**Ministry of Higher Education and Scientific Research**
وزارة التعليم العالي والبحث العلمي

**Badji Mokhtar Annaba University**
**Université Badji Mokhtar – Annaba**
**Faculté des Technologie**
**Département d'informatique**

**جامعة باجي ختار – عنابـــــــــة**
**كلية تكنولوجيا**
**قســـــــم الإعلام الآلي**

# Thesis

Presented to obtain the diploma of

# Doctorate

**Specialty:** Artificial Intelligence

The Computer Science Research Laboratory (LRI)

By:
**Oussama Harrouche**

Title:
## Recommender Systems in Bioinformatics

Thesis defended in 13/07/2026 in front of the committee:

| N° | Last Name and First Name | Grade | Institution | Quality |
|---|---|---|---|---|
| 01 | BAHI Halima | Prof | Badji Mokhtar Annaba University | Chair |
| 02 | MOHAMED BENALI Yamina | Prof | Badji Mokhtar Annaba University | Supervisor |
| 03 | AZIZI Nabiha | Prof | Badji Mokhtar Annaba University | Examiner |
| 04 | BENMACHICHE Abdelmadjid | MCA | Chadli Bendjedid-El Taref University | Examiner |
| 05 | LACHTAR Nadia | MCA | National Higher School of Technology and Engineering Annaba | Examiner |

# Acknowledgments

First and foremost, I am deeply grateful to Allah the Almighty for granting me the strength, patience, and health necessary to reach this stage in life, while blessing me with the presence and support of the people around me.

I would like to express my sincere gratitude to my thesis supervisor, Ms. Yamina Mohamed Ben Ali, for her invaluable guidance, unwavering support, patience, and enthusiasm throughout these five years, without which this work would not have been possible.

I also extend my thanks for the members of the jury committee for dedicating their precious time and effort to help evaluate and criticize this humble work.

Without forgetting to acknowledge all the teachers who contributed in my educational journey throughout my ten years of academic study in Mila, Setif, and Annaba.

Finally, I am deeply indebted to my family, colleagues, and dear friends, with whom I shared both joyful and challenging moments, and who were always there with their constant support and encouragement.

# Abstract

The rapid growth of biological data that is paired with the high cost, time consumption, and failure rates associated with traditional drug discovery pipelines have intensified the need for computational approaches capable of efficiently predicting intermolecular interactions and accelerating therapeutic development. Bioinformatics, coupled with machine learning techniques and recommender system paradigms, has emerged as a powerful framework for modeling complex biological relationships such as drug-target and protein-protein interactions, and drug-disease associations. However, these tasks are challenged by data sparsity, high dimensionality, heterogeneity of biological entities, and the presence of orphan proteins or drugs with limited to none interaction knowledge.

This thesis investigates the exploitation of recommender systems and deep learning models for intermolecular interaction prediction, drug repurposing, and *de novo* drug design by formulating these problems as matrix completion and matrix expansion tasks. After providing a comprehensive background on bioinformatics foundations, machine learning principles, and recommender system methodologies, this work analyzes their applicability and limitations in biomedical contexts.

The first contribution proposes a convolutional-recurrent deep learning-based recommender system (RSCRDL) that leverages convolutional-recurrent synergy to capture both local structural patterns and sequential dependencies embedded within biological sequences for a more efficient drug-target interaction prediction. Additionally, an enhanced greedy-variant, that is RSCRDL_F, is introduced to further improve feature representation and predictive performance through integrating explicit molecular features of drugs. Extensive experimental evaluations demonstrate the effectiveness and robustness of the proposed models across multiple datasets and biological targets (proteins that are part of the Cytokine and Cyclin-dependent kinase families and their interactions with drugs), providing viable drug candidates as recommendations to both regular and orphan protein sequences, while ensuring their relevance both computationally and biologically.

The second contribution introduces a transformer-based recommender and translator frameworks (RSSSTr and TSSTr) that are designed to exploit self-attention mechanisms for modeling long-range dependencies and complex interaction patterns. By integrating our newly proposed fingerprint indexation strategy alongside our enhanced encoder architecture; these models enable the accurate intermolecular interaction prediction (matrix completion task), while accomplishing our newly proposed matrix expansion task, where both approaches collaborate to provide newly generated, yet relevant drug candidates for both regular and mutated protein sequences. The proposed approaches are validated on diverse biological targets, including cytokines, Cyclin-dependent kinases, G-protein-coupled receptors, and protein that are part of the HIV, and SARS-CoV-2 organisms, showing strong performance in both drug repurposing and novel compound design tasks.

Overall, this thesis demonstrates that deep learning-driven recommender systems provide scalable, and biologically meaningful solutions for interaction prediction and drug discovery, offering a promising computational alternative to traditional experimental pipelines.

**Keywords:** Bioinformatics, Deep learning, Drug discovery, Drug Repurposing, Recommender Systems.

# الملخص

أدى النمو السريع للبيانات البيولوجية، وما يصاحبه من ارتفاع في تكاليف وأوقات إتمام ومعدلات فشل مناهج اكتشاف الأدوية التقليدية، إلى زيادة الحاجة إلى نماذج حسابية قادرة على التنبؤ بكفاءة بالتفاعلات ما بين الجزيئات وتسريع تطوير العلاجات. وقد برزت المعلوماتية الحيوية، بالاقتران مع تقنيات التعلم الآلي ونماذج أنظمة التوصية، كنموذج عمل فعال لنمذجة العلاقات البيولوجية المعقدة، مثل تفاعلات الأدوية مع الأهداف، تفاعلات البروتينات فيما بينهم، وعلاقات الأدوية بالأمراض. ومع ذلك، تواجه هذه المهام تحديات تتمثل في ندرة البيانات، وارتفاع أبعادها، وتنوع الكيانات البيولوجية، بالإضافة إلى وجود بروتينات يتيمة أو أدوية ذات تفاعلات محدودة أو منعدمة.

تسعى هذه الأطروحة إلى استغلال أنظمة التوصية ونماذج التعلم العميق للتنبؤ بالتفاعلات بين الجزيئات، وإعادة استخدام الأدوية، وتصميم الأدوية، وذلك من خلال صياغة هذه المشكلات كمهام إكمال المصفوفات وتوسيعها. بعد تقديم خلفية شاملة حول أسس المعلوماتية الحيوية، ومبادئ التعلم الآلي، ومنهجيات أنظمة التوصية؛ يحلل هذا العمل مدى قابليتها للتطبيق وحدودها في السياقات الطبية الحيوية.

يقترح البحث الأول نظام توصية قائم على التعلم العميق التلافيفي-التكراري(RSCRDL)، والذي يستفيد من التآزر بين شبكات التلافيف والشبكات التكرارية لالتقاط كل من الأنماط البنيوية المحلية والتبعيات التسلسلية الكامنة بالتسلسلات البيولوجية، وذلك لتحسين كفاءة مهمة التنبؤ بتفاعلات الأدوية مع أهدافها. بالإضافة إلى ذلك، تم تقديم نسخة محسّنة جشعة من النموذج، ألا وهي RSCRDL_F، وذلك لزيادة تحسين تمثيل الميزات وقدرات التنبؤ من خلال دمج الميزات الجزيئية الصريحة للأدوية. تُظهر التقييمات التجريبية المكثفة فعالية وقوة النماذج المقترحة عبر مجموعات بيانات وأهداف بيولوجية متعددة (بروتينات تنتمي إلى عائلتي السيتوكينات و الكينيز المعتمد على السيكلين وتفاعلاتها مع الأدوية)، مما يوفر مرشحين دوائيين قابلين للتطبيق كتوصيات لكل من تسلسلات البروتينات العادية واليتيمة، مع ضمان ملاءمتها حسابيًا وبيولوجيًا.

يقترح البحث الثاني إطارَي توصية وترجمة قائمين على المحولات (ألا وهما RSSSTr وTSSTr) اللذان تم تصميمهما لاستغلال آليات الانتباه الذاتي في نمذجة التبعيات بعيدة المدى وأنماط التفاعل المعقدة. ومن خلال دمج استراتيجية فهرسة البصمات البيولوجية مع بنية التشفير المحسّنة المقترحين من طرفنا، تُمكّن هذه النماذج من التنبؤ الدقيق بالتفاعلات بين الجزيئات (مهمة إكمال المصفوفة)، مع إنجاز مهمة توسيع المصفوفة المقترحة من طرفنا، والتي تخول كلا النموذجين من التعاون فيما بينهما من أجل توفير مرشحات دوائية مُولّدة، وذات صلة بكل من تسلسلات البروتين العادية والمتحولة. وقد تم التحقق من مدى فعالية كلا النهجين المقترحين على أهداف بيولوجية متنوعة، تشمل السيتوكينات، والكينيزات المعتمدة على السيكلين، ومستقبلات البروتين G المقترنة، بالإضافة إلى بروتينات فيروسية تابعة لكل من فيروس نقص المناعة البشرية، وفيروس كورونا، حيث أظهر النموذجان أداءً متميزًا في كلٍ من إعادة استخدام الأدوية، وتصميم مركبات دوائية جديدة.

بشكل عام، تُبيّن هذه الأطروحة أن أنظمة التوصية القائمة على التعلّم العميق تُوفّر حلولًا قابلة للتطوير وذات دلالة بيولوجية للتنبؤ بالتفاعلات واكتشاف الأدوية، مُقدّمةً بديلًا حسابيًا واعدًا للمسارات التجريبية التقليدية.

**الكلمات المفتاحية:** المعلوماتية الحيوية، التعلم العميق، اكتشاف الأدوية، إعادة استخدام الأدوية، أنظمة التوصية.

# Résumé

La croissance rapide des données biologiques, conjuguée aux coûts élevés, aux durée importante, et aux taux d'échec associés aux méthodes traditionnelles de découverte de médicaments, a intensifié le besoin d'approches computationnelles capables de prédire efficacement les interactions intermoléculaires et d'accélérer le développement thérapeutique. La bioinformatique, associée aux techniques d'apprentissage automatique et aux systèmes de recommandation, s'est imposée comme un cadre puissant pour la modélisation de relations biologiques complexes telles que les interactions et associations médicament-cible, protéine-protéine et médicament-maladie. Cependant, ces tâches sont complexifiées par la rareté des données, leur forte dimensionnalité, l'hétérogénéité des entités biologiques et la présence de protéines orphelines ou de médicaments dont les interactions sont peu ou pas connues.

Cette thèse explore l'exploitation des systèmes de recommandation et des modèles d'apprentissage profond pour la prédiction des interactions intermoléculaires, le repositionnement de médicaments et la conception de nouveaux médicaments, en formulant ces problèmes comme des tâches de complétion et d'expansion de matrices. Après avoir présenté un exposé complet des fondements de la bioinformatique, des principes de l'apprentissage automatique et des méthodologies des systèmes de recommandation, ce travail analyse leur applicabilité et leurs limites dans le contexte biomédical.

La première contribution propose un système de recommandation basé sur l'apprentissage profond convolutionnel-récurrent (RSCRDL) qui exploite la synergie convolutionnel-récurrente pour capturer à la fois les motifs structuraux locaux et les dépendances séquentielles inhérentes aux séquences biologiques, afin d'améliorer l'efficacité de la prédiction des interactions médicament-cible. De plus, une variante gloutonne améliorée, qu'est RSCRDL_F, est introduite pour optimiser la représentation des caractéristiques et les performances prédictives grâce à l'intégration de caractéristiques moléculaires explicites des médicaments. De nombreuses évaluations expérimentales démontrent l'efficacité et la robustesse des modèles proposés sur plusieurs jeux de données et cibles biologiques (protéines appartenant aux familles des cytokines et des kinases dépendantes des cyclines et leurs interactions avec les médicaments), fournissant des candidats médicaments viables pour les séquences protéiques régulières et orphelines, tout en assurant leur pertinence computationnelle et biologique.

La seconde contribution présente des modèles de recommandation et de traduction basés sur l'architecture Transformer (RSSSTr et TSSTr), conçus pour exploiter les mécanismes d'auto-attention afin de modéliser les dépendances à longue portée et les interactions complexes. En intégrant notre nouvelle stratégie d'indexation d'empreintes moléculaire à notre architecture d'encodeur améliorée, ces modèles permettent une prédiction précise des interactions intermoléculaires (tâche de complétion de matrice) et la réalisation de notre nouvelle tâche d'expansion de matrice, où les deux approches collaborent pour générer de nouveaux candidats médicaments pertinents, tant pour les séquences protéiques régulières que mutées. Les approches proposées sont validées sur diverses cibles biologiques, notamment les cytokines, les kinases

dépendantes des cyclines, les récepteurs couplés aux protéines G, et les protéines faisant partie des organismes VIH et SARS-CoV-2, démontrant de solides performances tant dans le repositionnement de médicaments que dans la conception de nouveaux médicaments.

En résumé, cette thèse démontre que les systèmes de recommandation basés sur l'apprentissage profond offrent des solutions évolutives, et biologiquement pertinentes pour la prédiction des interactions et la découverte de médicaments, constituant ainsi une alternative computationnelle prometteuse aux approches expérimentales traditionnelles.

**Mots-clés:** Bioinformatique, Apprentissage profond, Découverte de médicaments, Repositionnement de médicaments, Systèmes de recommandation,.

# Table of Contents

Acknowledgments ..... II
Abstract ..... III
الملخص ..... IV
Résumé ..... V
Table of Contents ..... VII
List of Figures ..... XVI
List of Tables ..... XX
General introduction ..... 22
1. Chapter 1: Introduction to the Bioinformatics Field ..... 28
1.1. Introduction ..... 29
1.2. Foundational concepts of bioinformatics ..... 30
1.2.1. Definition of Bioinformatics ..... 30
1.2.2. The history of bioinformatics ..... 30
1.2.3. The goals and objectives of bioinformatics ..... 31
1.3. Basic building blocks of biological systems ..... 32
1.3.1. Atoms ..... 32
1.3.2. Molecules and macromolecules ..... 32
1.3.3. Chemical bonds ..... 33
1.3.4. DNA sequence ..... 33
1.3.5. RNA sequence ..... 35
1.3.6. Biological pathways ..... 35
1.3.7. Protein sequences ..... 35
1.3.7.1. Amino acids ..... 36
1.3.7.2. Central dogma of molecular biology ..... 36
1.3.7.3. Proteins function-based taxonomy ..... 37
1.3.7.4. Proteins structure-based taxonomy ..... 38
1.4. Data infrastructures and analytical frameworks in bioinformatics ..... 40
1.4.1. Databases ..... 40
1.4.1.1. Sequence databases ..... 40
1.4.1.2. Structural databases ..... 40
1.4.1.3. Micro-array databases ..... 40

1.4.1.4. Metabolic and pathway databases ..... 41
1.4.1.5. Chemical databases ..... 41
1.4.1.6. Genomics/Proteomics databases ..... 41
1.4.1.7. Disease databases ..... 41
1.4.1.8. Bibliographic databases ..... 42
1.4.2. Bioinformatics computational tools ..... 44
1.4.2.1. Sequence data analysis tools ..... 44
1.4.2.2. Genomic data analysis tools ..... 44
1.4.2.3. Proteomic data analysis tools ..... 44
1.4.2.4. Pathway and functional annotation analysis tools ..... 44
1.4.2.5. Cheminformatics and drug discovery tools ..... 44
1.4.2.6. Machine learning and data mining tools ..... 45
1.4.2.7. Data visualization tools ..... 45
1.4.2.8. Other tools ..... 45
1.5. Application areas of bioinformatics ..... 46
1.5.1. Genomics ..... 46
1.5.2. Proteomics ..... 47
1.5.3. Transcriptomics ..... 47
1.5.4. Phylogenetics ..... 47
1.5.5. Drug discovery ..... 47
1.5.6. Forensic science ..... 48
1.5.7. Environmental sciences ..... 48
1.5.8. Agricultural sciences ..... 48
1.6. Bioinformatics challenges and limitations ..... 49
1.7. Conclusion ..... 49
2. Chapter 2: Machine Learning and its Core Concepts and Techniques ..... 51
2.1. Introduction ..... 52
2.2. Definition of Machine Learning ..... 52
2.3. Types of machine learning ..... 53
2.3.1. Supervised learning ..... 54
2.3.1.1. Regression ..... 54
2.3.1.2. Classification ..... 55

2.3.2. Unsupervised learning ..... 58
2.3.2.1. Clustering ..... 59
2.3.2.2. Association rules ..... 60
2.3.2.3. Dimensionality reduction ..... 61
2.3.2.4. Generative models ..... 61
2.3.3. Semi-supervised learning ..... 62
2.3.4. Reinforcement learning ..... 62
2.4. Key concepts in machine learning ..... 63
2.4.1. Data ..... 63
2.4.1.1. Quantitative data ..... 63
2.4.1.2. Qualitative data ..... 63
2.4.1.3. Time-series data ..... 64
2.4.1.4. Structured data ..... 64
2.4.1.5. Unstructured data ..... 64
2.4.1.6. Semi-structured ..... 64
2.4.2. Data preprocessing and transformations ..... 64
2.4.2.1. Data cleaning ..... 65
2.4.2.2. Data normalization ..... 65
2.4.2.3. Feature encoding ..... 65
2.4.2.4. Text-related transformations: ..... 65
2.4.2.5. Image-related transformations ..... 66
2.4.2.6. Feature discretization ..... 66
2.4.2.7. Data augmentation ..... 66
2.4.2.8. Data reduction ..... 66
2.4.3. Machine learning pipeline and work flow ..... 66
2.4.3.1. Problem definition ..... 67
2.4.3.2. Data processing ..... 67
2.4.3.3. Model selection ..... 68
2.4.3.4. Hyper parameter tuning ..... 68
2.4.3.5. Model training ..... 68
2.4.3.6. Model evaluation ..... 68
2.4.3.7. Model deployment, monitoring, and maintenance ..... 69

2.5. Common machine learning approaches ..... 69
2.5.1. Supervised machine learning algorithms ..... 69
2.5.1.1. Linear regression ..... 69
2.5.1.2. Multiple regression ..... 69
2.5.1.3. Logistic regression ..... 69
2.5.1.4. Polynomial regression ..... 70
2.5.1.5. Redge regression (Tikhonov regularization) ..... 70
2.5.1.6. Lasso regression (Least Absolute Shrinkage and Selection Operator) ..... 70
2.5.1.7. Support vector machines (SVM) ..... 71
2.5.1.8. Decision trees ..... 72
2.5.1.9. K-Nearest neighbor (k-NN) ..... 73
2.5.1.10. Naïve bayes classifier ..... 73
2.5.2. Unsupervised machine learning algorithms ..... 74
2.5.2.1. Principal component analysis (PCA) ..... 74
2.5.2.2. t-Distributed Stochastic Neighbor Embedding (t-SNE) ..... 74
2.5.2.3. Generative Adversarial Networks (GANs) ..... 75
2.5.2.4. Variational Autoencoders (VAEs) ..... 75
2.5.2.5. K-means ..... 76
2.5.2.6. Apriori algorithm ..... 76
2.5.3. Deep Neural network (DNN) based algorithms: ..... 77
2.5.3.1. Multi-layer perceptron (MLP) ..... 78
2.5.3.2. Convolutional neural network (CNN) ..... 79
2.5.3.3. Recurrent neural network (RNN) ..... 80
2.5.3.4. Gated-recurrent units (GRUs) ..... 81
2.5.3.5. Transformer architecture ..... 82
2.5.3.6. Autoencoders (AE) ..... 83
2.5.4. Ensemble methods ..... 84
2.5.4.1. Random forest (RF) ..... 84
2.5.4.2. Bootstrap aggregating (bagging) ..... 84
2.5.4.3. Boosting ..... 85
2.6. Applications of machine learning ..... 85
2.6.1. Computer vision ..... 85

