructs the input image from the compressed representation**. [35]**

2.6.6.1. Variational AutoEncoder

The Variational AutoEncoder (VAE): can be seen as a probabilistic version of an AE, where the output of the decoder is not directly a value of x but the parameters of a probability distribution of x. As we will see below, the same probabilistic formulation applies to the encoding of z. The resulting probabilistic model can be used to:

• Generate new data (from unseen values of z).

• Transform existing data within an encoding-modification-decoding scheme.

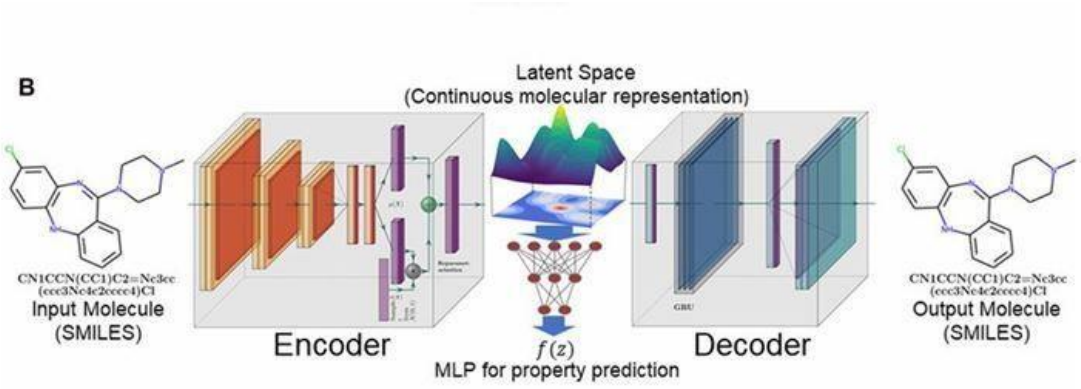


Figure2.18. Variational AutoEncoder architecture with molecule Input and Output.

2.6.7. Graph Neural Networks

2.6.7.1. What is a Graph?

In computer science, a graph is a data structure made up of two parts: nodes (vertices) and edges. A graph is G = (V, E), where V is the set of vertices and E is the set of edges

connecting them. If there are directional relationships between nodes, an edge is directed. The edge is otherwise undirected

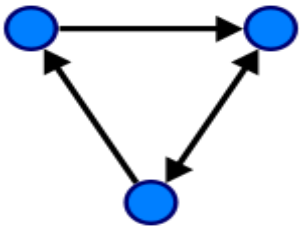


Figure 2.19. Directed Graph

A graph can represent things like social media networks, or molecules. Think of nodes as users, and edges as connections. A social media graph might look like figure16

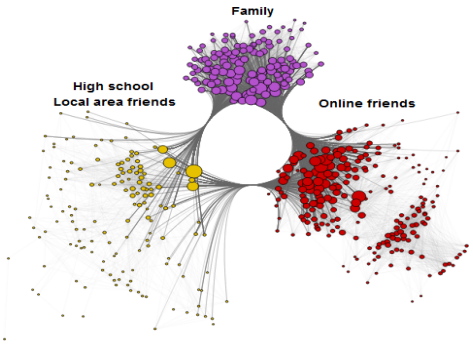


Figure.2.20.A Social Media Graph

A graph is often represented by A, an adjacency matrix, If a graph has ***n*** nodes, ***A*** has a dimension of ***(n × n)***

2.6.7.2. Why Is It Hard To Analyze A Graph?

Graph data is so complex that it's created a lot of challenges for existing machine learning algorithms. The reason is that conventional Machine Learning and Deep Learning tools are specialized in simple data types. Like images with the same structure and size, which we can think of as fixed-size grid graphs. Text and speech are sequences, so we can think of them as line graphs. There are, however, more sophisticated networks with no set structure, a variable size of unordered nodes, and nodes with varying numbers of neighbors.

It also doesn't help that present machine learning methods are built on the premise that instances are unrelated to one another. Because each node is connected to others by various forms of linkages, this is incorrect for graph data

2.6.7.3. Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs) are a class of deep learning methods designed to perform inference on data described by graphs.

GNNs are neural networks that can be directly applied to graphs, and provide an easy way to do node-level, edge-level, and graph-level prediction tasks.

GNNs can do what Convolutional Neural Networks (CNNs) failed to do.

2.6.7.4. Applications of GNNs

Graph-structured data can be found almost everywhere, the challenges that GNNs solve can be divided into five groups:

1. Node Classification: The goal here is to use neighboring labels to identify the labeling of samples (shown as nodes). Problems of this nature are usually trained semi-supervised, with only a portion of the graph labeled.

2. Graph Classification: The goal here is to categorize the entire graph into various groups. It's similar to picture classification, except the aim is now a graph. Graph classification has a wide range of applications, from detecting whether a protein is an enzyme or not in bioinformatics to categorizing articles in natural language processing or social network analysis.

3. Graph visualization: Is a branch of mathematics and computer science that combines geometric graph theory with data visualization. It is focused with the visual representation of graphs that discloses data structures and anomalies, as well as helps the user in comprehending the graphs.

4. Link prediction: In this case, the algorithm must comprehend the link between entities in graphs and attempt to forecast whether two entities are connected. In social networks, it's critical to infer social connections or suggest potential buddies to users. It's also been used to solve problems with recommender systems and forecast criminal linkages.

5. Data clustering: In the form of graphs is referred to as graph clustering. On graph data, there are two different types of clustering. Vertex clustering is a technique for grouping nodes in a graph into highly connected regions based on edge weights or edge distances.

The second type of graph clustering considers graphs to be clustered objects that are grouped together based on their similarity. **[28]**

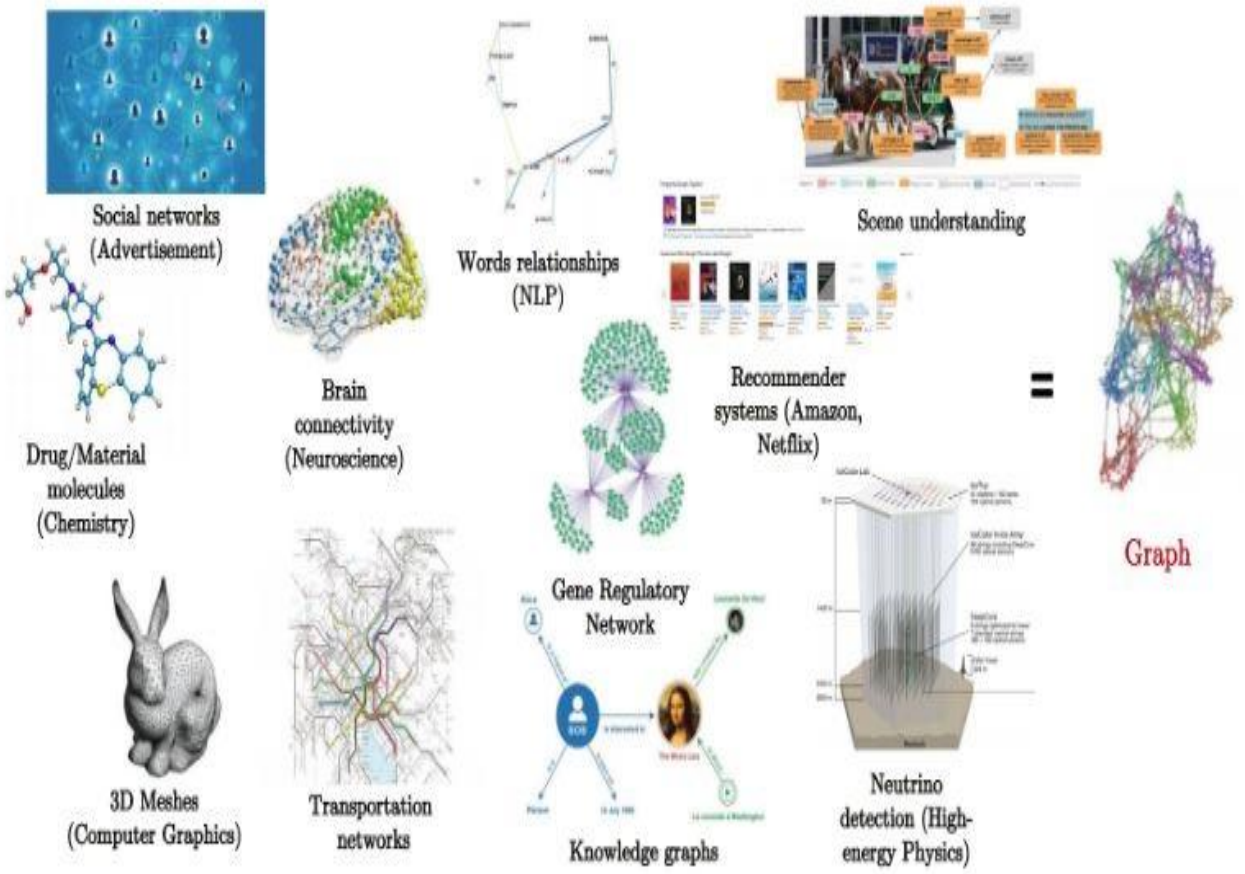


Figure 2.21. Applications of GNN.