2.6.2. Natural language processing ..... 86
2.6.3. Cybersecurity ..... 86
2.6.4. Recommendation systems ..... 86
2.6.5. Health care ..... 87
2.6.6. Finance ..... 87
2.6.7. Manufacturing and industry ..... 87
2.6.8. Agriculture ..... 88
2.6.9. Climate science and environmental monitoring ..... 88
2.7. Challenges and limitations of machine learning approaches ..... 88
2.7.1. Computational challenges ..... 88
2.7.1.1. Underfitting (high bias) ..... 88
2.7.1.2. Overfitting (high variance) ..... 89
2.7.1.3. Data acquisition, quality, and availability ..... 89
2.7.1.4. Model complexity and scalability ..... 89
2.7.1.5. The curse of dimensionality ..... 90
2.7.1.6. Interpretability and explainability ..... 90
2.7.1.7. Hardware and computational limits ..... 90
2.7.2. Real-world challenges ..... 91
2.7.2.1. Ethical constraints ..... 91
2.7.2.2. Privacy and security threats ..... 91
2.8. Conclusion ..... 91
3. Chapter 3: Recommender Systems in Bioinformatics ..... 93
3.1. Introduction ..... 94
3.2. Drug discovery in bioinformatics ..... 95
3.2.1. Definition and scope ..... 95
3.2.2. Traditional drug development pipeline ..... 95
3.2.2.1. Target identification and validation ..... 96
3.2.2.2. Lead discovery ..... 96
3.2.2.3. Lead optimization ..... 96
3.2.2.4. Preclinical testing ..... 97
3.2.2.5. Clinical trials ..... 97
3.2.2.6. Regulatory approval ..... 97

3.2.2.7. Post-marketing surveillance ..... 98
3.2.3. Challenges and limitations of traditional drug development ..... 98
3.2.4. Drug development vs drug repurposing ..... 99
3.2.5. Computational ML methods in drug discovery and repurposing strategies ..... 100
3.3. Intermolecular interactions in bioinformatics ..... 101
3.3.1. Definition ..... 101
3.3.2. Types of intermolecular interactions ..... 101
3.3.2.1. Drug-Target Interactions (DTIs) ..... 101
3.3.2.2. Protein-Protein Interactions (PPIs) ..... 102
3.3.2.3. Drug-Drug Interactions (DDIs) ..... 102
3.3.2.4. Protein-Nucleic Acid Interactions ..... 102
3.3.2.5. Drug-Disease associations ..... 103
3.3.3. The importance of computational prediction of intermolecular interactions ..... 103
3.4. Recommender systems: principles and foundations ..... 104
3.4.1. Definition and concept of work ..... 104
3.4.2. Atoms leveraged by recommender systems ..... 105
3.4.2.1. User atom ..... 105
3.4.2.2. Item atom ..... 105
3.4.2.3. Interaction atom: ..... 105
3.4.2.4. Rating matrix atom ..... 106
3.4.3. Aims and goals of recommender systems ..... 106
3.4.3.1. Relevance ..... 106
3.4.3.2. Diversity ..... 106
3.4.3.3. Novelty ..... 107
3.4.3.4. Serendipity ..... 107
3.4.4. Categories of recommender systems ..... 107
3.4.4.1. Content-based (CB) ..... 108
3.4.4.2. Collaborative filtering (CF) ..... 108
3.4.4.3. Knowledge-based ..... 109
3.4.4.4. Hybrid recommenders ..... 109
3.5. The exploitation of RS in bioinformatics ..... 110
3.5.1. Data representations exploitable via bioinformatics-oriented RS ..... 110

3.5.1.1. Proteins ..... 110
3.5.1.2. Drugs ..... 111
3.5.1.3. Genes ..... 112
3.5.1.4. Heterogeneous interaction networks ..... 113
3.5.2. Mapping bioinformatics problems as recommendation tasks ..... 113
3.5.2.1. Defining entities ..... 113
3.5.2.2. Representing biological data ..... 113
3.5.2.3. Constructing interaction matrix or graph ..... 113
3.5.2.4. Applying an appropriate RS ..... 114
3.5.2.5. Validating results ..... 114
3.5.3. The applications of RS in bioinformatics ..... 114
3.5.3.1. DTIs and PPIs in selico ..... 114
3.5.3.2. Disease-gene association predictions ..... 115
3.5.3.3. Drug repurposing ..... 115
3.5.3.4. Personalized therapeutic strategies ..... 115
3.5.3.5. Pathway and functional annotation prediction ..... 116
3.5.3.6. Metagenomics ..... 116
3.5.3.7. Scientific literature recommendation ..... 116
3.6. Summary of the most important contributions of RS approaches in bioinformatics across intermolecular interaction prediction tasks ..... 117
3.6.1. Matrix factorization-based techniques ..... 117
3.6.2. Deep learning-based models ..... 120
3.6.3. Graph-based recommenders ..... 125
3.6.4. Kernel-based estimators ..... 130
3.6.5. Other methods ..... 132
3.7. Conclusion ..... 135
4. Chapter 4: Proposed Recommender Systems for Intermolecular Interaction Prediction, Drug Repurposing, and de novo Drug Design ..... 136
4.1. Introduction ..... 137
4.2. Problem definition and research context ..... 138
4.2.1. Problem formulation ..... 138
4.2.2. The impact of the of the selected biological targets ..... 139
4.2.2.1. Cytokines ..... 139

4.2.2.2. Cyclin-Dependent Kinases (CDKs) ..... 140
4.2.2.3. G-Protein-Coupled Receptors (GPCRs) ..... 140
4.2.2.4. Human Immunodeficiency Virus (HIV) ..... 140
4.2.2.5. Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) ..... 140
4.2.3. Understanding the selected ML paradigms ..... 141
4.2.3.1. CNN-RNN synergy ..... 141
4.2.3.2. Transformer architectures ..... 141
4.2.4. Overview of the chosen criteria and hyperparameters ..... 142
4.2.4.1. Training criteria and hyperparameters ..... 142
4.2.4.2. Evaluation criteria ..... 148
4.2.4.3. Biological criteria ..... 152
4.3. First proposition: Convolutional and recurrent deep learning-based approaches for DTI prediction as matrix completion task ..... 154
4.3.1. Recommender System based on Convolution Recurrent Deep Learning (RSCRDL) ..... 154
4.3.2. The RSCRDL_F variant ..... 158
4.3.3. Evaluation and analysis ..... 158
4.3.3.1. Datasets ..... 158
4.3.3.2. Experimental setup ..... 158
4.3.3.3. Training process ..... 159
4.3.3.4. Performance assessment ..... 159
4.3.3.5. Performance validation ..... 161
4.3.3.6. The application of RSCRDL and RSCRDL_F in drug repurposing strategies 172
4.4. Second proposition: Transformer-based recommender system and translator approaches for intermolecular interaction prediction and drug generation as matrix completion and matrix expansion tasks ..... 178
4.4.1. FP_indexation method ..... 179
4.4.2. The architecture of RSSSTr ..... 180
4.4.2.1. Preprocessing phase ..... 181
4.4.2.2. Encoding phase ..... 181
4.4.2.3. Decoding phase ..... 182
4.4.2.4. Projection layer ..... 182
4.4.3. The architecture of TSSTr ..... 182

4.4.3.1. Encoding phase ..... 183
4.4.3.2. Decoding phase ..... 183
4.4.3.3. Projection layer ..... 184
4.4.4. Evaluation and analysis ..... 184
4.4.4.1. Datasets ..... 184
4.4.4.2. Experimental setup ..... 184
4.4.4.3. Training process ..... 185
4.4.4.4. Performance assessment ..... 186
4.4.4.5. The application of RSSSTr and TSSTr in drug repurposing and *de novo* drug design strategies involving both regular and orphan protein sequences ..... 190
4.5. Conclusion ..... 201
Conclusion and Future Work ..... 202
References ..... 205
Scientific Contributions ..... 236

# List of Figures

Figure 1.1 The interdisciplinary fields bioinformatics depend on ..... 31
Figure 1.2 The structure of an atom and its components ..... 32
Figure 1.3 The four categories of macromolecules ..... 33
Figure 1.4 Chemical bonds ..... 34
Figure 1.5 DNA structure ..... 34
Figure 1.6 RNA transcription ..... 35
Figure 1.7 Immuune response pathway ..... 36
Figure 1.8 The molecular structure of an amino acid ..... 36
Figure 1.9 The flow of genetic information from DNA to RNA to protein ..... 37
Figure 1.10 Protein structures ..... 40
Figure 2.1 Supervised learning paradigms ..... 54
Figure 2.2 Types of supervised classification tasks ..... 56
Figure 2.3 Multi-class decomposition strategies adopted by binary classifiers ..... 57
Figure 2.4 Heirachical classification ..... 58
Figure 2.5 Types of unsupervised machine learning ..... 59
Figure 2.6 Generative models ..... 62
Figure 2.7 Basic data types ..... 63
Figure 2.8 Machine learning pipeline ..... 67
Figure 2.9 Types of regression in machine learning ..... 70
Figure 2.10 The application of SVMs on a classification task using different types of kernels .. 71
Figure 2.11 Decision tree model architecture ..... 72
Figure 2.12 K-NN approach in assigning labels to new instances ..... 73
Figure 2.13 The classification process of a Gaussian-based naïve bayes classifier ..... 74
Figure 2.14 the application of PCA on a multivariate Gaussian distribution ..... 75
Figure 2.15 The application of t-SNE on a S-curve and swiss-roll shaped data distributions ..... 75
Figure 2.16 The architecture of a GAN ..... 76
Figure 2.17 The architecture of a convolutional VEA ..... 76
Figure 2.18 The convergence process of K-means towards optimal clusters ..... 77
Figure 2.19 The process of extracting frequent items via apriori algorithm ..... 77
Figure 2.20 The architecture of a neuron ..... 78

Figure 2.21 The architecture of a MLP ..... 79
Figure 2.22 The architecture of a CNN ..... 79
Figure 2.23 The mathematical application of the convolutional operation ..... 80
Figure 2.24 The architecture of a recurrent neural network ..... 81
Figure 2.25 The architrecture of GRU ..... 82
Figure 2.26 The architecture of a transfomer and its components ..... 83
Figure 2.27 The architecture of an autoencoder ..... 84
Figure 2.28 Random forest classifier ..... 85
Figure 2.29 Understanding underfitting and overfitting ..... 89
Figure 3.1 Schematic representation of the drug discovery process ..... 98
Figure 3.2 The main differences between drug discovery and drug repurposing ..... 100
Figure 3.3 Enzym-substrare interaction ..... 103
Figure 3.4 The concept of work of recommender systems ..... 104
Figure 3.5 Types of recommender systems ..... 107
Figure 3.6 Collaborative filtering technique in recommender systems ..... 108
Figure 4.1 Downsamplng operation of an image channel via the maxpooling layer ..... 143
Figure 4.2 ReLU ctivation function curve ..... 144
Figure 4.3 Sigmoid activation funtion curve ..... 145
Figure 4.4 The architecture of RSCRDL ..... 155
Figure 4.5 RSCRDL encoder's architecture ..... 157
Figure 4.6 RSCRDL decoder's architecture ..... 158
Figure 4.7 BCE loss values w.r.t number of epochs for the training and validation sets of the cytokine and CDK datasets ..... 160
Figure 4.8 General metrics comparison w.r.t number of epochs for the cytokine and CDK training sets ..... 161
Figure 4.9 General metrics comparison w.r.t number of epochs for cytokine and CDK validation sets ..... 161
Figure 4.10 Cohen's kappa score comparison w.r.t number of epochs for cytokine and CDK validation sets ..... 161
Figure 4.11 Drugs’ fingerprint similarity graph of the cytokine set ..... 164

Figure 4.12 Cytokine set's drugs projections via t-SNE algorithm with correlation distance-metric ..... 165
Figure 4.13 Cytokine set's drugs projections via t-SNE algorithm with Chebyshev distance-metric ..... 165
Figure 4.14 Drugs' fingerprint similarity graph of the CDK set ..... 166
Figure 4.15 CDK set's drugs projections via t-SNE algorithm with correlation distance-metric ..... 167
Figure 4.16 CDK set's drugs projections via t-SNE algorithm with Chebyshev distance-metrics ..... 167
Figure 4.17 Cytokine set's protein sequences similarity graph ..... 169
Figure 4.18 Cytokine set's protein sequences projections via t-SNE algorithm with correlation distance-metric ..... 170
Figure 4.19 Cytokine set's protein sequences projections via t-SNE algorithm with Chebyshev distance-metric ..... 170
Figure 4.20 CDK set's protein sequences similarity graph ..... 171
Figure 4.21 CDK set's protein sequences projections via t-SNE algorithm with correlation distance-metric ..... 171
Figure 4.22 CDK set's protein sequences projections via t-SNE algorithm with Chebyshev distance-metric ..... 171
Figure 4.23 Candidate interactions deduction process ..... 173
Figure 4.24 Difference between tokenizing a binary FP vector via (a) a regular tokenization method, and (b) via our FP indexation method ..... 179
Figure 4.25 The architecture of RSSSTr ..... 182
Figure 4.26 The architecture of TSSTr ..... 183
Figure 4.27 The BCE loss values of RS w.r.t the number of epochs across all four training sets (lower is better) ..... 186
Figure 4.28 The BCE loss values of RS w.r.t the number of epochs across all four validation sets (lower is better) ..... 186
Figure 4.29 Metrics' results w.r.t the number of epochs for all five RS across all four training sets (higher is better) ..... 187

Figure 4.30 Metrics' results w.r.t the number of epochs for all five RS across all four validation sets (higher is better) ..... 188
Figure 4.31 Cohen kappa score results w.r.t the number of epochs for all five RS across all validation sets (higher is better) ..... 189
Figure 4.32 CE loss values of translator models w.r.t the number of epochs regarding all eight translation tasks across all training and validation sets ..... 190
Figure 4.33 Meteor score values of translator models w.r.t the number of epochs regarding all eight translation tasks across all four validation sets ..... 190
Figure 4.34 Matrix expansion task work flow ..... 195

# List of Tables

Table 1.1 Databases used in bioinformatics ..... 42
Table 1.2 Most well-known bioinfomratics tools ..... 45
Table 2.1 Main differences between traditional programming and machine learning paradigms 53
Table 4.1 Characteristics of the deployed datasets ..... 159
Table 4.2 Metrics comparison regarding all models for both test sets ..... 162
Table 4.3 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the correlation distance-metric for the cytokine set ..... 164
Table 4.4 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the Chebyshev distance-metric for the cytokine set ..... 164
Table 4.5 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the correlation distance-metric for the CDK set ..... 166
Table 4.6 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the Chebyshev distance-metric for the CDK set ..... 166
Table 4.7 Protein inter/intra-cluster results of all 400 t-SNE models based on the correlation distance-metric for the cytokine set ..... 169
Table 4.8 Protein inter/intra-cluster results of all 400 t-SNE models based on the Chebyshev distance-metric for the cytokine set ..... 169
Table 4.9 Protein inter/intra-cluster results of all 400 t-SNE models based on the correlation distance-metric for the CDK set ..... 170
Table 4.10 Protein inter/intra-cluster results of all 400 t-SNE models based on the Chebyshev distance-metric for the CDK set ..... 171
Table 4.11 Top-5 drug-based candidate positive interactions for the protein sequences of the cytokine set ..... 173
Table 4.12 Top-5 drug-based candidate positive interactions for the protein sequences of the CDK set ..... 175
Table 4.13 Top-5 candidate positive interactions regarding Q9NPB9 and drugs with no rule-of-5 violations ..... 176
Table 4.14 Top-5 candidate positive interactions regarding F5H6Z0 and drugs with no rule-of-5 violations ..... 176

Table 4.15 Top-5 candidate positive interactions regarding Q9H8S5 and drugs with no rule-of-5 violations ..... 177
Table 4.16 Top-5 candidate positive interactions regarding Q14094 and drugs with no rule-of-5 violations ..... 178
Table 4.17 Demonstration of the tokenization process of a 3-character-long binary vector via our FP_indexation method ..... 180
Table 4.18 Characteristics of the deployed datasets ..... 185
Table 4.19 Metrics results' regarding all five RS concerning the two target-drug test sets ..... 188
Table 4.20 Metrics results' regarding all five RS concerning the two target-target test sets ..... 189
Table 4.21 Top 5 most confident protein-based candidate positive interactions involve drugs with no rule-of-5 violations regarding both target-drug sets ..... 192
Table 4.22 Top 5 most confident drug-based candidate positive interactions involving drugs with no rule-of-5 violations, deduced based on neighboring drugs with no rule-of-5 violations regarding CDK set ..... 194
Table 4.23 Top 5 most confident virus-protein-based candidate positive interactions across both target-target sets ..... 194
Table 4.24 Statistics of the newly generated valid drugs across both target-drug sets regarding five molecular descriptors listed in Eq (4.39), with values in parentheses referencing the statistics of FDA approved NCEs from 1998 to 2017 ..... 197
Table 4.25 Top 5 most confident drug-based candidate positive interactions that involve generated drugs (with no violations to the 8 molecular descriptors listed in Eq (4.39)) and non-mutated protein sequences across both target-drug sets ..... 199
Table 4.26 Top 5 protein-based candidate positive interactions that involve mutated protein sequences and existing drugs with no rule-of-5 violations regarding both target-drug sets ..... 200

# General introduction

## 1. Research context

The bioinformatics field is currently experiencing an unprecedented data overflow that is driven by high-throughput sequencing technologies, large-scale omics experiments, automated chemical screening, and the continuous curation of heterogeneous biological databases. While this data abundance holds immense potential for accelerating biomedical discoveries; it has simultaneously imposed severe constraints upon traditional drug discovery pipelines, which were originally designed for low-throughput, hypothesis-driven experimentation. In particular, the prediction of intermolecular interactions (e.g., drug-target and protein-protein interactions) has become increasingly complex due to the scale, heterogeneity, sparsity, and noise characterizing modern biological datasets, making exhaustive experimental validation impractical in terms of time, cost, and feasibility.

To overcome these limitations, researchers has progressively shifted toward computational and data-driven approaches, with machine learning emerging as a cornerstone technology for extracting knowledge from large-scale bioinformatics data. By leveraging statistical learning, and deep neural architectures; machine learning models have enabled the automated discovery of patterns that are difficult to capture through analytical or rule-based methods, thereby substantially accelerating tasks such as interaction prediction, virtual screening, drug repurposing, and *de novo* drug design. This paradigm shift has transformed drug discovery from a largely sequential and experimental process into a computationally assisted pipeline, where *in silico* predictions guide experimental efforts and reduce the search space of plausible biological hypotheses.

Despite these advances, most machine learning approaches deployed for intermolecular interaction prediction still face fundamental challenges when operating in real-world biomedical settings. As many models are heavily dependent on dense interaction matrices and well-annotated datasets, which limits their ability to generalize to orphan and mutated proteins, emerging pathogens, and newly synthesized drugs, for which little or no interaction data is available. In this constrained environment, recommender systems have emerged as one of the few viable computational paradigms, as they are inherently designed to handle sparsity, missing values, and cold-start scenarios while framing interaction prediction as a recommendation problem rather than a conventional classification task.

Unfortunately, the general-purpose nature of most recommender systems remain insufficient for addressing the specific complexities of intermolecular interaction prediction problems. These approaches often rely on handcrafted descriptors or shallow, precomputed features that fail to fully exploit the rich and multi-modal biological spaces characterizing bioinformatics-related entities, such as protein sequences, molecular structures, and chemical fingerprints. This limitation motivates the development of specialized, bioinformatics-oriented recommender systems that are capable of directly processing raw biological sequences and structural representations, integrating multiple sources of information, and learning task-specific latent spaces that are better aligned with the underlying biochemical mechanisms governing intermolecular interactions.

In parallel, current chemical libraries mainly used in drug repurposing strategies remain inherently limited in size, diversity, and chemical novelty, thereby restricting the process of

discovering innovative therapeutics. At the same time, generative models have demonstrated remarkable success across diverse domains by learning complex data distributions in order to enable the synthesis of novel, high-quality samples that satisfy domain-specific constraints. These achievements motivate a strategic shift toward *de novo* drug design, where generative and recommendation-based frameworks work jointly to enable the exploration of vast, previously inaccessible chemical spaces, providing the computational capabilities to potentially surpass traditional drug repurposing and design strategies.