Area	Application	References
Graph Mining	- Graph Matching - Graph Clustering	(Riba et al., 2018; Li et al., 2019b) (Zhang et al., 2019c; Ying et al., 2018b; Tsitsulin et al., 2020)
Physics	-Physical Systems Modeling	(Battaglia et al., 2016; Sukhbaatar Ferguset al., 2016; Watters et al., 2017; Hoshen, 2017; Kipf et al., 2018; Sanchez et al., 2018)
Chemistry	-Molecular Fingerprints -Chemical Reaction -Prediction	(Duvenaud et al., 2015; Kearnes et al., 2016) Do et al. (2019)
Biology	-Protein Interface Prediction -Side Effects Prediction -Disease Classification	Fout et al. (2017) Zitnik et al. (2018) Rhee et al. (2018)
Knowledge Graph	-KB Completion) -KG Alignment	(Hamaguchi et al., 2017; Schlichtkrull et al., 2018; Shang et al., 2019 (Wang et al., 2018b; Zhang et al., 2019d; Xu et al., 2019c)
Generation	-Graph Generation	(Shchur et al., 2018b; Nowak et al., 2018; Ma et al., 2018; You et al., 2018a, 2018b; De Cao and Kipf, 2018; Li et al., 2018d; Shi et al., 2020; Liu et al., 2019; Grover et al., 2019)
Combinatorial Optimization	-Combinatorial Optimization	(Khalil et al., 2017; Nowak et al., 2018; Li et al., 2018e; Kool et al., 2019; Bello et al., 2017; Vinyals et al., 2015b; Sutton and Barto, 2018; Dai et al., 2016; Gasse et al., 2019; Zheng et al., 2020a; Selsam et al., 2019; Sato et al., 2019)
Traffic Network	-Traffic State Prediction	(Cui et al., 2018b; Yu et al., 2018; Zheng et al., 2020b; Guo et al., 2019)
Recommendation Systems	-User-item Interaction -Prediction -Social Recommendation	(van den Berg et al., 2017; Ying et al., 2018a) (Wu et al., 2019c; Fan et al., 2019)

Others (Structural)	-Stock Market -Software Defined Networks -AMR Graph to Text	(Matsunaga et al., 2019; Yang et al., 2019; Chen et al., 2018c; Li et al., 2020; Kim et al., 2019) Rusek et al. (2019) (Song et al., 2018a; Beck et al., 2018)
Image	-Social Relationship -Understanding -Image Classification -Visual Question Answering -Object Detection -Interaction Detection -Region Classification -Semantic Segmentation	Wang et al. (2018c) (Garcia and Bruna, 2018; Wang et al., 2018d; Lee et al., 2018b; Kampffmeyer et al., 2019; Marino et al., 2017) (Teney et al., 2017; Wang et al., 2018c; Narasimhan et al., 2018) (Hu et al., 2018; Gu et al., 2018) (Qi et al., 2018; Jain et al., 2016) Chen et al. (2018d) (Liang et al., 2016, 2017; Landrieu and Simonovsky, 2018; Wang et al., 2018e; Qi et al., 2017b)
Other (Non-structural)	-Program Verification	(Allamanis et al., 2018; Li et al., 2016)

Table.2.3. Applications of Graph Neural Networks. **[29]**

2.6.8. Challenges

An outstanding challenge for de novo design is for desired property profiles to reflect the needs of medicinal chemistry more accurately. Although it is useful to demonstrate that methods can optimize molecules toward calculated molecular property profiles, similarity measures or quantitative structure–activity relationship (QSAR) models, drug discovery is multifaceted and current de novo design efforts are limited by a narrow view of the overall process, and obstacle of creating an objective function.

2.7. Previous Works

The generation of new compounds is a crucial step in drug discovery. Virtual screening (VS) of collections of compounds is one of the main computational methods to identify novel molecules.

Graph neural network have been implemented to generate new molecules only in the recent years, which makes it an unstable model, after some researches we found some articles having the same goal as ours, this article **[45]**. Is using a reinforcement learning (RL) to generate new molecules and took ChEMBL dataset that contains 1,347,173 molecules for the learning **[46]** , this large dataset contain many molecules with different atom types. The second article **[46]**. Is using a generative adversarial network(GAN), which also takes ChEMBL dataset for the learning.

The last article **[47]** also used Graph adversarial network(GAN) with two different dataset. first dataset is ZINC, and the second dataset is ChemDiv dataset. Where both of these datasets are a subset with a total of 150,000 molecules for each. In these previous works generating valid molecules was satisfying.

2.8. Discussion

The role of ML and AI in drug discovery is growing and there is a huge request for de novo molecular design methods because of their ability to navigate extreme large chemical spaces more effectively than either VS or a human expert.

Despite the concerns regarding the use of automated methods for molecular design, often relating to the instability, action ability, reactivity, and feasibility of the molecules suggested, engineers now have a variety of tools at their disposal that are proficient generators of sensible molecule structures.

Now the challenge for the field is to evaluate whether the generators and optimization objects are useful for the tasks at hand.

To this, scientists reinforce the categorization between atom-based generators, the maximally expressive paradigm that could encourage them to make molecules that have not been described, de novo molecular design and generative chemistry models remain a contentious issue in the field, but engineers believe there is a potential to learn from collective experience and add these methods to the medical chemistry toolbox. and this what we will talk about in the next chapter.

Summary

To summarize we believe that de novo molecular design is a promising method.

It is a data-driven generative model that does not rely on pre-defined building blocks and rules, which makes it clearly differentiated from traditional methods, by using deep learning, The current literature is composed of a rich array of representation strategies and model architectures.

As in many areas of generative modeling and deep learning, the present day work is largely applied in nature, there are many promising new techniques and architectures from deep generative modeling and optimization more broadly which are ripe to be transferred to the world of molecules.

For example using the AutoEncoder to generate new Molecules based on activity, toxicity… ect, science is trying to have a new representation method rather than using just SMILES, such as chemical grap

hs, and three dimensional chemical structures using a GNN.

For an executive summary:

Firstly generative neural networks are competent for molecular generation

Secondly Objective optimization of generated molecules is essential to de novo drug design,

Lastly the evaluation of molecular generation is a must for de novo drug design.

Chapter 3: Research Methodology

Chapter 3 Reaserch Methodology

Introduction

This research is an applied and scientific research and is carried out as a problem solving approach. It involves mainly several works like, development of molecular structure dataset, analysis of molecular database, development of novel Molecules based on the dataset, The goal of this chapter is to search for the best performing model, showing the usefulness of Graph Convolutional Networks and the advantages of representing data using graphs, The methodology employed will provide a systematic framework of procedures and principles to achieve the objective of this study.

3.1 Research Framework

The research work described in this thesis can be decomposed into four main phases which are outlined in **Figure3.1**. In phase 1, we have searched and thoroughly reviewed the related literature (as detailed in Chapters 2), thus the main issues involved in the fingerprint-based similarity approaches can be found. The data for the work in this thesis has been obtained from MDDR database. **[36]**

There are a number of licensed databases that assign biological activities to compounds.

One of the most popular is the MDDR, which has been compiled from the patent literature since 1988. MDDR database contains over 140,000 compounds with several distinct activities, some of which are therapeutic areas (e.g. Antihypertensive) and some of which are related to specific enzymes or receptors (e.g. Renin inhibitors). One clear advantage of the MDDR is that there is a limited set of activities to choose from, in contrast to some other databases that indicate activity by unstructured text. In the experiment detailed in this thesis, subsets of MDDR database have been used. The datasets have been organized in such a way as to be able to test the performance of our methods.

Chapter 3 Reaserch Methodology

The general research framework for this work is presented in Figure 3.1, which can be divided into four main phases as follows:

Phase 1: The characterization of molecular data and similarity methods where the current and past research will be reviewed in order to identify the requirements for developing novel molecules.

Phase 2: Prepare the dataset. And studying the similarity measurements that can be applied on this research, the detailed experimental set up and results discussed in Chapter 2.

Phase 3: Design Generative Adversarial Network to generate new molecular structure based on requested activity. The detailed experimental set up and results discussed in Chapter 3.

Phase 4: Apply the Graph Neural Network (GNN) on the MDDR dataset, and work on performances Evaluation of the GAN. The detailed experimental set up and results discussed in Chapter 4.