In summary, this thesis addresses the problematic arising from the combined challenges of data sparsity, biological complexity and heterogeneity, and limited generalization in existing computational approaches for intermolecular interaction prediction tasks. Motivated by the necessity to develop sequence-aware, deep learning-based recommender systems that are capable of handling heterogeneous biological data, supporting orphan and mutated proteins, and enabling both predictive and generative drug discovery strategies, while aiming to provide scalable, interpretable, and biologically grounded solutions to pressing challenges in modern drug discovery and molecular interaction analysis.

## 2. Main contributions

As stated earlier, this thesis aims to investigate the problem of predicting intermolecular interactions via sequence-processing deep learning-based recommender systems. To this end, we first aimed to collect curated, task-specific datasets, derived from large-scale bioinformatics resources, which can be projected as partially-observed interaction matrices, where biological entities are modeled as users and items that interact with each other, enabling the exploitation of biological sequences, while preventing the use of handcrafted global descriptors that may limit model generalization and applicability across unseen targets.

On one side, a detailed manual analysis of the ChEMBL relational schema enabled us to construct custom SQL queries that yielded three high-quality drug-target-oriented datasets. These datasets revolve around Cytokines, Cyclin-dependent Kinases (CDKs), and G-protein-coupled receptors (GPCRs), and their corresponding validated interactions with molecular compounds, that are assessed via criteria such as $IC50$, and inhibition measures. In parallel, an in-depth analysis of the IntAct database led to development of specialized IntAct-related queries, which resulted in the construction of two protein-protein interaction-oriented datasets. These datasets focus on proteins belonging to either the Human Immunodeficiency Virus (HIV), or Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) organisms and their various interactions with human protein sequences, which are curated based on associated MI-score values. Collectively, these datasets allow for a robust evaluation of our proposed frameworks across diverse biological targets, interaction types and scenarios.

To carry out the task of intermolecular interaction prediction; we proposed a convolutional-recurrent deep learning-based recommender system named RSCRDL, which exploits natural language processing techniques to jointly process protein and drug sequences. Additionally, given that intermolecular interaction-oriented datasets are highly sparse; we extended our base architecture to RSCRDL_F, incorporating additional drug-related molecular features to further

improve predictive performance and robustness. Experimental evaluations conducted on our curated Cytokine and CDK datasets demonstrated the significant outperformance accomplished by both our approaches over two general-purpose state-of-the-art recommender systems. Furthermore, our proposed frameworks provide the ability to process orphan protein sequences, allowing the inference of plausible drug candidates for previously unprocessed entities, highlighting the potential of our approaches for drug repurposing and early-stage target exploration. This study was published on 01 July 2024 in the "Expert Systems with Applications" journal.

Furthermore, to further extend the task of intermolecular interaction prediction toward suggesting newly generated molecular compounds; we proposed sequential and structural feature-processing transformer-based recommender system framework (RSSSTr) and its translator counterpart (TSSTr). RSSSTr leverages its transformer architecture in order to process the sequential and structural representations of biological entities in order to suggest relevant drugs (both seen and novel candidates) for protein sequences, including previously unobserved and mutated sequences. In parallel, TSSTr leverages the same aspects of biological entities via its transformer architecture in order to generate novel drug candidates through processing proteins and their mutated variants in an inter-molecular translation fashion. Additionally, to efficiently process and extract latent features from the structural representations of molecules; both our RSSSTr and TSSTr adopt our newly propose FP_indexation strategy. Moreover, to address the training instability commonly associated with highly-complex transformer models, we proposed an enhanced encoder design that incorporates additional residual connections that help improve gradient flow during backpropagation. Finally, we introduced an original "matrix expansion" task, which combines TSSTr for novel drug generation, and RSSSTr for relevance validation, resulting in the recommendation of newly generated drug candidates that satisfy eight rule-of-five molecular constraints, demonstrating the efficiency of the contribution of our proposed frameworks as an end-to-end novel-relevant-drug recommending strategy. Experimental evaluations conducted on our curated CDK, GPCR, HIV, and SARS-CoV-2 datasets demonstrated the significant outperformance achieved by RSSSTr and TSSTr over several state-of-the-art approaches.

Collectively, these contributions bridge recommender systems, generative models, deep learning, and bioinformatics, demonstrating that sequence and structure-aware approaches can effectively address intermolecular interaction prediction, drug repurposing, and *de novo* drug design, particularly in complex scenarios that involve orphan targets, mutated proteins, and limited experimental knowledge.

## 3. Thesis outline

This dissertation is organized into four main chapters, each addressing a specific aspect of machine learning and recommender system-driven bioinformatics tasks, with a particular focus on intermolecular interaction prediction and drug discovery. The structure of the thesis is as follows.

Chapter 1 provides a comprehensive introduction to the bioinformatics field, establishing the biological and computational foundations required for the remainder of the thesis. It begins by defining bioinformatics and tracing its historical evolution, and objectives. Then, it introduces the

fundamental building blocks of biological systems, ranging from atoms and chemical bonds to macromolecules, nucleic acids, biological pathway, and proteins. Furthermore, it surveys the data infrastructures that underpin modern bioinformatics, including diverse biological databases and computational tools for sequence analysis, genomics, proteomics, cheminformatics, and data visualization. The chapter concludes by reviewing the main application domains of bioinformatics, and discussing the key challenges and limitations that motivate the adoption of advanced machine learning and data-driven approaches.

Chapter 2 presents a systematic overview of machine learning concepts and methodologies that form the methodological backbone of this dissertation. It introduces machine learning definition, paradigms, and learning settings, including supervised, unsupervised, semi-supervised, and reinforcement learning, alongside their associated tasks such as classification, regression, clustering, and generative modeling. The chapter further details essential concepts related to data types, preprocessing techniques, feature engineering, and the standard machine learning pipeline. A broad range of classical, deep learning-based, and ensemble algorithms are then reviewed, with particular attention given to neural architectures such as CNNs, RNNs, and transformers. Finally, the chapter surveys major application domains of machine learning and discusses practical challenges, including computational constraints, model interpretability, ethical considerations, and privacy and security threats, thereby contextualizing the methodological choices adopted in later chapters.

Chapter 3 explores the role of recommender systems in bioinformatics, with a particular focus on intermolecular interaction prediction, drug discovery, and drug repurposing. It first reviews the traditional drug development pipeline and its inherent limitations, motivating the need for computational and machine learning-driven alternatives. The chapter then introduces the concept and types of intermolecular interactions, such as drug-target, protein-protein, and drug-disease associations, emphasizing the importance of their accurate computational prediction. Subsequently, the fundamental principles of recommender systems are presented, including their objectives, taxonomy, and core components. The chapter then demonstrates how bioinformatics problems can be reformulated as recommendation tasks through appropriate data representation and interaction modeling, and reviews existing machine learning, deep learning, graph-based, and kernel-based approaches, highlighting their contributions and limitations in biomedical contexts.

Chapter 4 presents the main methodological contributions of this thesis, proposing novel deep learning-based recommender systems for intermolecular interaction prediction, drug repurposing, and *de novo* drug design. It begins by defining the research problem and biological targets of interest, including cytokines, CDKs, GPCRs, HIV, and SARS-CoV-2, and justifying the selection of convolutional-recurrent and transformer-based paradigms. The chapter introduces a convolutional-recurrent recommender system (RSCRDL) and its enhanced variant (RSCRDL_F) to address drug-target interaction prediction as a matrix completion task, followed by extensive experimental evaluation and biological validation. It further proposes transformer-based recommender and translator frameworks (RSSSTr and TSSTr), that are supported by a newly proposed fingerprint indexation strategy and enhanced encode architecture, to jointly address matrix completion task, and a newly proposed challenge that is matrix expansion task. Through

comprehensive experiments, this chapter demonstrates the effectiveness of the proposed models in both drug repurposing and *de novo* drug design scenarios, including challenging cases involving orphan and mutated protein sequences.

# Chapter 1: Introduction to the Bioinformatics Field

Bioinformatics is an interdisciplinary field that integrates computational techniques, statistical methods, and biological knowledge to extract meaningful insights from vast and complex biological data. By enabling the efficient storage, analysis, and interpretation of genomic, proteomic, and molecular datasets, bioinformatics has become essential to modern life sciences, accelerating discoveries in areas such as disease mechanism elucidation, drug development and repurposing, and personalized medicine.

This chapter provides a foundational introduction to the field of bioinformatics. It is organized as follows: Section 2 presents the definition, historical evolution, and major objectives of bioinformatics. Section 3 outlines the core biological components of the discipline. Section 4 further elucidates the structural and computational methods used to store and analyze studied biological data. Section 5 highlights key application areas of bioinformatics across life sciences and other fields. Finishing by section 6 where we discusses the primary challenges and limitations that continue to shape the development and future directions of bioinformatics.

## 1.1. Introduction

The rapid progression of molecular biology, genomics, and computational technologies has led to an unprecedented explosion of biological data, ranging from DNA sequences and protein structures to large-scale transcriptomic and interaction datasets. This massive growth has introduced two major challenges: how to efficiently store these diverse data types, and how to analyze and interpret them in ways that generate meaningful biological insight. As traditional experimental approaches became insufficient at keeping pace with this data overflow, the scientific community recognized the need for advanced computational solutions capable of managing, integrating, and deciphering complex biological information. This need ultimately gave rise to bioinformatics, an interdisciplinary field that merges biology with computer science, mathematics, and statistics to uncover the principles governing life at the molecular and biological systems levels. Initially focused on fundamental tasks such as sequence alignment and database searching, bioinformatics has rapidly evolved to address multifaceted analyses, including genome annotation, gene expression profiling, structural modeling, molecular interaction prediction, and large-scale data integration. Today, it supports nearly every domain of the life sciences, with applications spanning medicine, pharmacology, agriculture, microbial research, biotechnology, and personalized therapeutics. As biological datasets become increasingly vast and heterogeneous, bioinformatics continues to expand in scope and sophistication, developing new tools and computational paradigms to transform raw data into exploitable scientific knowledge.

In this chapter, we provide an overview of the foundational concepts, data types, and methodological pillars that define modern bioinformatics. We introduce major biological databases and essential analysis tools widely used by researchers, and we highlight prominent applications across life sciences and biotechnology. Finally, we discuss the principal challenges and limitations that shape current bioinformatics research and the future directions that will continue to drive this dynamic field forward.

## 1.2. Foundational concepts of bioinformatics

### 1.2.1. Definition of Bioinformatics

Originally known as *"the study of information processing in biological systems"* [1]; Bioinformatics is an interdisciplinary field that engulfs and leverages the computational tools and techniques derived from the fields of biology, mathematics, computer science, chemistry, physics, and statistics (Figure 1.1), for the purpose of acquiring, accumulating, organizing, analyzing, interpreting, unraveling, and visualizing biological data, phenomena, and systems [2].

### 1.2.2. The history of bioinformatics

Despite the fact that the term "Bioinformatics" itself wasn't coined until 1970 [1]; the origins of this field date back to the mid-20$^{th}$ century. This era was deeply marked by the groundbreaking advancements in both biology and computer science (the two fields most intertwined with bioinformatics), especially the work of Hershey et al. [4], whom challenged the prevailing belief claiming that proteins carry genetic information by providing compelling evidence stating that DNA fulfills such role. This statement was further reinforced by the efforts of Watson et al. [5], who managed to elucidate the molecular structure of DNA, further advancing the overall understanding of its biological significance. Such breakthroughs not only revolutionized the molecular biology field, but also inspired numerous researchers to further explore said discipline, leading to the generation of unprecedented volumes of data that required advanced storage and analysis aptitudes [6].

This challenge necessitated leveraging the computational capabilities of computers, which was only conquered through the pioneering efforts of Margaret Dayhoff [7] and the pivotal role she played in establishing the foundations of computational biology, through introducing the fundamental bioinformatics tools that facilitated the creation of protein sequence databases and analysis algorithms, enabling an efficient and systematic storage, retrieval, and analysis of molecular data. The available machines at that time however lacked the potentials required to perform the proposed calculations efficiently, thus posing several limitations that affected the processing of biological data through computers for several years, until Richard Bellman's work managed to lay the fundamental principles of dynamic programming [8], which were later conceptualized as sequence analysis algorithms [9, 10]. Consequently, altering the evolutionary trajectory pursued by computers in both hardware and software, while encompassing numerous advancements such as the development of enhanced computer architectures through the introduction of supercomputers [11], the emergence of various parallel and distributed programing paradigms [12], which were adopted by contemporary programing languages [13] in order to optimize the utilization of the newly introduced database systems [14], and the rise of artificial intelligence (AI) and machine learning (ML) techniques [15, 16], that coincided with the advent of the internet, which significantly improved communication and facilitated efficient data transmission [17].

The last major milestone that aided in the emergence of bioinformatics as a discipline was the successful sequencing and mapping of the human genome accomplished through the intensive efforts of the "Human Genome Project" [18], further expending humanity's understanding of

genetic mechanisms, hereditary diseases, and precision medicine, while paving the way for revolutionary advancements in biotechnology and personalized healthcare.

The accumulation of such criteria altered the course of what was once a mere biological analysis tool into a fully developed scientific discipline that embraces a diverse range of approaches, algorithms, and techniques designed to challenge all sorts of complex problems associated with biological data.

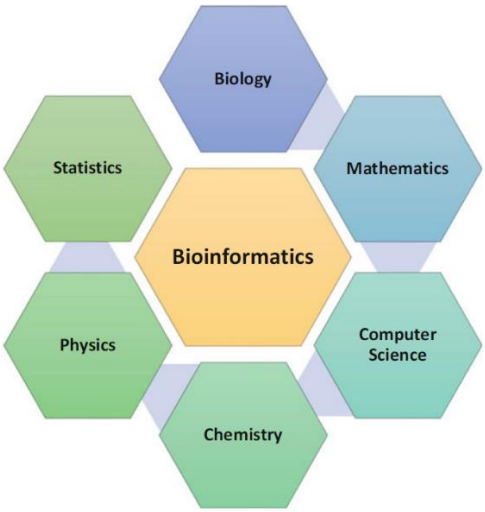



*Figure 1.1 The interdisciplinary fields bioinformatics depend on [3]*

### 1.2.3. The goals and objectives of bioinformatics

As an interdisciplinary domain that encompasses numerous fields; bioinformatics utilizes its various inherited approaches and methods for the purpose of computationally addressing biological problems, with its goals being summarized as [19-21]:

- ✓ **Collection and storage of biological data:** This process is achieved through the development of sophisticated tools and methods that are capable of storing, managing, and retrieving large-scale biological data from a multitude of databases, while ensuring constant accessibility and credibility.
- ✓ **Develop potent computational tools:** These specialized tools are dedicated for the purpose of analyzing, comparing, and interpreting both genomic and proteomic data. On the genomic level; such tools enable the identification of genetic variants and the annotation of genes' functions (advancing the understanding of hereditary diseases, and precision medicine and therapeutic development). On the proteomic level; they facilitate processing protein expression levels, post-translational modifications, and protein-protein interactions (aiding in the identification of disease biomarkers, disease mechanisms, and developing personalized therapeutic strategies.
- ✓ **Drug discovery and therapeutic development:** Bioinformatics field plays a crucial role in identifying potential drug targets, predicting the results of their interactions, and optimizing drug effectiveness and safety via leveraging computational approaches, which are also be utilized for the purpose of designing novel therapeutic strategies, further advancing precision medicine.

- ✓ **Eliminate unethical experimenting practices:** This goal is achievable through leveraging sophisticated computational models and simulation techniques in order to predict numerous biological outcomes, including disease mechanisms, molecular pathways, and host-pathogen interactions, thereby reducing the need for ethically challenging experiments, while enhancing research accuracy and efficiency.
- ✓ **Data expansion:** With the advancements occurring in the ML field and its tools and methods; bioinformatics databases can be enriched via the generation, curation, and estimation of various biological phenomena, encompassing protein structures, gene annotation, and intermolecular interaction prediction, thereby accelerating discoveries across relevant fields.

## 1.3. Basic building blocks of biological systems

The discipline of bioinformatics is structured around three core elements; the biological data requiring study and analysis, the databases responsible for storing and managing such data, and the computational tools and methodologies necessary for its processing and interpretation. To comprehensively grasp these approaches however, a foundational understanding of the underlying biological concepts is imperative.

### 1.3.1. Atoms

Atoms are the smallest component that can form a chemical element and retain its properties (with a total of 118 chemical elements being known). An atom consists of a nucleus that is surrounded by a cloud of the negatively charged subatomic particles known as electrons (Figure 1.2), with the nucleus itself being composed of two other types of subatomic particles which are protons and neutrons, with protons being positively charged while neutrons being neutral in charge [23].

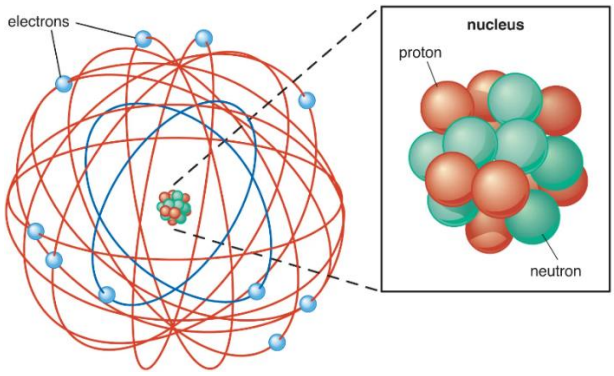



*Figure 1.2 The structure of an atom and its components [22]*

### 1.3.2. Molecules and macromolecules

A molecule is an electrically neutral group of two or more atoms that are chemically bond together via attraction forces [23], with molecules that are composed of the atoms of only one element being called homonuclear, whereas those that are composed of the atoms of more than one element are called heteronuclear [24]. Macromolecules on the other hand are a larger and a more complex counterpart of molecules, mainly characterized by their high molecular weights and

intricate 3D structures, which are typically constructed through the polymerization of smaller subunits called monomers (i.e., molecules that can link with each other forming a chain-like structure) through covalent bonds (Figure 1.3). Macromolecules are considered essential for organic life, and can be categorized into four main classes: proteins that are composed of amino acids, nucleic acids (DNA and RNA) that are made of nucleotides, carbohydrates built from monosaccharides, and lipids that are formed from fatty acids and glycerol. Macromolecules play vital roles in biological systems, ranging from structural support, enzymatic functions, genetic information storage, energy storage and transfer, cell signaling, to the intervention in divers immune responses [25].

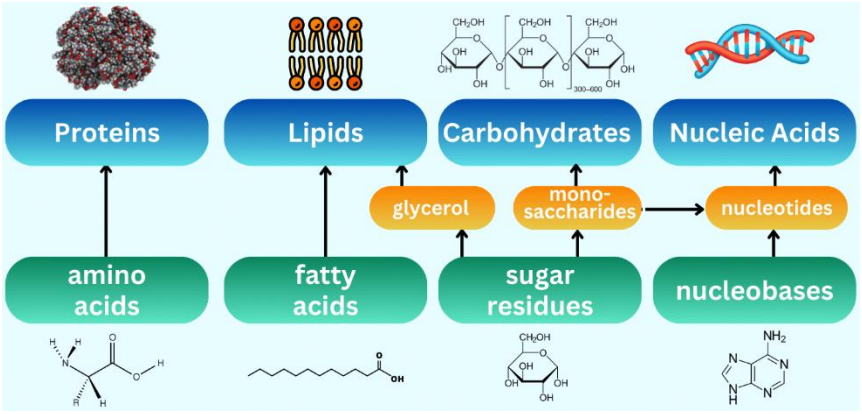



*Figure 1.3 The four categories of macromolecules [26]*

### 1.3.3. Chemical bonds

Chemical bonds are the forces that hold atoms or molecules together via inter-electron interactions in order to form more stable molecular entities (Figure 1.4), collectively determining their structure, properties, and reactivity, they primarily include:

- ✓ **Ionic bonds:** Usually observed between metals and nonmetals; ionic bonding form due to the electrostatic attraction occurring between the oppositely charged ions or atoms with significant differences in electronegativity (e.g., NaCl) [27].
- ✓ **Covalent bonds:** Primarily involving nonmetals; covalent bonds are formed by sharing covalent electron pairs among atoms or ions that possess comparable electronegativity values (e.g., $H_2O$, $O_2$) [28].
- ✓ **Metallic bonds:** They occur among metal atoms, where electrons are delocalized and shared among a lattice of positively charged ions, providing conductivity, malleability, and ductility (e.g., Cu, Fe) [29].
- ✓ **Hydrogen bonds:** Unlike the aforementioned bonds; hydrogen bonds involve partial electrostatic interactions, and only occur between a molecule's hydrogen atom and an electronegative atom (nonmetal) from a different molecule or a different region of the same molecule [27].