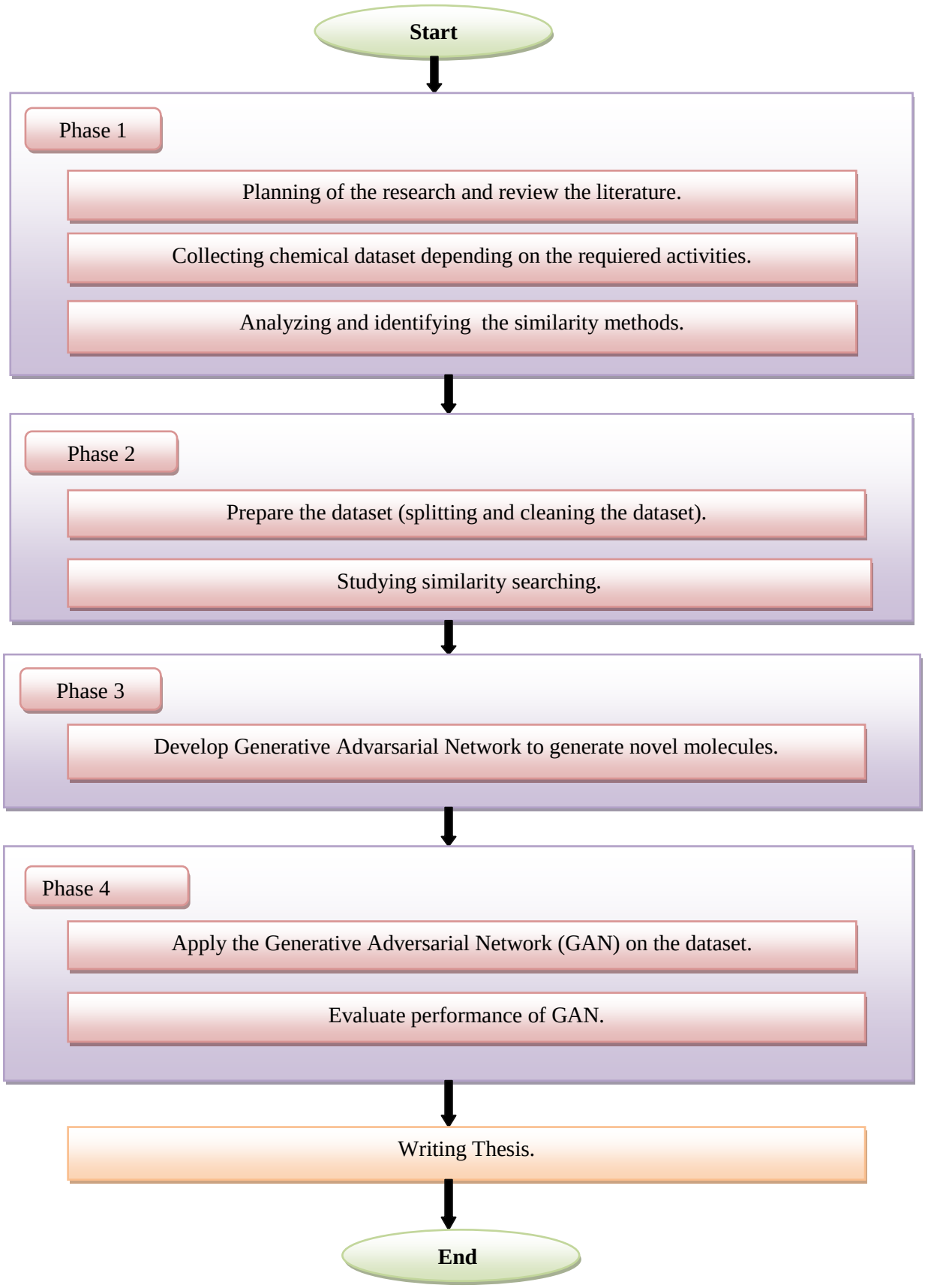


Figure3.1. The General Research Operational Framework.

3.2 Dataset

There are a number of licensed databases that assign biological activities to compounds, that can be used to evaluate the different algorithms in this thesis, but the MDDR database **[36]** is still the most popular, which is covering the patent literature, journals, meetings and congresses. MDDR contains over 140,000 compounds with several hundred distinct activities, with updates adding 10,000 a year to the database.

Although the MDDR (MDL Drug Database Report) contains many chemical compounds in groups of activities with different biological similarities, the data collection is focused on selecting some compounds of different activities.

The database was first filtered using a set of filters from Pipeline Pilot software **[37]**, to remove the duplicates and unprocessed molecules. The remaining database comprised 40751 compounds. Of the total number of compounds, 33947 were inactive compounds and 6804 were active compounds belonging to 20 activity classes. Details of these classes are given in Tables bellow:

Activity Index	Activity class	Active molecules	Pairwise Similarity
07707	adenosine (A1) agonists	207	0.229
07708	adenosine (A2) agonists	156	0.305
31420	renin inhibitors	1130	0.290
42710	CCK agonists	111	0.361
64100	monocyclic _-lactams	1346	0.336
64200	Cephalosporins	113	0.322
64220	Carbacephems	1051	0.269
64500	Carbapenems	126	0.260
64350	Tribactams	388	0.305
75755	vitamin D analogues	455	0.386

Table3.1. MDDR Activity Classes for DS1 Data Set.

Activity Index	Activity Class	Active molecules	Pairwise Similarity
09249	muscarinic (M1) agonists	900	0.111
12455	NMDA receptor antagonists	1400	0.098
12464	nitric oxide synthase inhibitors	505	0.102
31281	dopamine _-hydroxylase inhibitors	106	0.125
43210	aldose reductase inhibitors	957	0.119
71522	reverse transcriptase inhibitors	700	0.103
75721	aromatase inhibitors	636	0.110
78331	cyclooxygenase inhibitors	636	0.108
78348	phospholipase A2 inhibitors	617	0.123
78351	lipoxygenase inhibitors	2111	0.113

Table3.2. MDDR Activity Classes for DS2 Data Set.

The last two data sets (Tables 1-2) are taken from **[38]** and extensively used by many literatures to validate ligand-based virtual screening approaches. **[39, 40, 41, 42-44]** The data sets MDDR1 and MDDR2 contain 10 each for the homogeneous activity classes and heterogeneous ones.

3.3. Steps and Design

Introduction

This section will present and discus all the process we took during the creation of the model, starting by splitting the dataset depending on activity structure, to choosing the best activity class(DS1 or DS2), then reading the data and getting familiar with it. Then present a graph based model, and explain the important steps and modifications (parameters, functions).

3.3.1. Data Preprocessing

While our dataset is provided to us post-registration and has been mostly pre-processed for the competition participants, we still have to do some minor pre-processing before feeding the data to our model because of the resources limitations.

3.3.1.1 . MDDR Database Contents

- **ActivityClass_DS1.txt:** present each class of DS1 with the number of molecules.
- **DS1_SMILES.TXT:** present all the DS1 set of molecules in SMILES format.
- **DS1.SDF:** Present the MDL Molfile of DS1.
- **ActivityClass_DS2.txt:** present each class of DS2 with the number of molecules.
- **DS2_SMILES.TXT:** present all the DS2 set of molecules in SMILES format.
- **DS2.SDF:** Present the MDL Molfile of DS2.

3.3.1.2 Splitting the Dataset

Although the MDDR (MDL Drug Database Report) contains many chemical compounds in groups of activities with different biological similarities, the data collection is focused on selecting some compounds of same activities.

The MDDR-DS1 dataset provide our needs since DS1 present the homogeneous activity classes.

The follow table will explain it:

Activity class	Number of Molecules
adenosine (A1) agonists	207
adenosine (A2) agonists	156
renin inhibitors	1130
CCK agonists	111
monocyclic _-lactams	1346
Cephalosporins	113
Carbacephems	1051
Carbapenems	126
Tribactams	388
vitamin D analogues	455

Table3.3. Represent Activity Classes with Number of Molecules.

3.3.1.3. Reading Data

By using Pandas, we creates efficient data frame objects with default and customized indexing and provides a mechanism to load data objects for our data formats:

```
path='/content/drive/MyDrive/molsclass/DS1_SMILEScsv.csv'
df = pd.read_csv(path, header = None)
print(df)
```

Figure3.2. Loading and creating data frame in Pandas Python

The displayed data is shown in **Figure3.3**:

```
0     CC[C@H](C)[C@H](NC(=O)C[C@H](O)[C@H](CC1CCCCC1...
1     CC(C)C[C@@H](NC(=O)C\C=C\[C@@H](Cc1ccccc1)NC(=...
2     CC(C)C[C@H](O)[C@H](O)[C@H](CC1CCCCC1)NC(=O)[C...
3     CC(C)C[C@H](O)[C@H](O)[C@H](CC1CCCCC1)NC(=O)[C...
4     Cl.CC(C)[C@H](NC(=O)C1Cc2ccccc2CN1)C(=O)N[C@@H...
...                                                 ...
2044  Cl.Cl.NC(=N)NCCC[C@H](NC(=O)[C@@H]1C[C@@H](CN1...
2045  Cl.Cl.NC(=N)NCCC[C@H](NC(=O)[C@@H]1C[C@@H]2CCC...
2046  Cl.CCS(=O)(=O)N[C@@H](C(=O)N1C[C@H](C[C@H]1C(=...
2047  Cl.CCS(=O)(=O)N[C@@H](C(=O)N1[C@H]2CCCC[C@H]2C...
2048  Cl.CCS(=O)(=O)N[C@H](Cc1ccccc1)C(=O)N2[C@H]3CC...
```

Figure3.3. displaying loaded dataset.

Now we need to extract the smiles from the data frame as a list of strings:

```
smiles = data.values
print(smiles)
```

Figure.3.4. Extracting SMILES.

Here we end the data preprocessing phase with no problem while reading it.

3.3.2 Graph Based Models

For graph-based models it is required to convert molecules to a graph representation described as a collection of Data objects each representing a single graph. This is done by specifying the following elements:

• A matrix $x \in R^{N\times M}$ of atom-features where N is the number of nodes and M the number of features for each node

• An edge_index matrix of size $2 \times K$ containing edges expressed as couples of atom indices, where K is the number of edges.

The Data objects are then passed to a DataLoader object which performs shuffling at the beginning of each epoch and combines them into batches of a given size (the batch size we used is 25).

The conversion to graph representation is straightforward thanks to the graph structure that RDKit already uses to internally represent molecules, in our case after converting SMILES. With reference to Figure 3.5 that shows the relevant code

```
def mols2graph(mols, atoms, y=[0]):
    '''
    Convert a list of molecules and activities
    into graphs in the format required by
    PyTorch-Geometric.
    '''

    data = []
    atoms = sorted(atoms)
    MAX_MASS = Atom(max(atoms)).GetMass()
    SEEN_ATOMS = {k:i for i, k in enumerate(atoms)}

    for idx, m in enumerate(mols):
        feats = []
        edges = []
        for a in m.GetAtoms():
            feats.append(atom_features(a, SEEN_ATOMS, MAX_MASS=MAX_MASS))
            i = a.GetIdx()

            for neigh in a.GetNeighbors():
                j = neigh.GetIdx()
                edges.append([i, j])

        feats = torch.tensor(np.array(feats)).float()
        edges = torch.tensor(np.transpose(edges)).long()
        value = y[idx] if not y[0] == 0 else 0
        d     = Data(x=feats, edge_index=edges, y=value)
        data.append(d)

    return data
```

Figure.3.5. Python code to convert molecules into the graph representation required by the Data object provided by PyTorch-Geometric.