### 1.3.4. DNA sequence

Primarily recognized for its cardinal role as a carrier of genetic information (through encoding the instructions necessary for the development, functioning, growth, and reproduction of most

organisms); DesoxyriboNucleic Acid (DNA) is a polymer (i.e., a group of chemically bonded large molecules known as monomers) that consists of two strands that coil in an antiparallel manner forming a double helix structure (Figure 1.5). Each DNA strand is composed of stacked subunits called nucleotides, which consist of a phosphate group, a deoxyribose sugar, and one of four nitrogenous bases: adenine (A), thymine (T), cytosine (C), or guanine (G). These nucleotides are linked together via covalent bonds between the phosphate group of one nucleotide and the deoxyribose sugar of another, forming the backbone of each strand, whereas the overall double-helix structure of the DNA is stabilized by the hydrogen bonds formed between the complementary base pairs (A-T and C-G) on the opposing strands [31] [25].

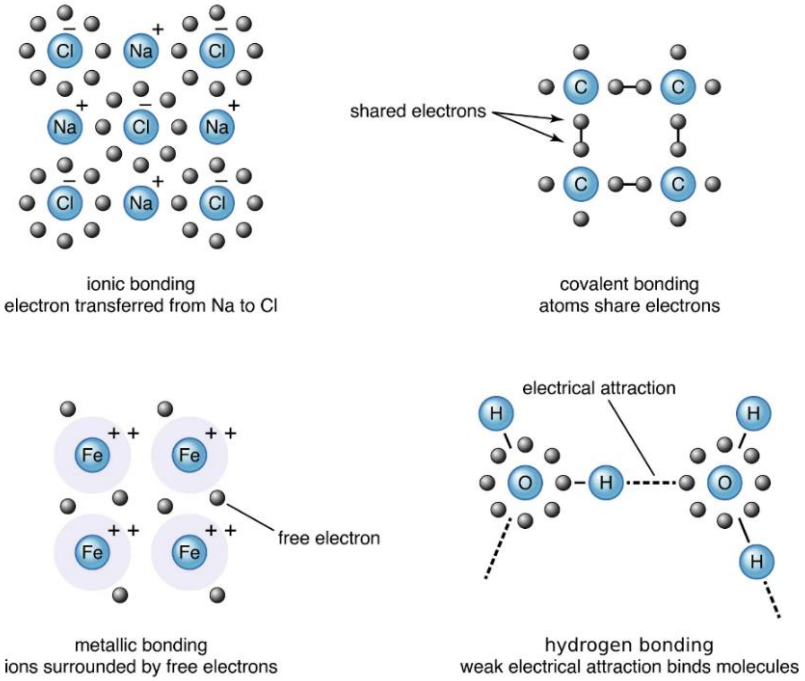



*Figure 1.4 Chemical bonds [30]*

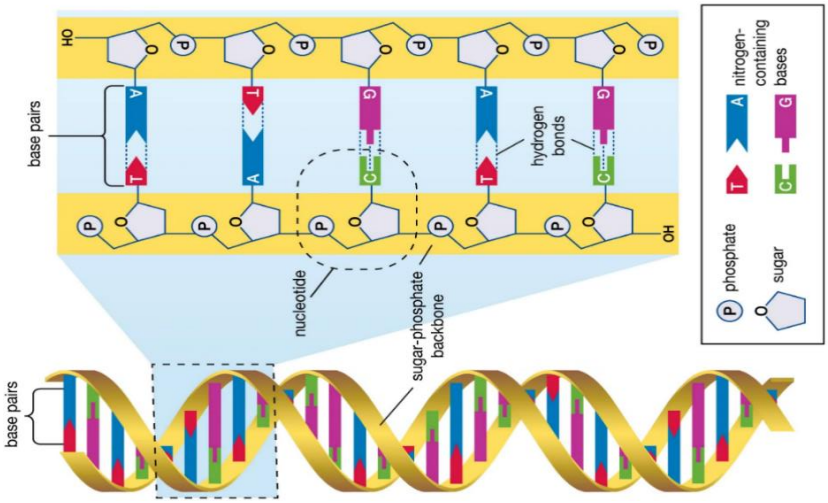



*Figure 1.5 DNA structure [31]*

### 1.3.5. RNA sequence

Similar to DNA; RNA (RiboNucleic Acid) is a linear polymer that consists of four distinct nucleotide subunits linked by phosphodiester bonds. Chemically, RNA differs from DNA in two primary ways: RNA is composed of ribonucleotides, which contain sugar ribose rather than deoxyribose, and it incorporates the base uracil (U) in place of thymine (T), while retaining the ability of uracil to form hydrogen bonds with adenine (A). Structurally, RNA is distinct from DNA in that it predominantly exists in a single-stranded form. This single-stranded nature allows RNA to fold into specific three-dimensional conformations, enabling it to perform precise structural and catalytic functions, thereby contributing to its functional diversity (e.g., protein synthesis, gene regulation, etc.) [25]. RNA is synthesized from DNA through transcription (Figure 1.6), which originates several RNA types including messenger RNA (mRNA) that carries genetic information from DNA to ribosomes, transfer RNA (tRNA) responsible for delivering amino acids to the ribosome during protein synthesis, and ribosomal RNA (rRNA) that functions as a structural and catalytic component of ribosomes [32].

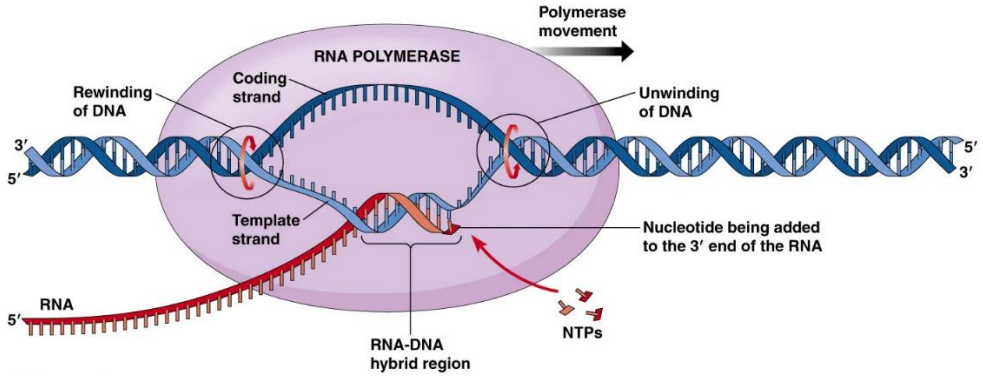



*Figure 1.6 RNA transcription [33]*

### 1.3.6. Biological pathways

These pathways refer to the structured, curated representations of biochemical or molecular processes that describe how genes, proteins, and other biomolecules interact with each other to carry out specific cellular functions, encompassing signal transduction, metabolism, or gene regulation. They typically integrate experimentally validated interactions into graph-like models where nodes correspond to biological entities and edges represent functional relationships or reaction steps (Figure 1.7). By organizing complex molecular events into interpretable frameworks; pathways enable the computational analysis of system-level behaviors, including the identification of dysregulated processes in a diseased state, prediction of functional gene modules, and interpretation of high-throughput omics data [34].

### 1.3.7. Protein sequences

Proteins are complex macromolecules that are composed of one or more polypeptide chains, which are linear sequences of amino acid residues covalently linked through peptide bonds. The sequence arrangement of these residues is dictated by the nucleotide sequence of the corresponding gene, which governs the protein's 3D-structure through the folding process, ultimately defining its functionalities. Proteins exhibit diverse lifespans, ranging from minutes to years, and play critical

roles in a wide range of cellular processes, including enzymatic catalysis, structural maintenance, cell signaling, immune responses, and cell cycle regulation. Moreover, they can even assemble into multi-protein complexes, enabling them to collaborate in performing intricate biological functions essential for sustaining cellular and physiological processes [25, 32].

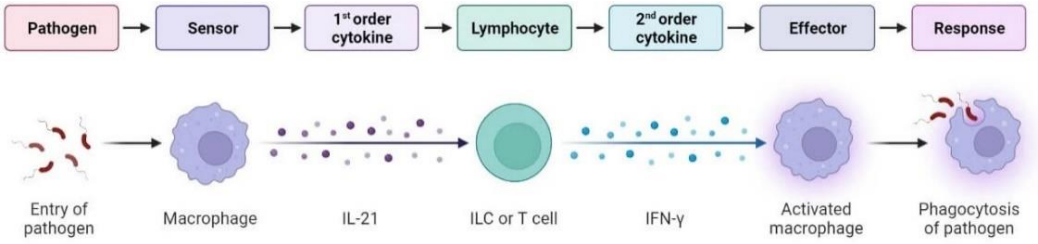



*Figure 1.7 Immuune response pathway [34]*

### 1.3.7.1. Amino acids

Amino acids (AA) are organic compounds that serve as the fundamental building blocks for proteins. Structurally, they consist of a central carbon atom (the α-carbon) bonded to four distinct groups: an amino group ($NH_2$), a carboxyl group (COOH), a hydrogen atom (H), and a unique side chain (R-group) that determines the amino acid's chemical properties (e.g., polarity, charge, and hydrophobicity) and classification (Figure 1.8), with a total of 20 canonical AA encoded by the universal genetic code. During protein syntheses, AAs are covalently linked through peptide bonds, formed between the carboxyl group of one amino acid and the amino group of another, resulting in the formation of polypeptide chains. Beyond their role in protein synthesis, amino acids also function as precursors for other biomolecules, participate in metabolic pathways, and act as signaling molecules, further highlighting their diverse biological significance [25].

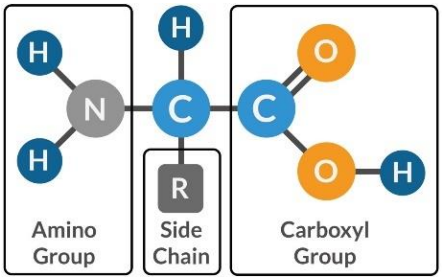



*Figure 1.8 The molecular structure of an amino acid [35]*

### 1.3.7.2. Central dogma of molecular biology

This belief is the foundational principle that governs the unidirectional flow of genetic information within biological systems, ensuring both stability and fidelity of genetic inheritance and expression by asserting that genetic information flows from DNA to RNA to protein (Figure 1.9). Specifically, DNA serves as the template for both its own replication (DNA to DNA) and the process of transcription into RNA (DNA to RNA), which is then translated to into functional proteins (RNA to protein) that carry the majority of cellular activities [36].

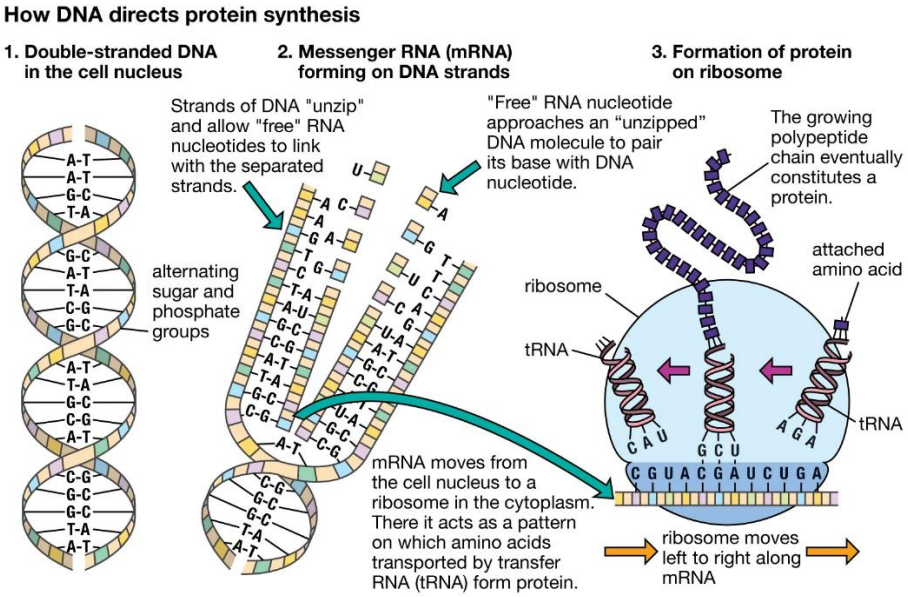



*Figure 1.9 The flow of genetic information from DNA to RNA to protein [31]*

### 1.3.7.3. Proteins function-based taxonomy

Proteins exhibit remarkable diversity both in structure and function, allowing them to participate in virtually all biological processes. Function-based protein taxonomy classifies proteins according to their biochemical activity, physiological role, and interactions within cells and tissues. This classification aids in understanding protein functionality, predicting the roles of uncharacterized proteins, and further advancing biomedical and biotechnological research. Broadly, proteins can be categorized into the following functional classes [25, 37]:

- ✓ **Enzymes:** Catalytic proteins that accelerate biochemical reactions by lowering the activation energy, thereby increasing reactions' efficiency and specificity. They play critical roles in metabolic pathways, DNA replication, and cellular signaling. Examples include "salivary amylase" which initiates the digestive process by catalyzing the breakdown of starch.
- ✓ **Structural Proteins:** A class of proteins that provide mechanical support and maintain the integrity of cells, tissues, and organs. They are often fibrous and form scaffolds that contribute to the stability, elasticity, and strength of biological structures, making them essential for maintaining cellular organization, tissue integrity, and mechanical functions. Examples encompass "keratin" which reinforces hair, nails, and skin's outer layer.
- ✓ **Transport Proteins:** Specialized proteins that facilitate the movement of molecules, ions, and other substances across biological membranes or within an intracellular environment. Varying from carrying proteins, channel proteins, to pump proteins; transport proteins play a crucial role in maintaining cellular homeostasis, regulating nutrient uptake, and waste removal.
- ✓ **Signaling Proteins:** Individuals that facilitate inter and intra cellular information transmission, via biochemical signals that ensure precise cellular responses to environmental stimuli. They function by detecting extracellular signals or intracellular changes, to which they trigger downstream responses through cascades of molecular interactions, rendering them essential for processes like growth, differentiation, and immune responses.

- ✓ **Regulatory proteins:** This class of proteins governs cellular processes by modulating the activities of other proteins, genes, or pathways, by attaching to their specific targets in order to either activate or inhibit their functions to ensure proper growth, development, and homeostasis. These proteins play critical roles in gene expression, signal transduction, metabolic pathways, and cell cycle.
- ✓ **Defensive proteins:** They are responsible for protecting organisms from pathogens, toxins, and other harmful agents. They play a critical role in immune responses by identifying, neutralizing, and eliminating foreign invaders. They include antibodies (immunoglobins), complement proteins, fibrinogen, and interferons.
- ✓ **Motor proteins:** They mainly focus on converting chemical energy (typically by the hydrolysis of ATP) into mechanical work, enabling movement at the cellular and molecular levels. They include myosin, kinesin, and dynein, with all three being considered essential for facilitating intracellular trafficking, and supporting various physiological processes.
- ✓ **Storage proteins:** These specialized proteins are capable of storing essential nutrients, ions, and other molecules for later use, providing a stable supply of resources that can be mobilized during periods of high demand or scarcity, granting them vital roles in maintaining metabolic balance and supporting growth and development.

#### 1.3.7.4. Proteins structure-based taxonomy

Protein structures are a fundamental aspect of molecular biology, biochemistry, and biophysics, as they determine the function, stability, and the interaction behaviors of proteins with other molecules Based on such vital factor; proteins can be organized into four hierarchical levels (Figure 1.10), with each level providing insight in regards to the protein's conformation and functionality [3, 32, 36].

**A. Primary structure**

Simply being the linear sequence of AA residues in a polypeptide chain, arranged according the genetic information encoded by the corresponding mRNA during translation. These residues are covalently linked together via the peptide bonds formed between the carboxyl group of one AA and the amino group of another, creating the protein backbone from which the R-groups extend, while leaving the first and last residues in the sequence with a free amino group (N-terminus) and carboxyl group (C-terminus) respectively. The peptide bonds linking residues are planar (non-rotatable), whereas the N-terminus and C-terminus exhibit a degree of rotational flexibility, thus enabling the generated sequence to fold into its unique 3D conformations [3, 36].

**B. Secondary structure**

Secondary structures in proteins are localized, repetitive spatial confrontations adopted by segments of a polypeptide chain, with different types of secondary structures potentially coexisting within a single polypeptide depending on its AA sequence. These structures are primarily stabilized via the hydrogen bonds formed between the backbone amide (NH) and the carbonyl groups (CO), while serving as an intermediate stage in the protein folding process. The most common secondary structures are the α-helices and β-sheets, which collectively constitute

approximately 60% of a protein's polypeptide chain, other less common structures include short U-shaped β-turns, random coils (loops), and irregular conformations [32, 36].

- ✓ **α-helices:** Compact right-handed helical conformations formed after the polypeptide chain adopts a spiral shape, stabilized primarily by the hydrogen bonds of the carbonyl's oxygen of an AA and the amide's hydrogen of another AA located four residues earlier in the sequence. This pattern features 3.6 AA residues per turn, with a vertical rise of $0.54nm$ per turn and $0.15nm$ per residue, with the hydrogen bonds aligning parallelly to the helix axis while the R-groups extend outward, further reinforcing the overall stability of both the α-helices and the protein.
- ✓ **β-sheets:** Pleated, sheet-like structures formed by β-strands, which are polypeptide segments that are 5-8 residues long. These strands are stabilized by hydrogen bonds between themselves, resulting in the construction of two types of β-sheets: parallel β-sheets, where the strands run in the same direction, resulting in evenly spaced but angled hydrogen bonds, and antiparallel β-sheets, where the strands run in opposite directions, producing perpendicular and more linear hydrogen bonds (generally considered more stable). In both types, the side chains of residues alternate above and below the sheet, and commonly found in regions of proteins that require structural strength, rigidity, or flexibility, contributing significantly to the overall stability and function of the protein.

### C. Tertiary structure

The tertiary structure of a protein refers to its 3D conformation, which is critical for determining its biological role and functions. This structure emerges from the precise folding of secondary structural elements that is governed by a combination of interactions among the AA side chains, including hydrophobic interactions, hydrogen bonds, disulfide bridges, and ionic bonds. The resulting 3D structure enables the protein's active sites to only interact with specific substrates or molecules with similar structures, enabling the protein to perform its biological roles with precision and efficiency. Tertiary structures are dynamic, allowing conformational changes in response to environmental cues or molecular interactions in order to regulate protein activity, further demonstrating the functional adaptability of tertiary structures in maintaining biological processes [38].

### D. Quaternary structure

Quaternary structures refer to the 3D arrangements adopted by multiple polypeptide chains (subunits) that bond together to form one functional protein complex. These subunits, which can be identical or different, are held together by the same non-covalent forces that stabilize tertiary structures. Quaternary structures are essential for the biological activities of many proteins, as they enable the formation of complex functional assemblies that cannot be formed by individual polypeptide chains, since the precise arrangements and interactions of subunits within quaternary confrontations are critical for the protein complex's stability, regulation, and functional efficiency [32, 36].

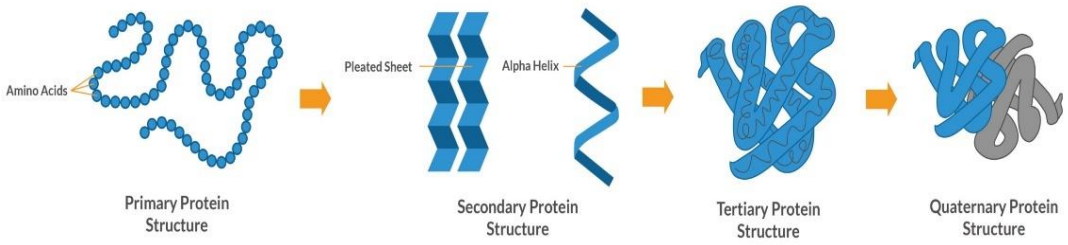



*Figure 1.10 Protein structures [35]*

## 1.4. Data infrastructures and analytical frameworks in bioinformatics

### 1.4.1. Databases

In bioinformatics, databases are extensive, structured repositories that house diverse range of biological data such as genomic sequences, protein structures, gene expression data, metabolic pathways, and phylogenetic information, all maintained in digital formats. These databases employ advanced storage, organization, management, and retrieval strategies to facilitate the systematic study of biological data, while ensuring its accuracy and consistency. By providing researchers with reliable access to well-curated data; bioinformatics databases play a crucial role in enabling data sharing, integration, and analysis, thereby driving advancements in fields such as genomics, proteomics, and systems biology [39]. Based on the nature of stored data and their specific applications; bioinformatics databases can be classified into several categories. Most well-known bioinformatics databases can be witnessed in Table 1.1.

#### 1.4.1.1. Sequence databases

Sequence databases are repositories that preserve raw genetic and proteomic information by storing digitalized versions of curated collections of biological sequences (e.g., DNA, RNA, or protein sequences) that are obtained either through experimental methodologies or via automated annotations. These databases serve as fundamental resources for researchers in genomics, proteomics, and other related disciplines [19].

#### 1.4.1.2. Structural databases

In bioinformatics; the 3D structural data of biological macromolecules (e.g., proteins, nucleic acids, carbohydrates, and small molecules) are preserved in special repositories known as structural databases. These databases serve as essential resources for studying molecular interactions, functional mechanisms, and drug design, by offering detailed atomic-level information regarding the bio-molecular architectures of molecules via the exploitation of experimental techniques such as X-ray crystallography, and nuclear magnetic resonance, which enable the visualization of molecular conformations and interactions at high resolutions [19, 40].

#### 1.4.1.3. Micro-array databases

Micro-array databases specialize in housing data derived from microarray experiments, which utilize high-throughput hybridization techniques to measure gene expression levels across thousands of genes simultaneously (e.g., gene expression profiles, annotations, and associated experimental metadata). These databases are designed to support scientific research by facilitating

the process of analyzing transcriptional regulations, biomarker discovery, and functional genomics, which in return enables researchers to study gene expression patterns across different conditions, tissues, or organisms [19].

#### 1.4.1.4. Metabolic and pathway databases

Metabolic and pathway databases are collections of systematically organized data that focus on biochemical reactions, metabolic pathways, regulatory networks across organisms, and the cellular process-related interactions of genes, proteins, and other molecules. These databases provide detailed annotations alongside curated data for the purpose of unraveling the flow of molecules within biological systems, which in return uncover concealed functional relationships among molecules. They contribute in drug discovery field by assisting in the development of novel therapeutic strategies for diseases by offering insights concerning metabolic, regulatory, and signaling pathways. They also support other disciplines such as metabolic engineering and systems biology by enabling metabolic analysis, pathway reconstruction, comparative genomics, and functional annotations [19].

#### 1.4.1.5. Chemical databases

Chemical databases specialize in storing, organizing, and providing detailed information concerning the molecular structures, physiochemical properties, biological activities, pharmacological effects, and the interactions of chemical compounds (e.g., small molecules, drugs, metabolites). Such data is typically obtained through experimental studies, computational modeling, and high-throughput screening techniques. Chemical databases play a crucial role in drug discovery, cheminformatics, toxicology, and metabolic research, enabling researchers to study compound-target interactions, predict drug efficacy, and bioactive compounds [19, 41].

#### 1.4.1.6. Genomics/Proteomics databases

Genomics and proteomics databases are repositories that preserve large-scale genomic and proteomic data (e.g., gene annotations, protein structures, and functional interactions) obtained from next-generation sequencing, genome-wide association studies, mass spectrometry, and other high-throughput techniques. Together; these databases provide comprehensive resources in functional genomics, biomarker discovery, and personalized medicine, with genomics databases focusing on whole-genome sequences, genetic variants, regulatory elements, and functional annotations, rendering them exploitable in comparative genomics, evolutionary biology, and disease genetics. Whereas proteomics databases store protein expression, post-translational modifications, interactions, and structural information, enabling them to be usable in functional proteomics, systems biology, and biomarker discovery [42].

#### 1.4.1.7. Disease databases

Disease databases systematically store genetic, molecular, and clinical data related to human diseases, obtained from experimental research, clinical studies, curated literature, and computational predictions. The curated information provided by these databases concerning disease-associated genes, mutations, pathways, and biomarkers; enables researchers to better study diseases' molecular basis, identify therapeutic targets, and develop personalized medicine

approaches, significantly advancing fields such as genomics, pharmacology, and clinical diagnostics [43].

#### 1.4.1.8. Bibliographic databases

Bibliographic databases focus on storing and indexing scientific literature, providing access to research articles, reviews, and conference proceedings, while incorporating text-mining tools, metadata indexing, and citation tracking to enhance literature retrieval and knowledge discovery. Their primary purpose is to ensure that researchers remain up-to-date to the latest findings, methodologies, and experimental results, which is accomplished through the aggregation of data from peer-reviewed journals, preprints, and academic publishers, entrusting them with a crucial role in systematic reviews, meta-analyses, and evidence-based research [44].

*Table 1.1 Databases used in bioinformatics [45-80]*

| Database name | Founder | Year | Data type | Database size |
|---|---|---|---|---|
| **Web of Science** | ISI | 1964 | Scientific literature | >170M records |
| **CSD** | Cambridge Crystallographic Data Centre | 1965 | Small molecule structures | >1M structure |
| **OMIM** | Johns Hopkins University | 1966 | Genes, genetic disorders | >27K entry |
| **PDB** | Cambridge Crystallographic Data Centre, Brookhaven National Laboratory | 1971 | Protein structures | >200K structure |
| **EMBL-EBI** | EMBL-EBI | 1980 | Nucleotide sequences, proteins | ? |
| **GenBank** | NIH | 1982 | Nucleotide sequences | >4.7B nucleotide sequence |
| **DDBJ** | National Institute of Genetics, Japan | 1986 | Nucleotide sequences | ? |
| **NDB** | Rutgers University | 1991 | Nucleic acid structures | ? |
| **KEGG** | Kanehisa Laboratories | 1995 | Pathways, genomes, diseases, chemicals | ? |
| **PubMed** | NLM, NIH | 1996 | Biomedical literature | >35M citations |
| **BioCyc** | SRI International | 1997 | Genomes, metabolic pathways | ? |
| **MetaCyc** | SRI International | 1997 | Metabolic pathways | >3K pathways |
| **NCBI Genome** | NCBI | 1997 | Genomic data | >2.8M genomes |
| **Orphanet** | INSERM | 1997 | Rare diseases | ? |

| | | | | |
|---|---|---|---|---|
| **dbSNP** | NCBI | 1998 | Genomic variations | >2 billion variation |
| **GEO** | NCBI | 2000 | Gene expression data | >7.6M sample |
| **RefSeq** | National Center for Biotechnology Information (NCBI) | 2000 | Nucleotide sequences, proteins | >300M sequence |
| **STRING** | EMBL, SIB | 2000 | Protein-protein interactions | >27B interactions |
| **BindingDB** | University of California, San Diego | 2001 | Protein-ligand interactions | >3M binding data |
| **Stanford Microarray Database** | Stanford University | 2001 | Gene expression data | ? |
| **ArrayExpress** | EMBL-EBI | 2002 | Gene expression data | ? |
| **UniProt** | EMBL-EBI, SIB, PIR | 2002 | Protein sequences and annotations | >240M protein sequence |
| **BioGRID** | Mount Sinai Hospital | 2003 | Protein, genetic, and chemical interactions | >2.8M interactions |
| **IntAct** | EMBL-EBI | 2003 | Protein-protein interactions | >1.6M interaction |
| **Reactome** | OICR, NYULMC, EMBL-EBI, OHCU | 2003 | Biological pathways | >22K pathway |
| **PeptideAtlas** | Institute for Systems Biology | 2004 | Proteomics data | ? |
| **PubChem** | NIH | 2004 | Chemical compounds | >100M compounds |
| **Scopus** | Elsevier | 2004 | Scientific literature | >82M records |
| **PRIDE** | EMBL-EBI | 2005 | Proteomics data | ? |
| **DrugBank** | University of Alberta | 2006 | Drug and drug target information | >65K drugs |
| **TCGA** | NCI and NHGRI | 2006 | Cancer genomics | ? |
| **ChemSpider** | Royal Society of Chemistry | 2007 | Chemical compounds | >100M compounds |
| **STITCH** | EMBL | 2008 | Chemical-protein interactions | >1.6B interactions |
| **WikiPathways** | Multiple contributors | 2008 | Biological pathways | >2K pathways |
| **ChEMBL** | EMBL-EBI | 2009 | Chemical compounds | >2.5M compounds |
| **ClinVar** | NCBI | 2012 | Clinical variants | >1M variants |

### 1.4.2. Bioinformatics computational tools

Bioinformatics tools are specialized software applications designed to employ computational algorithms and statistical methods in order to analyze, interpret, and visualize biological data (Table 1.2), enabling the extraction of meaningful insights from raw biological individuals. Their usage has become indispensable in modern bioinformatics, as they facilitate the processing of large-scale datasets, alongside other essential tasks including sequence alignment, genome assembly, protein structure modeling, pathway analysis, and machine learning-driven predictions. Consequently, they advance the identification biological patterns, functional annotations, evolutionary relationships, and the development of personalized medicine and drug discovery strategies [81]. Bioinformatics tools can be categorized based on their use cases, which include:

#### 1.4.2.1. Sequence data analysis tools

These tools are designed to process biological sequences (i.e., DNA, RNA, and proteins) via leveraging numerous methodologies such as sequence alignment, motif discovery, homology searching, and variant analysis in order to identify functional, structural, and evolutionary information [81]. Most well-known sequence analysis tools are depicted in Table 1.2.

#### 1.4.2.2. Genomic data analysis tools

They are developed to process and interpret large-scale genomic data by performing genome assembly, variant identification, and gene annotation and prediction to help uncover functional genomic elements, evolutionary relationships, and disease-associated genetic variations [82]. Numerous genomic data analysis tools are demonstrated in Table 1.2.

#### 1.4.2.3. Proteomic data analysis tools

Primarily constructed to process and interpret proteomic data such as proteins structures, functionalities, modifications, interactions, and expressions patterns; these tools are leveraged for the purpose of mass spectrometry data processing, protein identification and quantification, structural modeling, functional annotation, post-translational modifications profiling, and protein-protein interaction networks analysis [83]. Miscellaneous proteomic data analysis tools are displayed in Table 1.2.

#### 1.4.2.4. Pathway and functional annotation analysis tools

These tools are designed to interpret high-throughput biological data by mapping genes and proteins into their associated processes, functions, or pathways. Their main purpose is to identify functional roles, regulatory interactions, and molecular pathways of molecules within cellular processes [84]. Diverse pathway and functional annotation analysis tools can be witnessed in table 1.2.

#### 1.4.2.5. Cheminformatics and drug discovery tools

Special tools that leverage machine learning, molecular simulations, and other aspects (e.g., molecular docking, virtual screening, quantitative structure-activity relationship (QSAR) modeling, etc.), in order to universally comprehend chemical compounds, predict the results of

their interactions, and facilitate the identification and optimization of potential drug candidates [85]. Some of their well-known tools are depicted in Table 1.2.

#### 1.4.2.6. Machine learning and data mining tools

Prominent tools capable of extracting meaningful patterns, predict biological functions and structures, and analyze complex biological datasets, by leveraging supervised and unsupervised learning techniques to extract implicit knowledge from genomic, proteomic, and other omics data [86]. Several machine learning and data mining tools can be observed in table 1.2.

#### 1.4.2.7. Data visualization tools

Graphics-oriented tools designed to represent complex biological datasets, facilitating pattern recognition, interpretation, and hypothesis generation via utilizing statistical graphics, interactive visual analytics, and multidimensional data mapping to display individual-related information [87]. Various data visualization tools are depicted in table 1.2.

#### 1.4.2.8. Other tools

In addition to the aforementioned tools essentially required in the field of bioinformatics, there are other tools that despite being less prominent; remain indispensable, such as microbiome and metagenomics analysis tools leveraged to analyze microbial communities and environmental sequencing data, epigenomics analysis tools that specialize in studying DNA methylation and chromatin accessibility data, and metabolomics data analysis tools used to investigate metabolomics profiles.

*Table 1.2 Most well-known bioinfomratics tools [88-111]*

| Tool's category | Tool's name | Tool's purpose |
|---|---|---|
| **Sequence analysis** | BLAST | Compare nucleotide or amino acid sequences for similarity search (single alignment) |
| | Clustal Omega | Use seeded guide trees and HMM profile-profile techniques for multiple sequence alignment of DNA or protein sequences |
| | HMMER | Search for homologous sequences using hidden Markov models (motif search) |
| **Genomic data analysis** | SPAdes | Small genomes assembly |
| | GATK | Identify SNPs and indels (variant calling) |
| | AUGUSTUS | Predicts genes in eukaryotic genomic sequences |
| **Proteomic data analysis** | SwissDock | Predicts binding of small molecules to proteins (molecular docking) |
| | AlphaFold | Uses artificial intelligence to predict protein 3D structures |
| | PyMOL | Visualize proteins 3D structures |
| **Pathway and functional annotation analysis** | KEGG Mapper | Map and analyze biological pathways |
| | DAVID | Functional annotation and enrichment analysis |
| | STRING | Predicts and visualizes protein-protein interactions |

| | | |
|---|---|---|
| **Cheminformatics and drug discovery** | AutoDock | Predict how small molecules (ligands) bind to target proteins (molecular docking) |
| | Schrödinger Suite | Enables molecular docking, virtual screening, and drug design |
| | RDKit | Helps analyze chemical structures, predict properties, and optimize drug candidates |
| **Machine learning and data mining** | BioBert | Extracts relationships from biological literature using NLP (text mining) |
| | Scikit-learn | Provide general-purpose machine learning approaches for bioinformatics |
| | UMAP | Reduces high-dimensional biological data |
| **Data visualization** | Cytoscape | Visualize molecular interaction networks |
| | IGV | Interactive visualization of genomic data |
| | ggplot2 | Creates high-quality visualizations in R |
| **Other** | QIIME2 (microbiome and metagenomics) | Analyze and visualize microbiome sequencing data |
| | Bismark (epigenomics) | Map bisulfite-treated sequencing reads and perform methylation calls |
| | XCMS (metabolomic data) | Process and analyze mass spectrometry-based metabolomics data |

## 1.5. Application areas of bioinformatics

The promising potentials residing in bioinformatics discipline render it applicable in various domains, including but limited to:

### 1.5.1. Genomics

Bioinformatics is considered as an indispensable tool in the field of genomics, mainly due to the numerous techniques its tools leverage in order to process large-scale genomic data, whether by (1) facilitating genome sequencing and assembly through reconstructing whole genomes from raw sequencing reads, (2) providing deeper insights to genetic variations, disease mechanisms, and evolutionary relationships, by elevating the overall understanding of genes' roles, expressions, and regulations by identifying their functional characteristics thanks to its advanced gene annotation techniques, or (3) simply by helping collect, curate, and integrate diverse datasets together to better elucidate genes' interactions, biological pathways, and complex cellular processes. Additionally, the integration of bioinformatics tools in personalized medicine discipline enables the analysis of genetic profiles on an individual level, resulting in precise identifications of genetic mutations and their effects and risks, and more accurate diagnoses and therapeutics strategies. Finally, the pivotal role bioinformatics plays in comparative genomics allows for the examination of evolutionary relationships and the identification of conserved genes, helping further understand the evolutionary paths followed by certain species and the adaptation techniques they adopted to avoid threats, and prevent their extinct [112].

### 1.5.2. Proteomics

The vital role played by bioinformatics in proteomics is manifested in its ability to provide potent tools that're capable of processing extensive protein datasets, encompassing (1) the identification and quantification of proteins based on mass spectrometry data that aid in clarifying numerous cellular functions and disease mechanisms, (2) the prediction and modeling of protein structures that help elucidate protein functions and interactions with drugs by unraveling their binding sites, (3) the construction and analysis of protein-protein interaction networks which advance the overall understanding of cellular pathways, and (4) the simplification of post-translation modification analysis [113]. Bioinformatics is also exploited in biomarker discovery by analyzing proteins expressions across tissues and individuals to help identify potential prognostic markers, contribute in the process of vaccine production, regulation of expression patterns responding to diverse signals, and the analysis of disease-associated protein pathways [3].

### 1.5.3. Transcriptomics

Bioinformatics is considered integral in transcriptomics due to fundamental position it occupies in processing, aligning, quantifying, and interpreting large-scale transcriptomic data (e.g., RNA sequencing datasets, microarrays). Through leveraging the computational tools in possession of bioinformatics; researchers can easily align RNA sequencing reads, and quantify gene expression levels in order to uncover differentially expressed genes associated with diseases, analyze alternative splicing events, non-coding RNAs, and gene regulatory networks to assist in comprehending the impact of gene expression alterations on cellular diversity, biological functions, pathways, and diseases mechanisms. Moreover, bioinformatics can combine transcriptomics data with other omics data to provide a broader view to biological systems and their processes [114].

### 1.5.4. Phylogenetics

Phylogenetics depends heavily on bioinformatics, as it grants it the computational frameworks essential for constructing, analyzing, and interpreting phylogenetic trees, offering critical insights into evolutionary relationships among species, individuals, or molecules. The main use of bioinformatics in phylogenetics is to enable the detection of sequence similarities and divergences through alignment operations, facilitate the construction of phylogenetic trees from sequence data, and support the elucidation of evolutionary relationships and species taxonomies through whole-genome analysis pipelines. These operations are crucial for understanding complex evolutionary patterns, ancestry, and conserved genes. Additionally, bioinformatics plays a key role in unraveling microbial evolution and antibiotic resistance by detecting horizontal gene transfer events between species, thereby contributing in understanding the spread of resistance genes, alongside assisting in inferring divergence times between species through molecular clock analysis for evolutionary study purposes [115, 116].

### 1.5.5. Drug discovery

By possessing the ability to quickly and accurately analyze vast amounts of biological data, bioinformatics can easily advance drug discovery field by accelerating the identification, design, of optimization of potential drug candidates. This task is accomplished through processing both

genomic and proteomic data in order to pinpoint disease-associated genes and proteins suitable as potential targets, reduce both time and cost of experimental trails via predicting the binding affinities of molecules to target proteins using virtual screening methods, and repurpose drug usages and assign them novel therapeutic applications through re-analyzing chemical libraries. Moreover, in pharmacogenomics, bioinformatics accelerates the process of developing novel drugs by enabling the prediction of their absorption, distribution, metabolism, excretion, and toxicity properties (i.e., ADMET) via employing machine learning approaches, alongside enabling more accurate personalized medicines for unique genetic profiles by distinguishing the influence of genetic variations on drug responses [117, 118].

### 1.5.6. Forensic science

The impact of bioinformatics on forensic science is remarkable, as it provides it with the computational tools needed to accelerate and enhance the analysis and interpretation of biological evidence, significantly improving the accuracy and efficiency of investigations. Its main applications include (1) DNA profiling used to analyze and extract unique genetic fingerprints residing in DNA sequences, which help identify individuals and perform paternity tests, (2) enabling the use of metagenomic techniques to help link suspects to locations or objects by analyzing microbial communities left by human skin on contacted surfaces of picked from soil, (3) provide the ability to process degraded samples and trace ancestral lineage through analyzing mitochondrial DNA and Y-chromosome, and (4) manage forensic databases by storing and comparing genetic data to match crime scene evidence with known offenders [119].