For each molecule to convert, we specify:

- The max number of atoms for each molecule
- Atom labels (the atoms supported in our dataset)

These specifications are applied by creating a featurizer with the maxmimum number of atoms per molecule. Atom_labels is a parameter to pass the atomic number of atoms you want to be able to parse. Next step is Filtering out the molecules with too many atoms to reduce the number of unnecessary error messages in later steps.

3.3.3 Generative adversarial network model

GAN is an implicit generative models in the sense that it allows for inference of model parameters without requiring one to specify a likelihood. This proposed model consists of 3 main sections: a Generator G, a Discriminator D and a reward network R.

The generator takes a sample as input and generates an annotated graph representing a molecule. Nodes and edges of the graph are associated with variables denoting atom type and bond type respectively. The discriminator takes both samples from the dataset and the generator and learns to distinguish them. The reward network is used to approximate the reward function of a sample and optimize molecule generation towards non-differentiable metrics using reinforcement learning. Dataset and generated samples are inputs of R, but, differently from the discriminator.

3.3.3.1 Generator

The generator takes D-dimensional vectors $z \in R^D$ sampled from a standard normal distribution, and outputs a graph. Using a simple multi-layer perceptron (MLP) to limits our study to graphs of a pre-chosen maximum size to significantly faster and easier the optimization. By choosing a limited number of nodes, and, for each z, G outputs two continuous and dense objects:

- X that defines atom types.
- A that defines bonds type.

To generate a molecule we obtain discrete, sparse objects X' and A' via categorical sampling from X and A, respectively.

3.3.3.2 Discriminator and Reward Network

The discriminator D and reward network R, both receive a graph as an input, which is the output of the generator. And they output a scalar value each. We choose the same architecture for both networks but do not share parameters between them.
A series of graph convolution layers convolve node signals X' using the graph adjacency tensor A'.

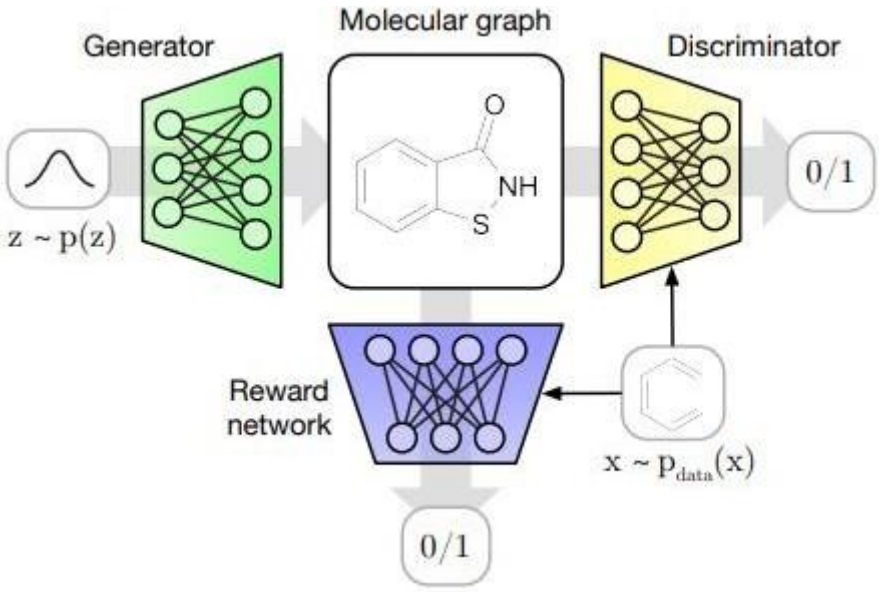


Figure.3.6. Schema of GAN.

3.3.4. Model Layers and Functions

3.3.4.1. MLP layers

It has 3 layers including one hidden layer, it's an organized oriented network. Input layer represent a virtual layer related to the input given by the system, while other layers represent effective layers. This network is used in the back propagation (BP).

3.3.4.2. Tanh Function

Tanh is usually used in hidden layers of a neural network because its values lies between -1 to 1 that's why the mean for the hidden layer comes out be 0 or its very close to 0, hence tanh functions helps in centering the data by bringing mean close to 0 which makes learning for the next layer much easier. So, tanh function is useful.

3.3.4.3. Batch Normalization Layer

Batch Normalization adds normalization layers between each existing layer. It is a technique to normalize activations in intermediate layers of deep neural networks. Its tendency to improve accuracy and speed up training has established BN as a favorite technique in deep learning.

3.3.4.4. Linear Layer

Linear layers are single layers of linear neurons. They are static, with input delays of 0, or dynamic, with input delays greater than 0. You can train them on simple linear time series problems, but often are used adaptively to continue learning while deployed so they can adjust to changes in the relationship between inputs and outputs while being used.

3.3.4.5. Dense Layer

The dense layer is a Neural Network layer that is connected deeply**,** which means each neuron in the dense layer receives input from all neurons of its previous layer. The dense layer is found to be the most commonly used layer in the models.

3.3.5. Model Architecture

This model architecture is based on each part of it, starting with the generator architecture that is fixed for all experiments. We used N = 12 as the maximum number of nodes, T = 9 as the number of atom types(C, N, O, F, P, S, CL, Br, I), and Y = 4 for bond types (single, double, triple and no bond).

These dimensionalities are enough to cover all molecules in MDDR DS1. The generator takes a 32-dimensional vector sampled from a standard normal distribution z, and process it with 3-layer MLP, where first layer starts with 128 units, the second layer contain 256 units and the last one with 512 units, with Tanh as activation functions (to center the data).

Then comes the Batch Normalization layer which enables normalization of parameters between intermediate layers to prevent slower training times. Then lastly comes a linear layer projected to match X and A dimensions. Passing to Discriminator and reward network architecture, where both networks use a Relational GCN encoder with 2 layers, where the first layer starts with 64 units and the second layer with 32 units, to process the input graphs, then we compute a 128-dimensional graph-level representation, processed by MLP layer, and with Tanh as hidden layer activation function, and lastly a Dense layer comes next, serves as an Output Layer with 1 unit.

This model architecture is used to learn from both forward and backward time series data:

In phase 1, the discriminator is only forward propagated and no back-propagation is done.

The Discriminator is trained on real data for n epochs, and sees if it can correctly predict them as real. The Discriminator is also trained on the generated data from the Generator and sees if it can correctly predict them.

In phase 2, The Generator is trained while the Discriminator is idle. After the Discriminator is trained by the generated data of the Generator, we can get its predictions and use the results.

3.4. Conclusion

In this chapter, we introduced the rather research framework for our work in this thesis. According to the research questions formulated in Chapter 1, this framework consists of four major phases. The first phase focused on reviewing the previous works carefully and also prepared the molecular database, so that the algorithms testing processes through the following phases conducted without any difficulty. The last three phases are an applied study on the MDDR dataset using Generative adversarial network to generate novel molecules having same activity class.

Chapter 4: Result and Implementation

Chapter 4 Result and Implimentation

Introduction

In this chapter, we present the details of the implementation of the system and the technical details, the hardware used. the platform or the development environment, the programming language, and the different modules composing the software. Then, we detail our evaluation protocol and discuss the results obtained.

4.1. Development Environment and Used Materials

In order to create and train our models we used the following configuration:

4.1.1 Programming language

Python



Is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed[1].

1. https://www.python.org

4.1.2. Environment

Colab



Training a deep GNN requires powerful computing resources, thus for convenience we have used Google colaboratory. or "Colab" for short, is a product from Google Research. It allows anybody to write and execute arbitrary, python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service (an open-source web application that allows creating and sharing documents that contain live code, equations, visualizations and narrative text) that requires no setup to use, while providing free access to computing resources including GPUs. Colab provides 12GB of RAM and Tesla K80 NVIDIA GPU[2].

4.1.3. Libraries

RDKit



RDKit is a widely used open-source toolkit for cheminformatics. One of its features is the conversion of molecules SMILES strings to 2D and 3D structures. The extension interface presents three tabs: Manage SMILES, Replace fragments, and Positional Analogue Scanning.

It was Developed and used at Rational Discovery for building predictive models for biological activity in 2006, and till the present the development continues to take hand for this tool[3].

---

2. https://colab.research.google.com

3. http://rdkit.org

Keras



Keras is a Python-based open-source library of neural network components. Keras can run on top of TensorFlow, Theano, PlaidML, and other frameworks.

Although the library was designed to be modular and user-friendly, it began as part of a research project for the Open-ended Neuro-Electronic Intelligent Operating System, or ONEIROS. After Keras was released, it was not integrated into Google's TensorFlow core library until 2017[4].

Numpy



NumPy (Numerical Python) is a free and open source Python library that is used in nearly every branch of science and engineering. It is the universal standard in Python for working with numerical data, and it is at the heart of the scientific Python and PyData ecosystems[5].

Pandas



Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,built on top of the Python programming language[6].

PyTorch



PyTorch is an open source machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab (FAIR). It is free and open-source software released under the Modified BSD license[7].

4. https://keras.io/

5. https://numpy.org/

6. https://pandas.pydata.org/

7. https://pytorch.org



TensorFlow is a Python library for high-performance numerical calculations that allows users to create sophisticated deep learning and machine learning applications. Released as open-source software in 2015, TensorFlow has seen tremendous growth and popularity in the data science community[8].

4.2. Proposed experimental protocol

In this project we are adopting a Generative Adversarial network model, which is a type of GNN, for generating new Valid Molecules. And we will discuss the results further.

In the proposed model, GAN use a generator and a discriminator on the MDDR DS1 dataset.

4.2.1. Results and Discussion

4.2.1.1. Test Parameters

Parameter	Value
Max number of atoms	12
Activation function	Tanh
Epoch	25
Batch size	32
Decayed learning ate	0.001

Table4.1. Test Parameters.

Usually, GNNs uses activation function to center the data, that's why in this model we used the Tanh function in each (Generator and Discriminator).

Each epoch is comprised batch with the specific size, to make sure we have passed through the whole dataset.

8. https://www.tensorflow.org/

4.2.1.2. Evaluation measurements

We measure the model depending on three conditions: Validity, Uniqueness and Novelty.

- Validity means there's no disconnected SMILES (molecule), Verifying it by testing if there's a dot (.) in the Smiles string.
- Uniqueness comes after validity phase to remove duplicates, and is defined as the ratio between the number of unique samples and valid samples.
- Novelty is calculated by measuring the ratio between the set of valid samples that are in the dataset and the total number of valid samples generated.

4.2.1.3. Training

In the whole experiment, we use an epoch size of 25 with batch size of 32. And train using a learning rate of 10^{-3} (0.001).

Setting the number of iteration to 10 and max number of generated molecules to 1000. We always report the results of each iteration by revealing valid molecules, unique molecules and novelty. Although the use of GAN prevents us to some extent, undesired behaviors like mode collapse, we notice that our model suffer from crash and freeze when it used too much, especially with the resources we have. As a simple countermeasure, we employ early stopping to avoid completely collapsed modes.

We have chosen 10 molecules from each activity class of the MDDR DS1 dataset, to compare the results and see if there are molecules generated similar to the one in the actual dataset to validate that the generator is predicting new molecules respecting the DS1 activities.

Novelty is calculated as fellow:

X: number of valid generated molecules.

Y: number of molecules of the DS1 dataset.

$$\text{Novelty} = 100 \times (X \div Y)$$

4.2.1.4. Model Results

For 10 iterations, and considering that DS1 dataset contains 5083 molecules, the model provided the following results:

Number of iteration	Valid molecules	Unique molecules
1	554	21
2	275	33
3	403	8
4	720	13
5	655	26
6	600	11
7	130	4
8	617	15
9	468	7
10	346	9

Table4.2. Result of the Model for 10 Iterations.

Table4.3 presents the uniqueness and novelty values calculated using the results obtained from **Table4.2**:

Number of iteration	Validity	Uniqueness	Novelty
1	55.4%	3.7%	10.89%
2	27.5%	12%	5.41%
3	40.3%	1.98%	7.92%
4	72%	1.8%	14.16%
5	65.5%	3.96%	12.88
6	60%	1.83%	11.80%
7	13%	3.07%	2.55%
8	61.7%	2.43%	12.13%
9	46.8%	1.49%	9.20%
10	34.6%	2.6%	6.80%

Table4.3. Calculation of The Uniqueness Novelty Values.

4.2.1.5. Results Analysis

Now using the RDKit to draw set of 2D molecules generated from our GAN model and compares it to some of the MDDR DS1 samples:

First Iteration:

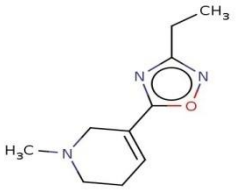
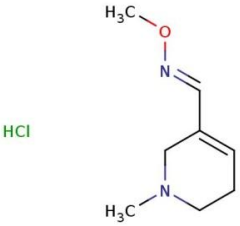
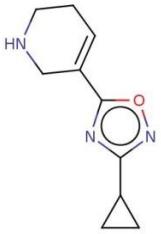
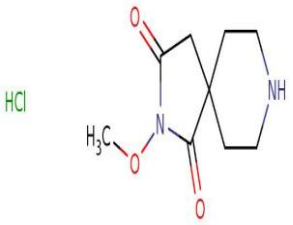
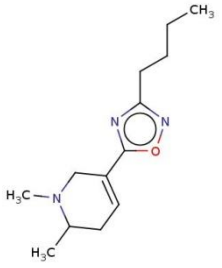
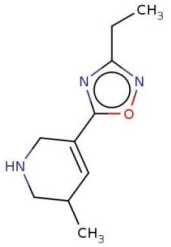
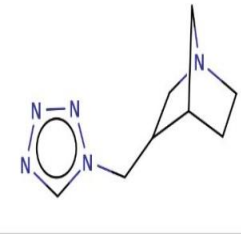
Activity class 1 Smiles	Activity class 1 samples	Activity class 1 Smiles	Activity class 1 samples
CCc1noc(n1)C2=CCC N(C)C2		Cl.CO\N=C\C1=CCCN(C)C1	
C1CC=C(CN1)c2onc (n2)C3CC3		Cl.CON1C(=O)C C2(CCNCC2)C1=O	
CCCCc1noc(n1) C2=CCC(C)N(C)C2		Brc1ccoc1C2=CN3CCC2CC3	
CCc1noc(n1) C2=CC(C)CNC2		C(C1CN2CCC1C2)n3cnnn3	

Table 4.4: Presents the samples of activity 1 from the MDDR dataset

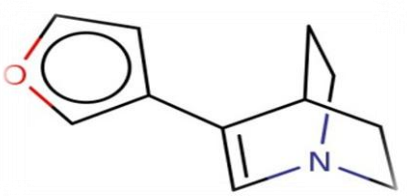
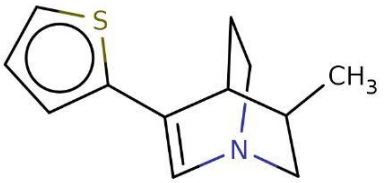
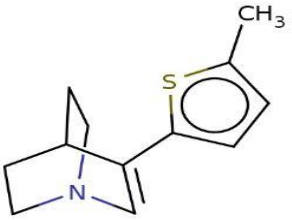
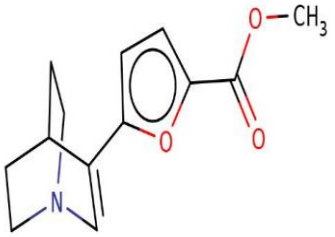
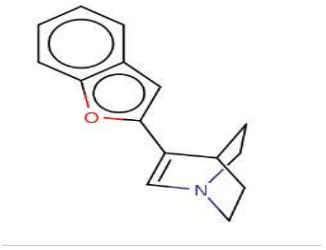
The generated valid molecules smiles	Generated valid molecule samples
C1CN2CCC1C(=C2)c3cocc3	
C1CN2CCC1C(=C2)c3cccs3	
Cc1ccc(s1)C2=CN3CCC2CC3	
COC(=O)c1oc(cc1)C2=CN3CCC2CC3	
C1CN2CCC1C(=C2)c3oc4ccccc4c3	

Table 4.5: Presents the generated samples from GAN model in iteration 1.

Here we are compared 8 samples taken from the activity 1 in MDDR DS1 dataset and 5 generated samples from the model.

Second Iteration:

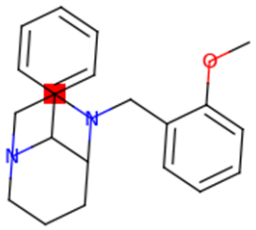
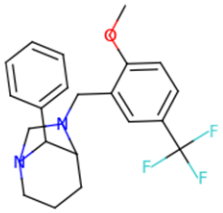
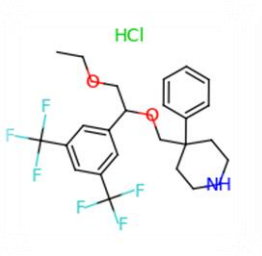
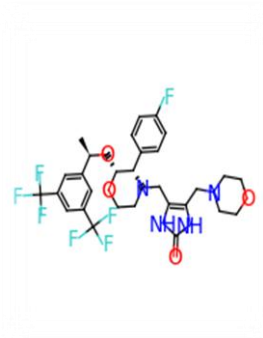
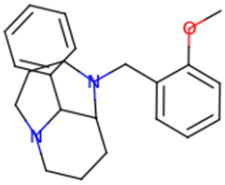
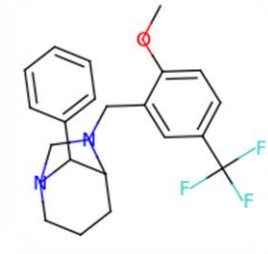
Activity class 2 Smiles	Activity class 2 samples	Activity class 2 Smiles	Activity class 2 samples
C[C@@H](O[C@H]1OCCN(CC2=C(CN3CCOCC3)NC(=O)N2)[C@H]1c4ccc(F)cc4)c5cc(F)cc(c5)C(F)(F)F		COc1ccccc1CN2CCN3CCCC2C3c4ccccc4	
COc1ccc(cc1CN2CN3CCCC2C3c4ccccc4)C(F)(F)F		Cl.CCOCC(OCC1(CCNCC1)c2ccccc2)c3cc(cc(c3)C(F)(F)F)C(F)(F)F	
C[C@@H](O[C@H]1OCCN(CC2=C(CN3CCOCC3)NC(=O)N2)[C@H]1c4ccc(F)cc4)c5cc(cc(c5)C(F)(F)F)C(F)(F)F		COc1ccccc1CN2CCCN3CCCC2C3c4ccccc4	
Cl.COCC(OCC1(CCNCC1)c2ccccc2)c3cc(cc(c3)C(F)(F)F)C(F)(F)F		O=C(Cc1ccccc1)NC(Cc2ccccc2)NC(=O)[C@H]3CCCC[C@H]3NC(=O)c4c[nH]c5ccccc45	

Table 4.6: Presents the samples of activity 2 from the MDDR dataset.