### 1.5.7. Environmental sciences

The diverse environmental sciences utilize bioinformatics for numerous purposes, ranging from (1) leveraging metagenomic data to help identify microbial species, enzymes, and microorganisms' pathways that can provide efficient waste cleanup and sustainable alternative energy resources, (2) enriching climate change studies through studying microbial roles in carbon and nitrogen cycling, (3) the identification of genes associated heavy metals and toxins resistance to help prevent pollution, to (4) the assistance in monitoring the spread of pathogens in wildlife and ecosystems especially those that can be transmitted to humans [120, 121].

### 1.5.8. Agricultural sciences

Similar to all other disciplines that are interested in biological life forms; the miscellaneous agricultural sciences do exploit the computational tools of bioinformatics for a plethora of causes, whether by (1) leveraging its genomic tools to either select plant seedlings with most desirable traits, or to alter the genetic data of plants for improved environmental tolerance, increased disease resistance, elevated nutrition content, and agricultural productivity, (2) studying the genetic data of pests and help infer insect control strategies that are more effective and less inhuman, (3) conducting genomic analysis on livestock and their associated pathogens to ameliorate traits like milk production, meat quality, as well as enhance the precision of disease identification to facilitate the development of targeted vaccines, or (4) by distinguishing beneficial microorganisms that can be utilized as biofertilizers to help augment soil's fertility [122, 123].

## 1.6. Bioinformatics challenges and limitations

Despite the fact that bioinformatics has successfully revolutionized biological research on numerous levels by leveraging diverse computational tools and algorithms; it still has its own challenges and limitations that hinder its data acquisition, retrieval, analysis, interpretation, and visualization tasks. Some of its major challenges and limitations include [19, 124]:

- ✓ **Data management and storage problems:** The exponential growth of biological data generated from high-throughput experiments and large-scale literature curation has created significant challenges in efficiently aggregating, organizing, and storing this information, requiring more advanced data-management strategies as well as increasingly powerful computational infrastructure.
- ✓ **Data quality issues:** This challenge arises from the inherent susceptibility of biological data to noise, errors, and biases introduced during experimental generation, sequencing, annotation, and sample collection, which can compromise downstream analyses and lead to misleading interpretations or unreliable conclusions.
- ✓ **Data integration challenges:** This obstacle arises from the breadth of biological domains and the diversity of data sources, as integrating information across heterogeneous datasets is often hindered by inconsistencies in formats, standards, metadata, and annotation quality, making it difficult to merge and interpret data in a coherent and biologically meaningful way.
- ✓ **Complexity of biological systems:** The inherent complexity of biological systems poses major challenges for computational modeling, as accurately representing their dynamic, multiscale interactions without losing or oversimplifying critical biological information is difficult.
- ✓ **Computational limitations:** Computational limitations arise when processing large-scale, heterogeneous, and high-dimensional biological datasets, restricting the ability of bioinformatics tools to efficiently analyze complex biological systems and generate accurate predictions within reasonable computational time and resource constraints.
- ✓ **Lack of experimental validation:** This issue stems from the reliance of many bioinformatics analyses on computational inference, statistical correlations, and predictive modeling without accompanying it with wet-lab confirmation, which introduces uncertainty and increases the risk of false positives or biologically irrelevant findings.
- ✓ **Shortage of interdisciplinary expertise:** The limited number of researchers with advanced training in biology, computer science, and statistics hampers effective collaboration and innovation, resulting in inefficiencies in data analysis, tool development, and translational bioinformatics research.
- ✓ **Ethical and privacy concerns:** The handling of sensitive genetic and health data poses significant risks of misuse, highlighting the need for robust ethical frameworks and privacy regulations, particularly in personalized medicine and genetic testing.

## 1.7. Conclusion

The field of bioinformatics has radically transformed the study of biological systems, providing essential computational frameworks to manage, analyze, and interpret the growing volume of molecular and clinical data. By integrating biology, computer science, and statistics; bioinformatics enables the modeling of complex interactions, prediction of molecular functions,

and discovery of novel therapeutic targets, thereby accelerating research in drug discovery, genomics, and systems biology. As datasets become increasingly large and heterogeneous, advanced computational approaches are becoming more crucial in extracting meaningful insights and guiding experimental validation. The continued development of bioinformatics promises to revolutionize personalized medicine, biotechnology, and molecular therapeutics, offering unprecedented opportunities to understand disease mechanisms, predict treatment responses, and design novel interventions with precision.

# Chapter 2: Machine Learning and its Core Concepts and Techniques

Machine learning has emerged as a foundational paradigm for extracting knowledge, making predictions, and supporting decision-making from increasingly large, complex, and heterogeneous datasets. By enabling systems to learn patterns and relationships directly from data, ML has become a cornerstone of modern artificial intelligence, with applications spanning science, engineering, healthcare, and industry.

This chapter presents a comprehensive overview of the fundamental concepts underpinning MLs. It begins by defining ML and situating it within its broader research context, followed by a systematic presentation of the main learning paradigms. Key concepts related to data types, preprocessing techniques, and end-to-end ML workflows are then discussed. The chapter further reviews common ML algorithms, deep neural network architectures, and ensemble methods, before highlighting representative application domains. Finally, major computational and real-world challenges associated with deploying ML systems are examined. Collectively, this chapter establishes the theoretical and methodological foundation necessary for understanding and developing advanced ML-driven systems in subsequent chapters.

## 2.1. Introduction

ML is a rapidly evolving field that focuses on enabling computational systems to automatically learn from data and improve their performance on a given task without being explicitly programmed. Drawing from disciplines such as statistics, optimization, computer science, and AI; ML provides powerful tools for modeling complex phenomena and uncovering hidden patterns in data. The unprecedented growth in data availability (driven by advances in digital technologies, sensors, and large-scale information systems) has further accelerated the adoption of ML across a wide range of domains, including computer vision, natural language processing (NLP), recommender systems (RS), healthcare, and finance. At its core, ML aims to transform raw data into actionable knowledge through well-defined learning paradigms that include: supervised learning, where models learn from labeled data; unsupervised learning, which discovers latent structures without explicit supervision; semi-supervised learning, a hybridization of the aforementioned two paradigms; and reinforcement learning, where agents learn optimal behaviors through interacting with an environment. Each paradigm encompasses a variety of tasks such as regression, classification, clustering, dimensionality reduction, and generative modeling, all of which are fundamental to modern intelligent systems. Beyond learning algorithms; effective ML relies heavily on data-related considerations and systematic workflows. Data preprocessing, feature engineering, normalization, augmentation, and transformation play a critical role in shaping model performance and robustness. Equally important is the ML pipeline, which includes problem formulation, model selection, hyperparameter tuning, training, evaluation, and deployment. Recent advances in deep learning (DL) and ensemble methods have further expanded the representational power of ML models, enabling them to tackle highly complex and high-dimensional problems.

This chapter provides a structured and comprehensive introduction to these general concepts of ML. By reviewing learning paradigms, algorithms, workflows, applications, and challenges, it lays the conceptual groundwork required for understanding the more specialized and domain-specific ML methodologies developed in later chapters of this thesis.

## 2.2. Definition of Machine Learning

The term “machine learning” was first introduced in 1959 by Arthur Samuel as a façade that describes *“the discipline that grants computers the ability to learn without being explicitly*

*programmed"* [125]. In other words, ML is a subfield of AI that focuses on developing algorithms and statistical models capable of rendering machines able to imitate humans' ability to learn and generalize knowledge without explicit instructions [126], with the main differences between ML and traditional programming paradigms being depicted in Table 2.1. In order to accomplish such goal; ML approaches leverage methods from various domains, encompassing linear algebra, probability and statistics, information theory, and optimization techniques, for the sole purpose of improving performance for tasks based on past experience [127]. The adaptability provided by ML algorithms at solving complex problems rendered them indispensable across diverse disciplines, as their ability to autonomously extract knowledge from data, facilitate pattern recognition, provide predictive analytics, and minimize human intervention in decision-making processes, resulted in an unprecedented leap in how systems operate, how research is conducted, and how society interacts with technology. Particularly, in biomedicine and healthcare, ML has significantly accelerated disease diagnostic capabilities and drug discovery procedures [128], financial systems on the other hand have undergone paradigm shifts thanks to ML-powered fraud detection algorithms and quantitative trading frameworks [129], similar to transportation systems that have been fundamentally transformed by autonomous vehicle technologies [130], while entertainment and e-commerce platforms greatly benefited from adaptive recommendation systems [131], ultimately demonstrating ML pivotal role in modern computational problem-solving and its capacity to reshape industry standards.

*Table 2.1 Main differences between traditional programming and machine learning paradigms [132, 133]*

| Aspect | Traditional programming | Machine learning |
|---|---|---|
| **Decision-making process** | Limited to the explicit logic set by the programmer in each explicit instruction | Derived from observed data (can even uncover patterns unknown to the developer) |
| **Development process** | Linear and structured, requires deep understanding of the problem | Focus more on data collection, cleaning, model selection, training, and evaluation |
| **Adaptability** | Static, requires manual updates for alternative scenarios | Dynamic, can easily adapt to new data and changing environments |
| **Scalability** | Struggle with complex or ambiguous tasks | Excels at complex tasks (mainly due to the pattern recognition ability) |
| **Generalization** | Depend on the foresight of the developer, only applicable on the predefined cases | Depend on quality and diversity of data, generalizes well to unknown scenarios |
| **Outcome interpretability** | Traceable and explainable | Mostly impossible to interpret due the black-box nature and high-complexity of models |
| **Solvable problems** | Deterministic tasks (inventory management, financial calculations, sorting tasks) | Probabilistic tasks (image and speech recognition, personalized recommendations) |

## 2.3. Types of machine learning

ML algorithms and approaches are broadly categorized into four main paradigms, which include:

### 2.3.1. Supervised learning

Supervised learning is a ML paradigm in which a predictive model aims to learn a mapping function from input variables to corresponding output labels based on a labeled dataset composed of input-output pairs. The objective is to exploit patterns present in observed instances to accurately infer the labels of previously unseen samples [134]. In practice, supervised learning models are trained using a dataset that is typically partitioned into distinct subsets, most commonly training, validation, and test sets. The training set, which often comprises approximately 70-80% of the available data, should be representative of the underlying data distribution to reduce bias and improve generalization performance. During the training phase, the model iteratively processes the input features and their associated ground-truth labels, optimizing its parameters by minimizing a predefined loss function that quantifies the discrepancy between predicted outputs and true labels, while updating them using an optimization algorithm (usually based on a gradient-based method). Model performance and generalization capabilities are then evaluated on a separate validation or test set (whose labels are not used during parameter learning) using appropriate evaluation metrics, ensuring an unbiased assessment of the model's ability to generalize to unseen data [135]. The most common supervised learning paradigms are illustrated in Figure 2.1, and they include:

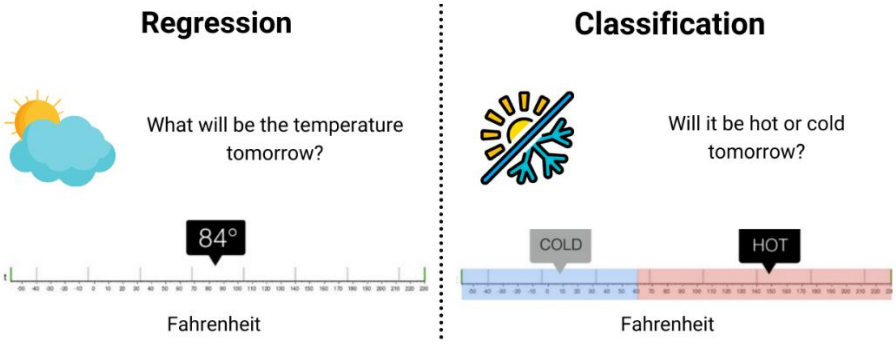



*Figure 2.1 Supervised learning paradigms [138]*

#### 2.3.1.1. Regression

A statistical supervised learning paradigm where an algorithm aims to model the continuous relationship between a dependent variable (label) and one or more independent variables (input features) [136]. This process involves estimating a mapping function (vary from linear to nonlinear depending on both the deployed model's complexity and the underlying problem) that best fits the observed data by minimizing a predefined loss function (e.g., mean square error) that incorporates both predicted and actual training labels [125]. Such optimization process however may be affected by several challenges, including multicollinearity (i.e., occurs when independent variables are highly correlated, undermining the model's statistical reliability.), noise, limited data, or inappropriate model complexity, which can hinder convergence toward an optimal solution, therefore necessitating the adoption of mitigation strategies such as feature engineering, regularization techniques, and cross-validation to help reduce bias, prevent underfitting, overfitting, and improve generalization to unseen data. Regression constitutes a cornerstone of both statistical modeling and ML due to its interpretability, flexibility, and wide applicability. It is particularly valued for its ability to quantify relationships between variables, supporting both

predictive tasks (e.g., forecasting stock prices) and inferential analyses (e.g., identifying key factors influencing disease progression). As a result, regression methods are extensively used across diverse domains, including economics (sales prediction), biomedicine (dose-response modeling), engineering (system performance optimization), and climate science (temperature estimation), highlighting their foundational role in data-driven modeling and decision-making [137].

#### 2.3.1.2. Classification

Classification is a supervised learning paradigm in which an algorithm learns to assign instances to a finite set of discrete categories (classes) by identifying discriminative patterns within their input features. Unlike regression, which models relationships involving continuous target variables, classification focuses on learning a mapping from one or more independent variables $X$ to a discrete dependent variable $Y$. This process is achieved by exposing the model to labeled training data, enabling it to learn a decision function (also called boundary) that best separates instances belonging to different classes (or equivalently, groups together instances of the same class). During training, the classifier optimizes a task-specific loss function (e.g., cross-entropy loss) that penalizes misclassifications, thereby minimizing empirical risk and capturing latent patterns governing class membership. Once trained, the model can generalize this learned decision function to predict the labels of unseen or new instances. The effectiveness of a classification task is strongly influenced by data quality, feature representation, class balance, and the choice of learning algorithm. Common classification algorithms include logistic regression, decision trees, support vector machines (SVMs), k-nearest neighbors (k-NN), and neural networks, which differ in terms of interpretability, scalability, and their ability to model linear or non-linear decision boundaries. Due to their flexibility and robustness, classification models are widely applied across numerous domains, including medical diagnosis (e.g., benign vs. malignant tumor classification), computer vision (e.g., object recognition), and natural language processing (e.g., sentiment analysis) [139, 140]. In general, classification problems can be grouped into five main categories (Figure 2.2), which include:

**A. Binary classification**

Binary classification is the simplest form of classification, where instances are assigned to one of two mutually exclusive classes (typically expressed as 0 and 1, negative and positive, etc.) based on a selected set of input features (Figure 2.3 (A)), with the objective being the identification of the decision boundary that optimally separates the two classes in the feature space. This type of classification is employed in numerous fields including medical diagnosis, fraud and spam detection, and disaster prediction, with classifiers such as logistic regression, SVM, and neural networks being the most suited algorithms to approach such tasks [141].

**B. Multi-class classification**

Unlike binary classification, multi-class classification involves assigning instances to one of three or more mutually exclusive classes, as illustrated in Figure 2.2 (B) (e.g., handwritten digit recognition). This setting introduces additional complexity, as the model must learn discriminative boundaries among multiple classes simultaneously, often under conditions of class imbalance.

Consequently, specialized evaluation metrics (such as the confusion matrix, macro- and micro-averaged precision/recall, F1-score, and Cohen's kappa) are commonly employed to provide a more reliable assessment of performance across all classes [143]. Multi-class classification problems can be addressed through two principal strategies. The first is direct (native) multi-class modeling, where algorithms inherently support multiple classes by optimizing a single global objective function over all classes, typically using categorical cross-entropy loss. Examples include neural networks, decision trees, random forests, k-nearest neighbors, and multinomial logistic regression. The second strategy relies on problem decomposition techniques, which reduce the multi-class problem into multiple binary classification tasks. Common decomposition schemes include One-vs-Rest (OvR), where one classifier is trained per class against all remaining classes, and One-vs-One (OvO), where classifiers are trained for every pair of classes (Figure 2.3). These approaches are particularly useful for classifiers that are inherently binary (e.g., linear SVMs or binary logistic regression), allowing them to be extended to multi-class settings by combining the outputs of several binary models [142].

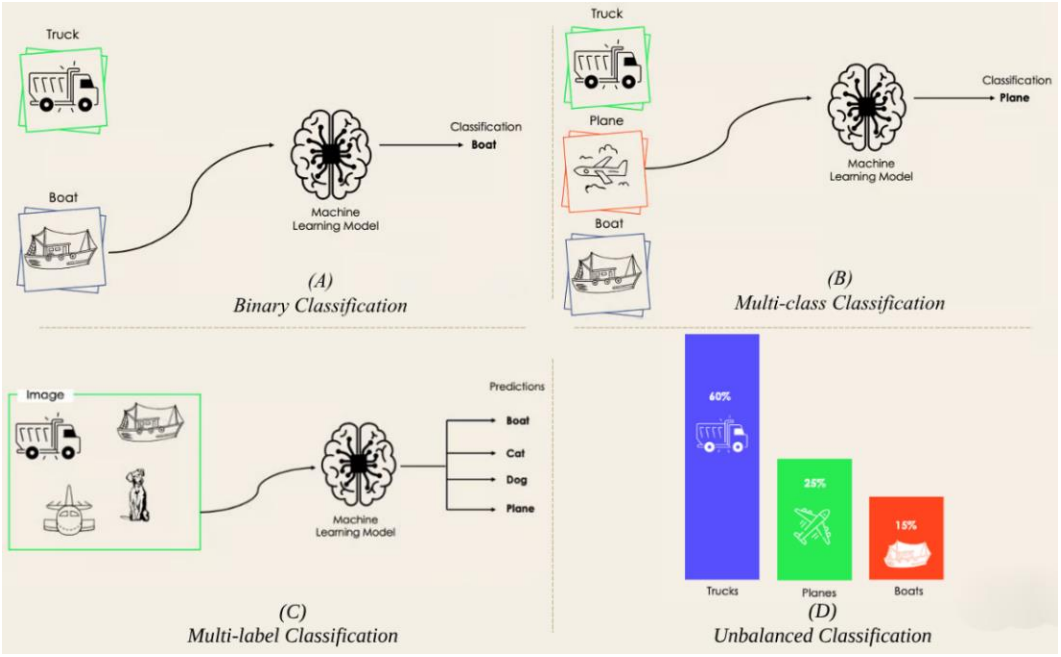



*Figure 2.2 Types of supervised classification tasks [142]*

## C. Multi-label classification

This classification paradigm extends traditional multi-class classification by addressing scenarios in which a single instance may simultaneously belong to multiple, non-mutually exclusive classes (Figure 2.2 (C)). Formally, the objective is to learn a mapping function that projects an input feature vector $X \in \mathbb{R}^d$ into a binary label vector $Y \in \{0,1\}^L$, where $L$ denotes the cardinality of the label space and each component indicates the presence or absence of a given label. Unlike multi-class classification, where exactly one label is active per instance, multi-label settings require the model to capture label co-occurrence patterns and inter-label dependencies. This paradigm is commonly implemented using neural networks with sigmoid-activated output

units, where each label is modeled independently and optimized using loss functions such as binary cross-entropy, although more advanced approaches explicitly model label correlations through structured prediction, classifier chains, or attention-based mechanisms. Multi-label classification has wide applicability across numerous domains, including bioinformatics (e.g., gene or protein function annotation), computer vision (e.g., multi-object or scene recognition), and natural language processing (e.g., document or topic tagging), where entities naturally exhibit multiple semantic associations simultaneously [142, 144].