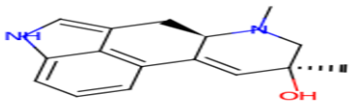
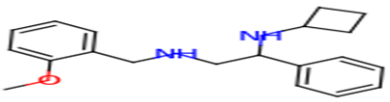
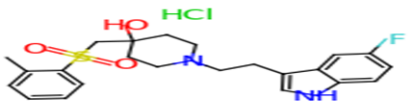
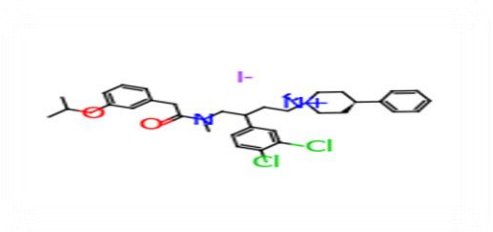
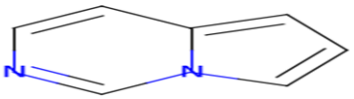
The generated valid molecules smiles	Generated valid molecule samples
C[C@@H](O[C@H]1OCCN(CC2=NC(=O)NN2P(=O)(O)O)[C@H]1c3ccc(F)cc3)c4cc(cc(c4)C(F)(F)F)C(F)(F)F	
C[C@@H](O[C@H]1OCCN(Cc2n[nH]c(OP(=O)(O)O)n2)[C@H]1c3ccc(F)cc3)c4cc(cc(c4)C(F)(F)F)C(F)(F)F	
[K+].CN(Cc1ccccc1)C(=O)[C@H](Cc2ccc3ccccc3c2)NC(=O)[C@@H]4C[C@@H](O)CN4C(=O)c5cn(CCCCc6nnn[n-]6)c7ccccc57	
[I].CC(C)Oc1cccc(CC(=O)N2CCCC(CC[N+]3(C)CCC4(CC3)NCCc5ccccc45)(C2)c6ccc(Cl)c(Cl)c6)c1	
CN1C[C@@](C)(O)C=C2[C@H]1Cc3c[nH]c4cccc2c34	

Table 4.7: Presents the generated samples from GAN model in iteration 2.

Here we are compared 8 samples taken from the activity 2 in MDDR DS1 dataset and 5 generated samples from the model.

Sixth Iteration:

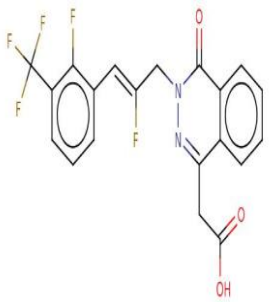
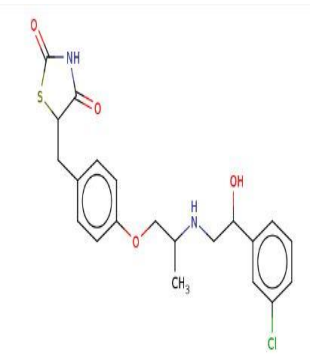
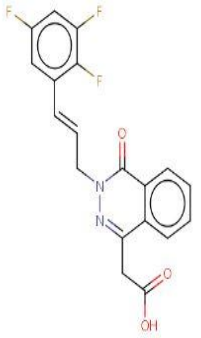
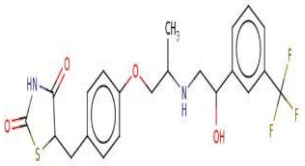
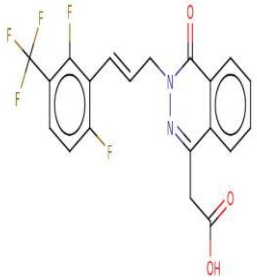
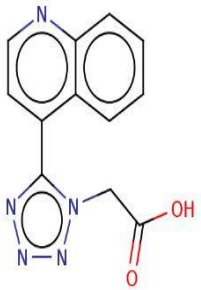
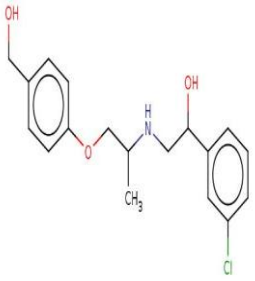
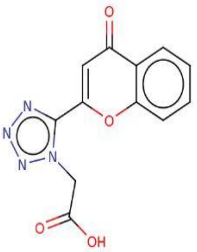
Activity class 5 Smiles	Activity class 5 samples	Activity class 5 Smiles	Activity class 5 samples
OC(=O)CC1=NN(C\C(=C\c2cccc(c2F)C(F)(F)F)\F)C(=O)c3ccccc13		CC(COc1ccc(CC2SC(=O)NC2=O)cc1)NCC(O)c3cccc(Cl)c3	
OC(=O)CC1=NN(C\C=C\c2cc(F)cc(F)c2F)C(=O)c3ccccc13		CC(COc1ccc(CC2SC(=O)NC2=O)cc1)NCC(O)c3cccc(c3)C(F)(F)F	
OC(=O)CC1=NN(C\C=C\c2c(F)ccc(c2F)C(F)(F)F)C(=O)c3ccccc13		OC(=O)Cn1nnnc1c2ccnc3ccccc23	
CC(COc1ccc(CO)cc1)NCC(O)c2cccc(Cl)c2		OC(=O)Cn1nnnc1C2=CC(=O)c3ccccc3O2	

Table 4.8: Presents the samples of activity 5 from the MDDR dataset.

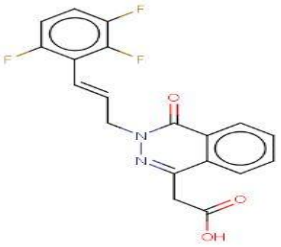
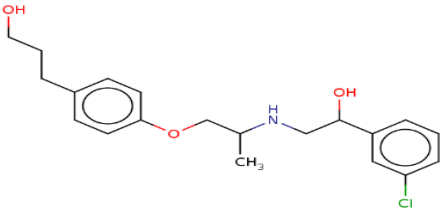
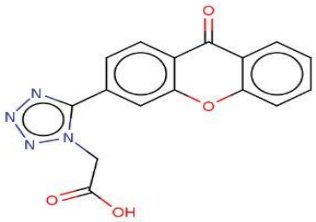
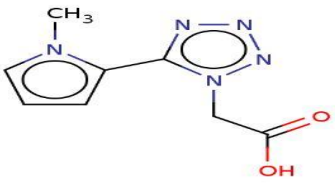
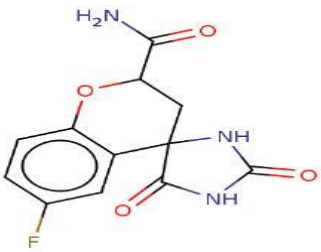
The generated valid molecules smiles	The generated valid molecules samples
OC(=O)CC1=NN(C\C=C\c2c(F)ccc(F)c2F)C(=O)c3ccccc13	
CC(COc1ccc(CCCO)cc1)NCC(O)c2cccc(Cl)c2	
OC(=O)Cn1nnnc1c2ccc3C(=O)c4ccccc4Oc3c2	
Cn1cccc1c2nnnn2CC(=O)O	
NC(=O)C1CC2(NC(=O)NC2=O)c3cc(F)ccc3O1	

Table 4.9: Presents the generated samples from GAN model in iteration 6.

Here we are compared 8 samples taken from the activity 5 in MDDR DS1 dataset and 5 generated samples from the model.

Ninth Iteration:

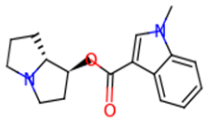
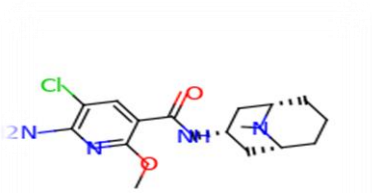
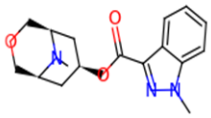
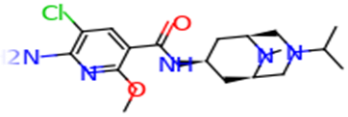
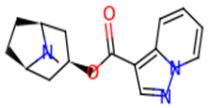
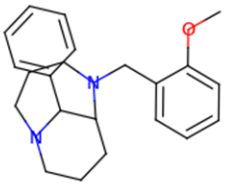
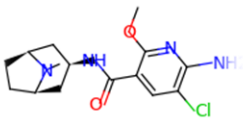
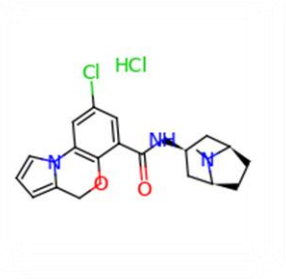
Activity class 6 Smiles	Activity class 6 samples	Activity class 6 Smiles	Activity class 6 samples
Cn1cc(C(=O)O[C@H]2CCN3CCC[C@H]23)c4ccccc14		COc1nc(N)c(Cl)cc1C(=O)N[C@H]2C[C@H]3CCC[C@@H](C2)N3C	
CN1[C@@H]2COC[C@H]1C[C@H](C2)OC(=O)c3nn(C)c4ccccc34		COc1nc(N)c(Cl)cc1C(=O)N[C@@H]2C[C@@H]3CN(C[C@H](C2)N3C)C(C)C	
CN1[C@@H]2CC[C@H]1C[C@H](C2)OC(=O)c3cnn4ccccc34		CN1[C@@H]2COC[C@H]1C[C@H](C2)OC(=O)c3c[nH]c4ccccc34	
COc1nc(N)c(Cl)cc1C(=O)N[C@H]2C[C@H]3CC[C@@H](C2)N3C		Cl.CN1[C@@H]2CC[C@H]1C[C@H](C2)NC(=O)c3cc(Cl)cc4c3OCc5cccn45	

Table 4.10: Presents the samples of activity 6 from the MDDR dataset.