### D. Hierarchical classification

Hierarchical classification mainly focuses on associating instances with classes that are organized within taxonomic hierarchy (tree, directed acyclic graph), where labels exhibit parent-child and sibling dependencies (Figure 2.4), with higher-level classes maintaining broader semantic scopes, in contrast to lower-level ones that provide increasingly specific categorizations [145, 146]. The primary objective of this classification is to assign instances to their most specific valid classes in the hierarchy; while maintaining hierarchical constraints (the prediction of a child class requires the validation of all its parent classes), which can be achieved via either: (i) a top-down approach, that starts at a root class, and recursively descend through classes by splitting to more specific ones until reaching the most suited one, or (ii) a bottom-up approach, which initiates estimation at a specific class, then aggregates evidence upward through the hierarchy. The inherent structural constraints impose unique computational challenges (e.g., error propagation) that necessitate specialized algorithms and evaluation metrics (H-loss, path accuracy) [19], rendering hierarchical classification valuable in domains that require precise taxonomic alignment, including e-commerce (product categorization), and biomedical ontology mapping.

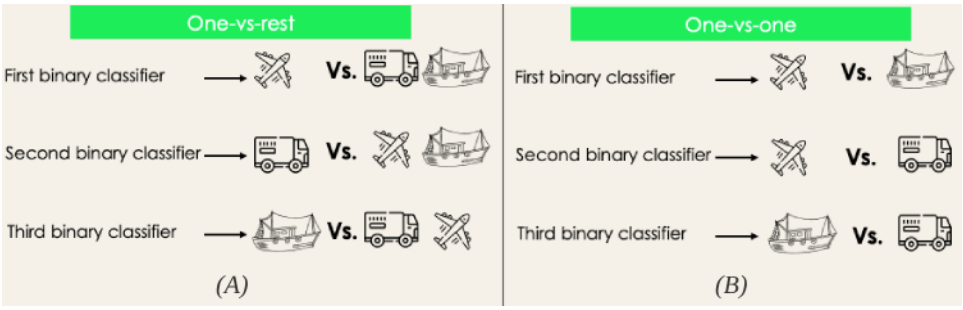



*Figure 2.3 Multi-class decomposition strategies adopted by binary classifiers [142]*

### E. Unbalanced classification

Imbalanced classification addresses learning scenarios in which the class distribution is highly skewed, with one or more minority classes being significantly underrepresented relative to majority classes, as illustrated in Figure 2.2 (D). This setting commonly arises in real-world applications where the minority class is of primary interest, such as fraud detection (rare fraudulent transactions), medical diagnosis (low prevalence of positive disease cases), anomaly detection, and endangered species monitoring. The central objective of imbalanced classification is to improve the recognition and predictive performance on minority-class instances while preserving acceptable performance across all classes. Achieving this objective typically requires specialized

strategies, including data-level approaches (e.g., oversampling, undersampling, and synthetic data generation), algorithm-level methods (e.g., cost-sensitive learning and ensemble techniques), and hybrid solutions that mitigate the model's bias toward majority classes. Additionally, conventional accuracy is often misleading in imbalanced settings due to the accuracy paradox (high accuracy can be achieved by ignoring minority classes), necessitating the use of more informative evaluation metrics such as precision, recall, and F1-score [143] [147].

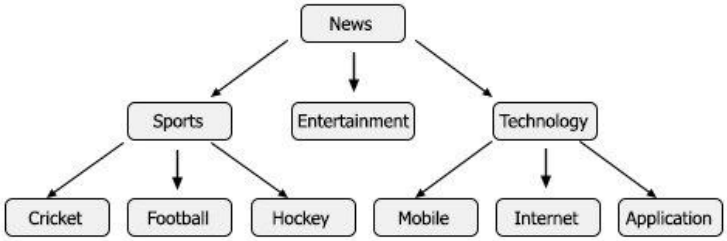



*Figure 2.4 Heirachical classification [148]*

### 2.3.2. Unsupervised learning

Unlike supervised learning, unsupervised learning is a ML paradigm that aims to uncover latent patterns, discover hidden relationships, and model the underlying structure governing data distributions without relying on labeled outputs. Instead, it operates solely on the intrinsic properties of unlabeled input data, completely eliminating the need for explicit supervision or ground-truth annotations [141]. This characteristic makes unsupervised learning particularly valuable in scenarios where labeled data is scarce, costly, or impractical to obtain. By autonomously identifying inherent features and statistical regularities within the data, unsupervised methods can group data points based on learned similarities, reveal low-dimensional representations, and expose latent structures that may remain undetected by supervised approaches. As a result, unsupervised learning plays a central role in exploratory data analysis, anomaly detection, and feature discovery across diverse domains [149]. Despite these advantages, unsupervised learning presents several challenges. The absence of labeled data deprives models of explicit optimization targets, which can lead to ambiguous solutions and makes performance evaluation inherently difficult. Consequently, the quality of learned representations may be sensitive to noise, feature scaling, and the choice of similarity measures or model assumptions, potentially resulting in suboptimal or unstable outcomes. Furthermore, unsupervised methods often incur substantial computational costs, as they must explore large and complex hypothesis spaces to identify meaningful structures, particularly in high-dimensional settings, leading to increased training time and scalability concerns [141] [149]. Therefore, the adoption of unsupervised learning typically involves a trade-off between modeling autonomy and predictive reliability. In practice, unsupervised learning has been successfully applied across a wide range of real-world applications. In marketing, clustering algorithms such as k-means are used for customer segmentation and personalized targeting. In finance, techniques like isolation forests and clustering-based anomaly detection support fraud identification. In bioinformatics, dimensionality reduction and clustering methods (including hierarchical clustering and t-distributed stochastic neighbor embedding) are employed to analyze gene expression profiles and identify disease

subtypes. Similarly, in computer vision, principal component analysis and autoencoders are widely used for image compression and representation learning [150] [151]. The most common unsupervised learning paradigms are illustrated in Figure 2.5, and include:

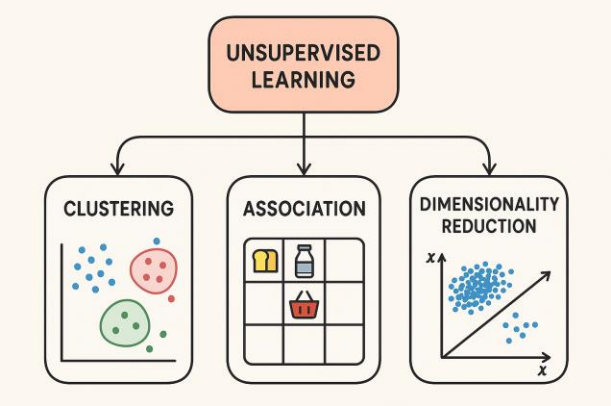



*Figure 2.5 Types of unsupervised machine learning [156]*

### 2.3.2.1. Clustering

Clustering in an unsupervised technique that aims to group unlabeled instances into subsets called clusters based on common similarities or differences. The main goal of a clustering algorithm is to discover the inherent structures and patterns residing in the features of the processed data points via either optimizing a similarity (via minimizing an intra-cluster distance metric) or a dissimilarity measure (through maximizing an inter-cluster distance metric) so that instances of the same cluster are closer to each other, while clusters themselves are as far from each other as possible [152, 153]. Key application areas of clustering algorithms include customer segmentation based on behaviors, image segmentation and computer vision, and anomaly and fraud detection. Common clustering approaches include [154, 155]:

#### A. Hierarchical and connectivity-based clustering

This class of clustering methods constructs a hierarchical organization of data points in the form of a tree structure, known as a dendrogram, by recursively merging or splitting clusters based on a similarity or distance criterion. Two main strategies are employed: (i) agglomerative clustering, which follows a bottom-up approach by successively merging the closest clusters, and (ii) divisive clustering, which adopts a top-down strategy by recursively partitioning the dataset into smaller clusters. These methods do not require a predefined number of clusters and are well suited for uncovering nested or taxonomic data structures; however, they are computationally expensive and sensitive to the choice of linkage criteria and distance metrics.

#### B. Partitioning-based clustering (centroid-based methods)

Partitioning-based clustering algorithms (e.g., k-means, k-medoids) divide the dataset into a predefined number of clusters $K$ by optimizing the positions of representative centroids according to a distance measure (e.g., Euclidean distance). The objective is to minimize intra-cluster variance while maximizing inter-cluster separation. These methods are computationally efficient and

scalable to large datasets, but they assume roughly spherical, non-overlapping clusters and are sensitive to centroid initialization, noise, and outliers.

**C. Distribution-based clustering**

Distribution-based clustering assumes that data points are generated from a mixture of underlying probability distributions (most commonly Gaussian distributions). The clustering task is formulated as a parameter estimation problem, where each cluster corresponds to a probabilistic component. Algorithms such as Gaussian Mixture Models (GMMs) optimized via the Expectation–Maximization (EM) algorithm estimate the parameters of these distributions and assign data points based on likelihood. This paradigm provides a principled statistical framework and supports soft assignments, at the cost of being sensitive to initialization and model assumptions.

**D. Density-based clustering (model-based methods)**

Density-based clustering defines clusters as contiguous regions of high data density that are separated by regions of low density, thus rendering methods deploying it particularly effective at identifying clusters of arbitrary shapes, while being robust to both noise and outliers. Prominent algorithms include DBSCAN and OPTICS, which rely on density thresholds rather than explicit distance minimization. While effective for irregular and overlapping clusters, their performance depends heavily on density-related hyperparameters and may degrade in high-dimensional spaces.

**E. Fuzzy clustering**

Fuzzy clustering extends traditional partitioning methods by allowing data points to belong to multiple clusters simultaneously with varying degrees of membership. Instead of hard assignments, each instance is associated with membership probabilities that sum to one across clusters. Fuzzy C-means is the most widely used algorithm in this category. Fuzzy clustering is particularly useful when cluster boundaries are ambiguous or overlapping, though it is sensitive to noise and requires careful tuning of the fuzzification parameter.

**F. Grid-based clustering**

Grid-based clustering partitions the feature space into a finite number of grid cells and performs clustering by identifying and merging homogeneous neighboring cells. This approach significantly reduces computational complexity by operating on grid structures rather than individual data points. While efficient for large datasets, grid-based methods (e.g., CLIQUE, STING) are sensitive to grid resolution and may struggle with clusters of irregular shapes or varying densities.

#### 2.3.2.2. Association rules

Association rule mining is an unsupervised data mining technique aimed at discovering frequent patterns, correlations, or co-occurrence relationships among variables in unlabeled datasets, where no explicit target variable is defined. These rules are commonly expressed in the form $X \rightarrow Y$, indicating that the occurrence of itemset $X$ is associated with the occurrence of item-

set $Y$ with a certain strength. The primary objective of association rule mining is to uncover meaningful and interpretable relationships within data by evaluating rule significance using measures such as support, confidence, and lift. Popular algorithms for extracting association rules include Apriori, Eclat, and FP-Growth, which efficiently identify frequent item-sets by pruning the search space and exploiting properties such as downward closure. Owing to their transparency and ease of interpretation, association rules are widely adopted in decision-support systems and recommender systems. However, they may generate a large number of redundant or weak rules and can suffer from high computational complexity when applied to large-scale or high-dimensional datasets. A prominent application of association rule mining is market basket analysis, where it is used to identify products that are frequently purchased together, thereby supporting strategies such as cross-selling, recommendation generation, and optimized product placement [157] [158].

#### 2.3.2.3. Dimensionality reduction

Dimensionality reduction is a set of techniques aimed at transforming high-dimensional data into a lower-dimensional representation while preserving its most informative properties, such as variance, intrinsic structure, and salient patterns [159]. This process helps alleviate the curse of dimensionality (i.e., a phenomenon in which model performance degrades as dimensionality increases due to data sparsity, noise amplification, and exponential growth of the feature space) thereby improving computational efficiency, model generalization, and data interpretability. Unlike feature selection, which reduces dimensionality by selecting a subset of existing features (potentially discarding information), dimensionality reduction methods generate new features through transformations of the original variables (e.g., linear or nonlinear projections) that aim to retain as much relevant information as possible. By embedding data into a lower-dimensional manifold, these methods minimize information loss while facilitating tasks such as visualization, clustering, and downstream learning. Dimensionality reduction techniques are widely applied in domains characterized by high-dimensional data, including image processing, genomics, bioinformatics, and natural language processing, where datasets often contain hundreds to thousands of correlated variables and require compact representations for efficient analysis and modeling [160].

#### 2.3.2.4. Generative models

Generative models are a class of algorithms designed to learn the underlying probability distribution of input data in order to generate new, synthetic samples that exhibit similar statistical properties (e.g., images, text, chemical compounds). Unlike discriminative models, which focus on learning the decision boundaries between classes, generative models aim to model how the data is generated by estimating the joint probability distribution $P(X)$ for unlabeled data, or $P(X, Y)$ when labels are available. These models are valuable in a wide range of applications, including data augmentation, image synthesis, anomaly detection, and molecular generation, as they can produce highly realistic and diverse outputs. Despite their benefits, generative models typically require substantial computational resources and careful training to ensure stability and convergence. Prominent generative models include Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) (Figure 2.6) [161] [162].

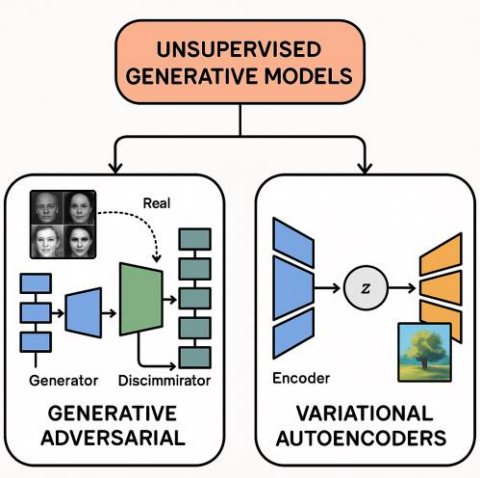



*Figure 2.6 Generative models [156]*

### 2.3.3. Semi-supervised learning

Semi-supervised learning is a ML paradigm that combines the aspects of both supervised and unsupervised learning through leveraging both labeled and unlabeled data to help improve generalization performance [163]. This approach takes advantage of the abundance of easily accessible unlabeled data to reduce reliance on costly labeled data, operating under several assumptions: cluster assumption (data can be partitioned into discrete clusters, and points within the same cluster are likely to share the same label), continuity assumption (points that are close in the feature space are likely to have the same label), low-density assumption (decision boundaries should avoid passing through high-density regions), and manifold assumption (high-dimensional data lies on multiple low-dimensional manifolds, and points on the same manifold are likely to share labels) [164]. Semi-supervised learning has demonstrated significant success in applications such as speech analysis, content annotation on the internet, natural language processing, and protein sequence classification. It employs a variety of techniques including self-training, co-training, graph-based models, and semi-supervised extensions of deep generative models (e.g., semi-supervised VAEs), which reduce annotation costs while maintaining robust predictive accuracy, although their performance remains sensitive to violations of the underlying assumptions, which can lead to the propagation of incorrect pseudo-labels [165].

### 2.3.4. Reinforcement learning

Reinforcement learning is a branch of ML that focuses on training autonomous agents to make sequential decisions through interactions with their environment, with the goal of learning an optimal or near-optimal policy that maximizes cumulative rewards. RL problems are commonly formulated as Markov Decision Processes, which consist of a set of environment and agent states, a set of possible actions for the agent, a transition probability function that defines the likelihood of moving from one state to another given an action, a reward function that assigns values to state-action transitions, and a discount factor that determines the relative importance of future rewards. Reinforcement learning methods such as Q-learning (value-based), REINFORCE (policy-based), and Dyna-Q (model-based), have demonstrated remarkable success in domains such as robotics, autonomous transportation, recommendation systems, and strategic games, showcasing their

ability to adaptively handle non-deterministic environments and complex sequential tasks. Despite these advantages, reinforcement learning remains computationally expensive and faces challenges including reward specification, exploration in high-dimensional spaces, and sample inefficiency [166-168].

## 2.4. Key concepts in machine learning

### 2.4.1. Data

In ML, data refers to a finite set of facts, raw observations, records, or measurements, typically organized into features (variables) and instances (examples), collected from diverse sources to serve as the foundational input for algorithms. Data enables ML models to learn patterns, extract insights, and make predictions, with its quality, quantity, and relevance being pivotal, as they directly influence the deployed model's performance, generalizability, and robustness. ML approaches rely on computational methods to uncover latent structures and relationships within the distribution of features across instances. To support scalability, security, and efficient access, data must be stored in appropriate structures such as databases, data warehouses, or distributed storage systems [169]. Data can take diverse forms (Figure 2.7), which may include:

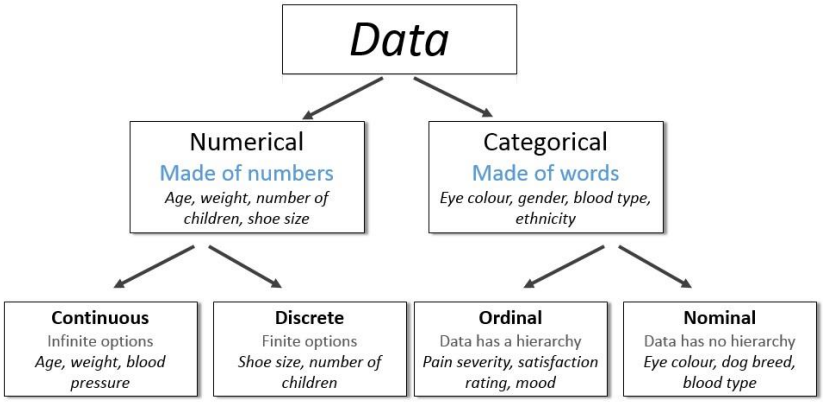



*Figure 2.7 Basic data types [170]*

#### 2.4.1.1. Quantitative data

Quantitative data refers to variables or features that can be measured numerically and subjected to arithmetic operations and statistical analyses. This type of data can be further classified as (i) discrete variables that consist of countable values, typically resulting from counting processes, and take distinct numerical values (e.g., number of products sold, shoe size), and (ii) continuous variables that represent measurable quantities that can assume any value within a specific range (e.g., height, temperature). The numerical nature of quantitative data facilitates trend identification, correlation analysis, and dependency modeling. Additionally, it supports preprocessing techniques such as scaling, normalization, and standardization, which enhance the efficiency of regression, classification, and clustering tasks [140].

#### 2.4.1.2. Qualitative data

Qualitative data, also known as categorical data, represents non-numerical variables describing attributes, labels, or classes that lack inherent mathematical ordering. Qualitative data can be

nominal, where categories cannot be ordered (e.g., gender, country of origin), or ordinal, where categories have a meaningful order (e.g., education level, user satisfaction ratings). Because ML models require numerical inputs, qualitative data must be encoded using methods such as one-hot encoding or label encoding to allow proper model processing [140].

#### 2.4.1.3. Time-series data

Time-series data is a specialized subset of quantitative data, consisting of sequences of observations recorded at successive time intervals (e.g., hourly, daily, monthly). This type of data captures temporal dependencies, trends, and seasonal effects, making it essential for forecasting, anomaly detection, and sequence modeling tasks. Common applications include market trend prediction, stock price estimation, and weather forecasting. Time-series data often requires dedicated models such as autoregressive models, state-space models, and recurrent neural networks to account for temporal correlations [171].

#### 2.4.1.4. Structured data

Structured data consists of information organized in tabular formats or predefined schemas, with rows representing instances and columns representing features. It is typically stored in relational databases or spreadsheets, facilitating efficient querying, retrieval, and analysis. Structured data is widely used in ML due to its organized nature; however, it requires careful handling of missing values, outliers, and class imbalances to ensure robust model performance and generalization [172, 173].

#### 2.4.1.5. Unstructured data

Unstructured data encompasses raw, heterogeneous information lacking a predefined schema or tabular organization, making it incompatible with conventional relational databases. Examples include text (e.g., social media posts, chemical representations), images, audio, and video. Despite its richness, unstructured data requires specialized preprocessing techniques, such as feature extraction, embeddings, and dimensionality reduction, to enable its efficient processing by ML models [172, 173].