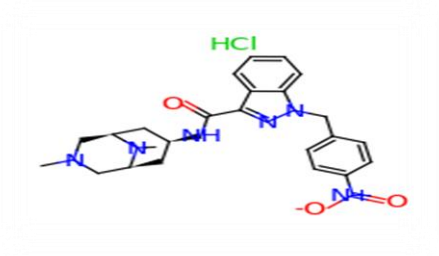
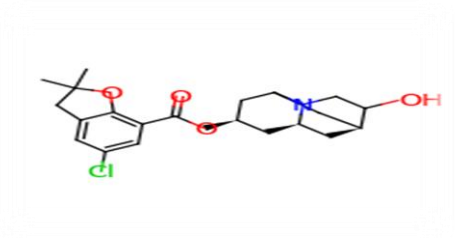
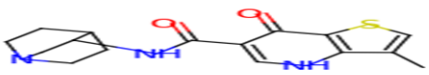
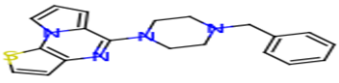
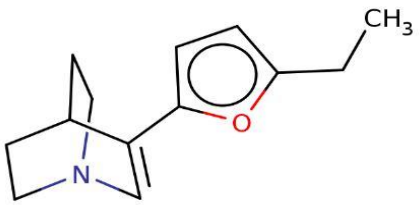
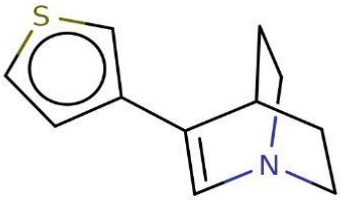
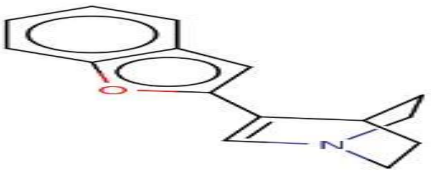
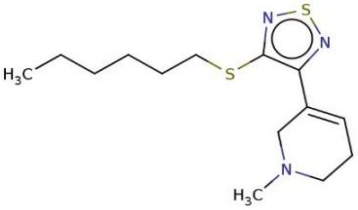
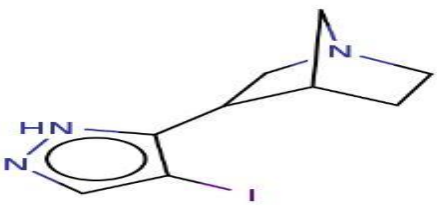
The generated valid molecules smiles	The generated valid molecules samples
Cl.CN1C[C@H]2C[C@H](C[C@@H](C1)N2C)NC(=O)c3nn(Cc4ccc(cc4)[N+](=O)[O-])c5ccccc35	
CC1(C)Cc2cc(Cl)cc(C(=O)O[C@@H]3C[C@@H]4C[C@H]5CC(C3)N4CC5O)c2O1	
Cl.Nc1c(Cl)cc2C(=O)N([C@@H]3CN4CCC3CC4)C(=O)c5cccc1c25	
Cc1csc2C(=O)C(=CNc12)C(=O)NC3CN4CCC3CC4	
C(N1CCN(CC1)c2nc3ccsc3n4cccc24)c5ccccc5	

Table 4.11: Presents the generated samples from GAN model in iteration 9.

Chapter 4 Result and Implimentation

Here we are compared 8 samples taken from the activity 6 in MDDR DS1 dataset and 5 generated samples from the model.

In order to show the results of our model and presents the generated molecules we filtered the duplication from the valid molecules generated in the 2 last iterations then print the samples in 2D to have just the unique samples, the next table shows this results.

The unique smiles	The unique samples
CCc1oc(cc1)C2=CN3CCC2CC3	
C1CN2CCC1C(=C2)c3ccsc3	
C1CN2CCC1C(=C2)c3oc4ccccc4c3	
CCCCCCSc1nsnc1C2=CCCN(C)C2	
Ic1cn[nH]c1C2CN3CCC2C3	

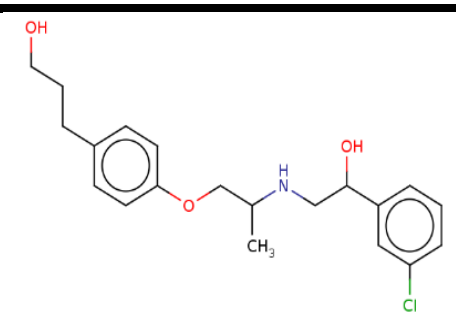
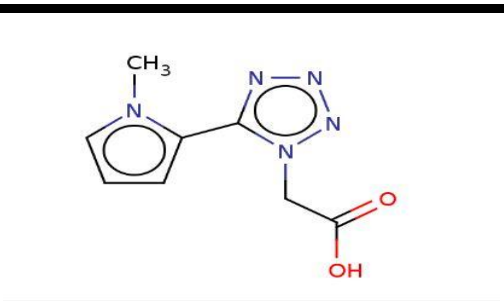
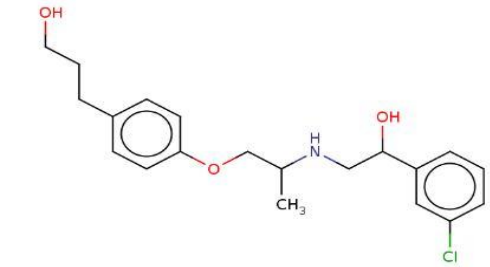
CC(COc1ccc(CCCO)cc1)NCC(O)c2cccc(Cl)c2	
Cn1cccc1c2nnnn2CC(=O)O	
OC(=O)CC1=NN(C\C=C\c2c(F)ccc(F)c2F)C(=O)c3ccccc13	
CC(COc1ccc(CCCO)cc1)NCC(O)c2cccc(Cl)c2	

Table4.12: The unique molecules generated after deleting the duplication in iteration 10.

4.2.1.6. Discussion

After analyzing the values obtained in table 4.2 and table 4.3, we noticed that our model has converged to very high validity outputs of 72%, with novelty up to 14%.

Also, after many iterations and comparisons between the obtained results and the samples from the trained data, we note a significant similarity between the MDDR molecules (samples) and the generated molecules, The generated graphs look like trained data, and refers to the same activities.

Putting in fact that our model is constrained to generate up to 9 atoms .And by choice we set up to 12 atoms per molecule taking in consideration that MDDR dataset containing heavy molecules with large number of atoms. Increasing the number of atoms leads to having a large space of molecules which damage the model. This can explain the difference in unique score since the chance of generating different molecules in a smaller space preferable.

4.3. Obstacles

It is known that the implementation of deep learning algorithms requires resources such as large datasets and solid computer hardware resources (CPU, GPU and storage space) to form models.

During the implementation of our project, we encountered many problems due to our resources as a student we quote:

- The diversity of the datasets and the very large number of molecules of drugs existing in the world.

Google Colab was the most appropriate service to implement our project, but always with limitations and from them we mention:

- Only 12 hours of continuous work per session.
- Use only one GPU per session.
- Colab also needs a stable internet connection like all cloud services.

infortunately internet in Algeria is unstable, which makes the implementation of our project even more difficult.

4.4. Conclusion

We have presented in the last chapter the programming language, libraries and environment used to implement our project. After that we shown the model parameters and the measures we took during the evaluation, passing to results with analysis and finely, the obstacles we faced.

General Conclusion

General Conclusion

To design new drugs from scratch automatically is the dream of all drug researchers, This study presented a GNN method which is generative adversarial networks for automatic generation of new molecules with the goal of improving the relation between the activity structure and the biological activity of the drugs and The importance that Quantitative Structure-Activity Relationship (QSAR) models have in Chemistry has been highlighted along with some of the challenges that are involved in developing them. Many choices can be made in how data is represented and what algorithms are used, ranging from simple linear models on molecule descriptors of different kind to Deep methods able to work on graph data representation. Special attention has been given to how to compute or derive somehow information about the "inner-workings" of a model as this information is valuable both to understand the performance of the model and to give a reasonable justification to the end-user which is going to use it.

The project's primary objective was to show how Graph Neural Networks and Explainable AI two promising aspects of modern Deep Learning research can be integrated into a generation method to be used as a tool to aid researchers in their job of discovering new drugs and evaluating their properties. In this endeavour, Generator adversarial networks (GANs) have been shown to be an especially suited method to work on molecules because of the natural graph interpretation that is adopted when describing molecules. This suggests that an ensemble approach using the scores returned by multiple methods and act according to their mean rather than following only one possible interpretation of the model which is a good approach to follow. This doesn't exclude that better input-attribution methods may be developed in the future, which would also improve the quality of the generation.

Limitations

In recent years, with the assistance of deep learning technologies represented by GNNs, remarkable progress has been made in many aspects related to this target. However, there are also some obvious problems that need to be addressed. For example, GANs generates small molecular graphs with the desired chemical activity by combining GANs and reinforcement learning. It can generate chemically valid molecules if the number of heavy atoms used for molecular representation is small. However, when this number increase, many disconnected molecular graphs are generated. Furthermore, as the most popular deep learning models for dealing with chemical molecule data, GNNs also have distinct problems such as 'oversmoothing'. When stacking too many multiple layers, the features of different nodes in a graph become indistinguishable, which seriously damages the performance of models, Given this limitation, most current GNNs are no deeper than four layers. However, with the increase in model depth, the representation ability of the model will also increase. The effectiveness of ultra-deep neural networks has been proved in the field of computer vision. The current state-of the art CNNs are usually >100 layers.

Future Steps

GNNs models and specially GAN improved good results in molecular generation with a valid, unique and totally connected molecules with the desires activity structure, that's make this methods of GNNs for molecule learning need to be further explore. Also what comes next will be usefull for drug discovery, where we can suggest further subject:

- Calculation of Similarity of novel molecules to get classified in groups of activities.
- Calculate toxicity of novel molecules.
- Calculate solubility of generated molecules.

References

References

[1]. DiMasi, J.A. et al. (2016) Innovation in the pharmaceutical industry: new estimates of R&D costs. J. Health. Econ.

[2]. Schneider, G. and Fechner, U. (2005) Computer-based de novo design of drug-like molecules. Nat. Rev. Drug Discov. 4,

[3].Nishibata, Y. and Itai, A. (1993) Confirmation of usefulness of a structure construction program based on three-dimensional receptor structure for rational lead generation. J. Med. Chem. 36,

[4].Verlinde, C.L. et al. (1992) In search of new lead compounds for trypanosomiasis drug design: a protein structure-based linked-fragment approach. J. Comput. Aided Mol. Des. 6,

[5]. Pagadala, N.S. et al. (2017) Software for molecular docking: a review. Biophys. Rev. 9,

[6]. Zhavoronkov, A. et al. (2019) Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nat. Biotechnol. 37,

[7]. Yang, Y. et al. (2020) Discovery of highly potent, selective, and orally efficacious P300/CBP histone acetyltransferases inhibitors. J. Med. Chem. 63,

[8]. MIT Technology Review, www.technologyreview.com/10-breakthrough-technologies/2020/ [accessed 10.2.21].

[9]. Ekins, S. et al. (2019) Exploiting machine learning for end-to-end drug discovery and development. Nat. Mater. 18.

[10]. Chen, H. et al. (2018) The rise of deep learning in drug discovery. Drug Discov. Today 23

[11]. Kipf, T.N. and Welling, M. (2016) Semi-supervised classification with graph convolutional networks. arXiv 2016,

[12]. Jin, W. et al. (2018) Junction tree variational autoencoder for molecular graph generation. Proc. Machine Learn. Res.

[13]. Mak, K.-K. and Pichika, M.R. (2019) Artificial intelligence in drug development: present status and future prospects. Drug Discov. Today

[14]. J.W. Scannell, J. Bosley, When quality beats quantity: decision theory, drug discovery, and the reproducibility crisis, PLoS ONE 11 (2016) e0147215.

[15]. T.J. Sejnowski, The unreasonable effectiveness of deep learning in artificial intelligence, Proc Natl Acad Sci U S A 117 (2020) 30033–30038.

[16]. Emmert-Streib F, Yang Z, Feng H, Tripathi S, Dehmer M. An introductory review of deep learning for prediction models with big data. Front Artif Intell Published online February 28, 2020. https://doi.org/10.3389/frai.2020.00004.

[17]. Savita Shetty, "Deep Learning Algorithms and Applications in Computer Vision," July 2019. [Online]. Available: https://www.researchgate.net/publication/335799065.

[18]. Y. LeCun, "There can be no intelligence without learning.," 1 January 2020. [Online]. Available: https://twitter.com/ylecun/status/1212381611838361602.

[19]. Li Y, Vinyals O, Dyer C, et al (2018) Learning Deep Generative Models of Graphs. Iclr 1–16. https://doi.org/10.1146/annurev-statistics-010814-020120.

[20]. Pirashvili, M., Steinberg, L., Belchi Guillamon, F., Niranjan, M., Frey, J. and Brodzki, J. (2019). *Improved understanding of aqueous solubility modeling through topological data analysis.*

[21]. Gasteiger, J. (n.d.). *Solved and Unsolved Problems in Chemoinformatics*.

[22]. Gillet, V. J., & Leach, A. R. (2006). Chemoinformatics. In *Comprehensive Medicinal Chemistry II* (Vol. 3, pp. 235–264). Elsevier Ltd. https://doi.org/10.1016/b978-0-12-409547-2.14327-6.

[23]. Elton, D. C., Boukouvalas, Z., Fuge, M. D., & Chung, P. W. (2019). Deep learning for molecular design - A review of the state of the art. In *Molecular Systems Design and Engineering* (Vol. 4, Issue 4, pp. 828–849). Royal Society of Chemistry. https://doi.org/10.1039/c9me00039a

[24]. Sattarov, B., Baskin, I., Horvath, D., Marcou, G., Jannik Bjerrum, E., Varnek, A., De, al, & Baskin, I. I. (2018). Novo Molecular Design by Combining Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping. *Journal of Chemical Information and Modeling*, *59*(3). https://doi.org/10.1021/acs.jcim.8b00751ï

[25]. Bero, S. A., Muda, A. K., Choo, Y. H., Muda, N. A., & Pratama, S. F. (2017). Similarity Measure for Molecular Structure: A Brief Review. *Journal of Physics: Conference Series*, *892*(1). https://doi.org/10.1088/1742-6596/892/1/012015.

[26]. Willett P 2009 Similarity methods in chemoinformatics (Annu. Rev. Inf. Sci. Technol. vol 43), pp 1–117

[27]. Papadopoulos, K., Giblin, K. A., Janet, J. P., Patronov, A., & Engkvist, O. (2021). De novo design with deep generative models based on 3D similarity scoring. *Bioorganic and Medicinal Chemistry*, *44*. https://doi.org/10.1016/j.bmc.2021.116308

[28]. Lim, J., Ryu, S., Kim, J. W., & Kim, W. Y. (2018). Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of Cheminformatics*, *10*(1). https://doi.org/10.1186/s13321-018-0286-7

[29]. Berglund, M., Raiko, T., Honkala, M., Kärkkäinen, L., Vetek, A., & Karhunen, J. (n.d.). *Bidirectional Recurrent Neural Networks as Generative Models*.

[30]. Arús-Pous, J., Patronov, A., Bjerrum, E. J., Tyrchan, C., Reymond, J. L., Chen, H., & Engkvist, O. (2020). SMILES-based deep generative scaffold decorator for de-novo drug design. *Journal of Cheminformatics*, *12*(1). https://doi.org/10.1186/s13321-020-00441-8

[31]. Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning

[32]. A. sehgal and N. kehtarnavaz, "A Convenutional neural networks Smartphone apps for Real-Time Voice Activity Detection", IEEE Access , 2018, doi:10.1109/ACCESS.2018.2800728.

[33]. Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J., & Chen, H. (2018). Application of Generative Autoencoder in De Novo Molecular Design. *Molecular Informatics*, *37*(1). https://doi.org/10.1002/minf.201700123

[34]. Simon Haykin. Neural Networks: A Comprehensive Foundation (3rd Edition). Prentice-Hall, Inc., USA, 2007.

[35]. Bengio, Y.; Courville, A.; Vincent, P., Representation learning: A review and new perspectives.

[36]. The MDL Drug Data Report Database is available from MDL Information Systems Inc. http://www.mdli.com. (accessed August 31, 2009)

[37]. Pipeline Pilot Basic Chemistry Component collection, SciTegic Inc.

[38]. Kier, L. B., & Hall, L. H. (1976). Molecular connectivity VII: specific treatment of heteroatoms. Journal of pharmaceutical sciences, 65(12), 1806-1809.

[39]. Hansch, C. (1969). Quantitative approach to biochemical structure-activity relationships. Accounts of Chemical Research, 2(8), 232-239.

[40]. Zahradnik, R. (1962). Correlation of the biological activity of organic compounds by means of the linear free energy relationships. Experientia, 18(11), 534-536.

[41]. Wiener, H. (1947). Structural determination of paraffin boiling points. Journal of the American Chemical Society, 69(1), 17-20.

[42]. Randic, M. (1975). Characterization of molecular branching. Journal of the American Chemical Society, 97(23), 6609-6615.

[43]. Hosoya, H. (1971). Topological index. A newly proposed quantity characterizing the topological nature of structural isomers of saturated hydrocarbons. Bulletin of the Chemical Society of Japan, 44(9), 2332-2339.

[44]. Trinajstic, N. (1992). Chemical graph theory. CRC press.

[45]. Olivecrona, M., Blaschke, T., Engkvist, O., & Chen, H. (2017). Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, *9*(1). https://doi.org/10.1186/s13321-017-0235-x.

[46]. Prykhodko, O., Johansson, S. V., Kotsias, P. C., Arús-Pous, J., Bjerrum, E. J., Engkvist, O., & Chen, H. (2019). A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, *11*(1). https://doi.org/10.1186/s13321-019-0397-9.

[47]. Putin, E., Asadulaev, A., Ivanenkov, Y., Aladinskiy, V., Sanchez-Lengeling, B., Aspuru-Guzik, A., & Zhavoronkov, A. (2018). Reinforced Adversarial Neural Computer for de Novo Molecular Design. *Journal of Chemical Information and Modeling*, *58*(6), 1194–1204. https://doi.org/10.1021/acs.jcim.7b00690.