#### 2.4.1.6. Semi-structured

Semi-structured data combines elements of structured and unstructured data. It does not fully adhere to rigid schemas but contains organizational cues such as tags, hierarchies, or metadata, which allow partial parsing and feature extraction. Common representations include JSON, XML, and HTML. Semi-structured data is prevalent in web applications, internet-of-things (IoT) systems, and distributed data environments, bridging the gap between fully structured and unstructured formats while supporting flexible analysis [174].

### 2.4.2. Data preprocessing and transformations

In ML, data preprocessing, cleaning, and transformation refer to the systematic operations applied to convert raw data into a refined and consistent format suitable for analysis, while ensuring compliance with the assumptions and requirements imposed by the learning algorithms. These techniques aim to enhance model performance, interpretability, and generalization to unseen data. Data preprocessing and transformation techniques include, but are not limited to:

#### 2.4.2.1. Data cleaning

Data cleaning is the process of detecting, correcting, or removing corrupt, inaccurate, incomplete, or inconsistent data instances to improve overall data quality and reliability. This process typically involves handling missing values (through imputation or deletion), identifying and mitigating outliers using statistical or model-based methods, resolving inconsistencies by standardizing formats or measurement units, and eliminating duplicate records that may arise from human error, faulty sensors, or system integration issues. Data cleaning is essential in ML, as erroneous data propagated through a learning pipeline can introduce bias, generate spurious patterns, and degrade a model's ability to generalize to unseen data. Consequently, effective data cleaning enhances the accuracy, validity, and interpretability of machine learning models [175].

#### 2.4.2.2. Data normalization

Data normalization refers to techniques used to address scale disparities among numerical features by rescaling their values to a common range or distribution, thereby ensuring uniform feature contribution, while reducing the disproportionate influence of large-magnitude values. Common normalization methods include min-max scaling, which rescales values to a fixed range (typically [0,1]), and z-score standardization, which centers data around its mean with unit variance. Normalization improves convergence behavior, reduces numerical instability, and mitigates bias in gradient-based optimization models (e.g., neural networks) and distance-based algorithms (e.g., k-nearest neighbors), thereby enhancing model stability and generalization performance [140].

#### 2.4.2.3. Feature encoding

Feature encoding is the process of transforming non-numerical data, such as categorical variables, class labels, or textual symbols, into numerical representations that can be processed by ML models while preserving relevant semantic relationships. Common encoding techniques include one-hot encoding, which converts categories into binary vectors; label encoding, which assigns unique integer values to categories; ordinal encoding, which preserves inherent category order; and embedding-based methods, which map entities such as words into dense, continuous vector spaces. Appropriate feature encoding is critical to prevent misleading numerical interpretations and ensure effective learning [140, 176].

#### 2.4.2.4. Text-related transformations:

Text-related transformations are systematic preprocessing steps designed to address challenges such as sparsity, polysemy, and syntactic variability in natural language processing (NLP) tasks by converting raw text into structured numerical representations. These transformations commonly involve tokenization (segmenting text into words, sub-words, or characters), normalization (including lowercasing, stemming, or lemmatization), and encoding through suitable vectorization techniques applied in a sequential pipeline. Proper text preprocessing improves representation quality and downstream model performance in NLP-based ML systems [176, 177].

#### 2.4.2.5. Image-related transformations

Image-related transformations consist of mathematical operations applied at the pixel or spatial level to modify raw image data while preserving semantic content relevant to learning tasks. These transformations typically include geometric operations (e.g., rotation, scaling, cropping, and padding), photometric adjustments (e.g., brightness and contrast modification, pixel-value normalization, and histogram equalization), and spatial filtering techniques such as Gaussian smoothing and edge detection. Such operations improve model robustness, invariance, and generalization in computer vision applications [178].

#### 2.4.2.6. Feature discretization

Feature discretization refers to the process of converting continuous numerical variables into discrete categorical intervals (bins) with finite representations. This approach partitions the domain of a continuous feature into non-overlapping intervals; each assigned a nominal or ordinal label. Common discretization methods include equal-width binning, equal-frequency binning, k-means-based discretization, and entropy-based approaches, which can reduce noise, mitigate outlier influence, lower model complexity, and enhance interpretability, albeit at the cost of potential information loss [179][180].

#### 2.4.2.7. Data augmentation

Data augmentation comprises techniques used to artificially expand the size and diversity of training datasets by generating synthetic yet realistic variations of existing samples. These techniques include geometric and photometric transformations in image-based tasks, synonym replacement, back-translation, and random token insertion or deletion in NLP tasks, as well as pitch shifting, time stretching, and background noise addition in audio-related tasks. Data augmentation exposes models to a wider range of input conditions, helping them learn invariance to irrelevant distortions and improving robustness in noisy and diverse environments [181].

#### 2.4.2.8. Data reduction

Data reduction refers to procedures aimed at decreasing data volume, dimensionality, or complexity while preserving essential structural and informational characteristics. These techniques include dimensionality reduction methods (e.g., PCA, t-SNE), feature selection strategies, and sampling techniques, which collectively aim to reduce computational costs, accelerate training, improve inference efficiency, and mitigate the curse of dimensionality. Effective data reduction contributes to improved generalization and scalability of machine learning models [139].

### 2.4.3. Machine learning pipeline and work flow

ML pipeline is the systematic end-to-end workflow that organizes the sequential stages of developing, training, and deploying a ML model, ensuring consistent data flow, reproducibly, scalability, and optimal performance (Figure 2.8) [172]. Formally, a ML pipeline is composed of:

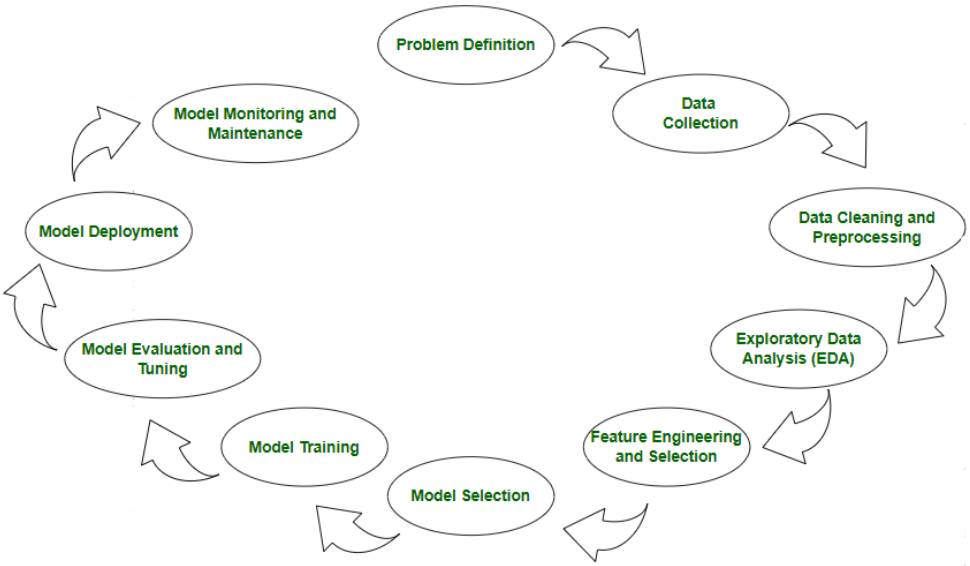



*Figure 2.8 Machine learning pipeline [172]*

### 2.4.3.1. Problem definition

Problem definition is the foundational stage in ML pipeline development that involves the precise formulation of real-world challenges into a well-posed computational problem amenable to algorithmic solutions [182]. This step requires goal identification (what needs to be predicted or inferred), task specification (explicit characterization of the required ML paradigm based on the nature of the available data and desired outputs), constraint analysis (incorporation of domain-specific, operational, and ethical requirements), variable definition (rigorous specification of the input and output spaces), and feasibility assessment (determining data availability, quality, and computational resources needed) [182].

### 2.4.3.2. Data processing

Data processing is a crucial step that comprises a series of procedural transformations converting raw, heterogeneous data into structured representations suitable for computational analysis and model training [172]. This process involves:

- ✓ **Data acquisition:** the systematic collection of relevant raw data from diverse sources (e.g., databases, sensors, clinical records), while ensuring statistical representativeness, measurement reliability, and compliance with ethical and legal guidelines.
- ✓ **Data preprocessing:** as detailed in Section 2.4.2, this step involves applying mathematical and statistical transformations that help handle missing values through imputation or deletion, normalize feature scales, detect and mitigate outliers, and encode categorical variables, in order to produce a clean and structured dataset suitable for analysis and model training.
- ✓ **Exploratory data analysis:** the application of statistical summaries and visualization techniques (e.g., correlation matrices, dimensionality-reduction visualizations such as t-SNE) to examine data distributions, feature relationships, and potential biases that guide subsequent modeling decisions.

- ✓ **Data splitting:** the stratified partitioning of the dataset into mutually exclusive subsets (commonly training, validation, and test sets, e.g., 70%, 10%, 20%), preserving class distributions while preventing information leakage. The validation set supports hyperparameter tuning, while the test set provides an unbiased final evaluation of model performance.

#### 2.4.3.3. Model selection

Model selection is a critical phase in the ML pipeline that involves evaluating and comparing candidate algorithms to identify the most appropriate model architecture for the problem at hand given the available data. This phase requires careful consideration of multiple interdependent factors, including computational constraints (hardware compatibility and resource requirements), model characteristics (complexity, scalability, and interpretability), data-specific properties (feature dimensionality, sample size, and noise), and domain-specific requirements and constraints [183].

#### 2.4.3.4. Hyper parameter tuning

Hyperparameter tuning is a significant optimization process that involves the systematic search for optimal model configurations (that are unlearnable during training) that control its learning dynamics and generalization behavior. This procedure explores hyperparameter spaces (e.g., learning rates, regularization coefficients, or architectural depth) using strategies such as grid search, random search, Bayesian optimization, or evolutionary algorithms, often combined with cross-validation protocols to ensure robust performance estimation, mitigate overfitting or underfitting, balance the bias–variance trade-off, and improve convergence properties [184].

#### 2.4.3.5. Model training

Model training is the fundamental process through which ML algorithms infer underlying patterns and relationships within data distributions by optimizing model parameters to minimize a predefined loss function. In supervised learning settings, models iteratively adjust parameters by comparing predicted outputs with ground-truth labels, while unsupervised approaches optimize similarity measures, likelihood functions, or distance metrics to capture latent data structures. Optimization is commonly achieved using gradient-based methods, such as stochastic gradient descent and its variants, which progressively update model parameters to reduce prediction errors and improve performance [169].

#### 2.4.3.6. Model evaluation

Model evaluation concludes the ML pipeline by verifying whether the trained model satisfies the original problem definition through rigorous assessment of predictive performance, generalization capability, and practical suitability for deployment. Evaluation relies on appropriate quantitative metrics (e.g., balanced accuracy, precision, recall, F1-score, and Cohen's kappa for classification tasks, or mean squared error for regression tasks), which quantify how effectively the model captures underlying data patterns while avoiding overfitting or underfitting. Beyond statistical performance, evaluation should also consider qualitative aspects such as interpretability,

fairness, robustness, and computational efficiency to ensure alignment with practical, ethical, and regulatory requirements [140].

#### 2.4.3.7. Model deployment, monitoring, and maintenance

This phase represents a continuous lifecycle process that ensures sustained model reliability and effectiveness in production environments. Deployment involves integrating trained models into operational systems, with continuous monitoring tracking performance metrics and data characteristics to detect issues such as concept drift, data distribution shifts, or data quality degradation. While maintenance includes periodic retraining, hyperparameter re-optimization, and feature space updates to ensure long-term alignment with evolving data and operational requirements [185].

## 2.5. Common machine learning approaches

### 2.5.1. Supervised machine learning algorithms

#### 2.5.1.1. Linear regression

Linear regression is the simplest regression technique, designed to model the linear relationships between a continuous dependent variable and one independent variables by fitting a linear estimation function that best captures the patterns exhibited by the observed instances (Figure 1.9 (A)), to help predict outputs for unseen data. This method is valued for its computational efficiency, interpretability, and simplicity, making it suitable for preliminary data analysis and inference tasks. However, it relies on strong assumptions (including linearity, independence of errors, homoscedasticity (constant error variance), and normality of residuals) which make it sensitive to outliers and incapable of capturing nonlinear relationships without explicit feature engineering [186, 187].

#### 2.5.1.2. Multiple regression

Multiple regression generalizes simple linear regression by modeling the relationship between a dependent variable and multiple independent variables through a multidimensional hyperplane, enabling the estimation of coefficients that quantify the contribution of each predictor. This method supports the analysis of complex systems where outcomes are influenced by several interacting factors, making it widely applicable in econometrics, healthcare, and engineering. Multiple regression is used both for explanatory analysis through coefficient inference and for predictive modeling via fitted values, but it retains the same core assumptions as simple linear regression, which may limit its applicability when these assumptions are violated [188, 189].

#### 2.5.1.3. Logistic regression

Logistic regression is a statistical method specifically designed for binary classification tasks (despite its name) by modeling the probability of class membership through the application of the logistic (sigmoid) function to a linear combination of input features (Figure 1.9 (B)). It is mainly valued for its interpretability, probabilistic output, and computational efficiency, making it a standard baseline in domains such as medical diagnosis, social sciences, and financial risk assessment. Nevertheless, logistic regression assumes linearity in the log-odds space, limiting its

ability to capture complex nonlinear decision boundaries without feature transformations, and it can suffer from overfitting in high-dimensional settings unless appropriate regularization techniques are applied [186, 190].

#### 2.5.1.4. Polynomial regression

Polynomial regression extends linear regression by modeling the relationship between dependent and independent variables as an $n^{th}$-degree polynomial, allowing nonlinear relationships to be captured within a linear-in-parameters framework (Figure 1.9 (C)). While it retains interpretability for low polynomial degrees while providing flexibility in modeling curvilinear trends; the use of higher-degree polynomials significantly increases the risk of overfitting, reduces interpretability, and compromises generalization performance. Consequently, polynomial regression is commonly applied in domains such as economics (growth trend modeling), engineering (sensor calibration), and biology (dose-response analysis), where nonlinear yet smooth relationships are expected [186, 187].

#### 2.5.1.5. Redge regression (Tikhonov regularization)

Ridge regression is a regularized extension of ordinary least squares regression that addresses multicollinearity and overfitting by incorporating an $L_2$-norm penalty term into the loss function. This penalty constrains the magnitude of regression coefficients, reducing variance and improving numerical stability without eliminating features entirely. As a result, ridge regression yields more stable and generalizable models, particularly in high-dimensional settings where predictors are highly correlated. It is widely employed in genomic analysis, econometric modeling, and large-scale ML systems as a foundational regularization technique for enhancing predictive robustness [191].

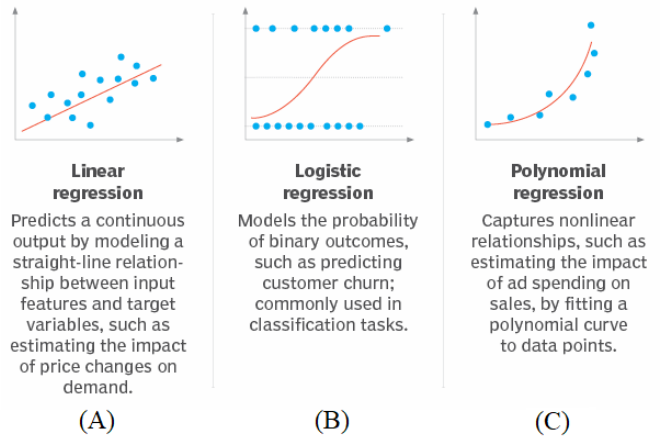



*Figure 2.9 Types of regression in machine learning [193]*

#### 2.5.1.6. Lasso regression (Least Absolute Shrinkage and Selection Operator)

Lasso regression is an advanced regularization technique that enhances both the accuracy and interpretability of ordinary least squares regression by adding an $L_1$-norm penalty term to the loss function. This formulation not only controls model complexity but also promotes sparsity by shrinking the coefficients of less relevant features exactly to zero, thereby performing implicit feature selection. Lasso regression is particularly effective in high-dimensional settings, where it

facilitates the identification of the most informative predictors while mitigating overfitting and multicollinearity, making it widely used in computational biology, finance, and production ML pipelines that require both interpretability and strong predictive performance [192].

#### 2.5.1.7. Support vector machines (SVM)

SVMs are kernel-based supervised learning models widely used for classification, regression, and anomaly detection tasks (Figure 2.10). Their core objective is to construct an optimal separating hyperplane that maximizes the margin between classes, defined as the minimum distance between the decision boundary and the closest training samples from each class, known as support vectors. Margin maximization enhances robustness to noise and improves generalization to unseen data.

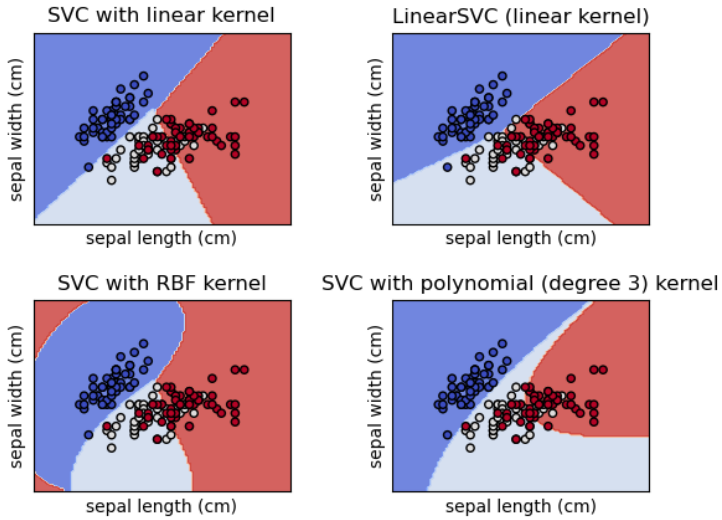



*Figure 2.10 The application of SVMs on a classification task using different types of kernels [196]*

SVMs offer several advantages that make them effective across a wide range of applications. Their strong performance in high-dimensional spaces enables them to handle problems where the number of features exceeds the number of samples, a scenario common in domains such as bioinformatics and text mining. Overfitting is mitigated through margin maximization combined with explicit regularization. Furthermore, SVMs can model nonlinear decision boundaries by employing kernel functions (e.g., radial basis function, polynomial kernels), which implicitly map input data into higher-dimensional feature spaces without requiring explicit transformations. Decision functions depend only on a subset of training samples (support vectors), which contributes to efficient inference once the model is trained. These properties have led to the widespread adoption of SVMs in both academic research and industrial settings, including bioinformatics (gene expression analysis, disease diagnosis), NLP tasks (text classification, named entity recognition), and remote sensing and geoscience (land cover classification). Despite their strengths, SVMs exhibit notable limitations. Training complexity scales super-linearly with the number of samples, making them computationally expensive and impractical for very large-scale datasets. Model performance is highly sensitive to kernel choice and hyperparameter tuning, which often requires extensive cross-validation. Additionally, the use of kernel methods reduces interpretability, as feature contributions are not directly observable. Although multi-class

extensions exist (e.g., one-vs-one and one-vs-rest strategies), SVMs are inherently binary classifiers, and performance may degrade as the number of classes increases. These constraints limit their applicability in real-time systems, large-scale data environments, and scenarios requiring high model transparency or complex multi-class discrimination [194, 195].

#### 2.5.1.8. Decision trees

Decision trees are a supervised learning approach that constructs a hierarchical, tree-structured model through recursive partitioning of the feature space into increasingly homogeneous subsets, supporting both classification and regression tasks (Figure 2.11). At each internal node, a splitting criterion (e.g. Gini impurity or entropy for classification, and variance reduction for regression) is used to select the feature and threshold that best separates the data. This recursive process continues until predefined stopping conditions are met (e.g., maximum tree depth, minimum samples per leaf, or node purity), yielding a model where predictions correspond to interpretable sequences of logical if-then rules [197-199].

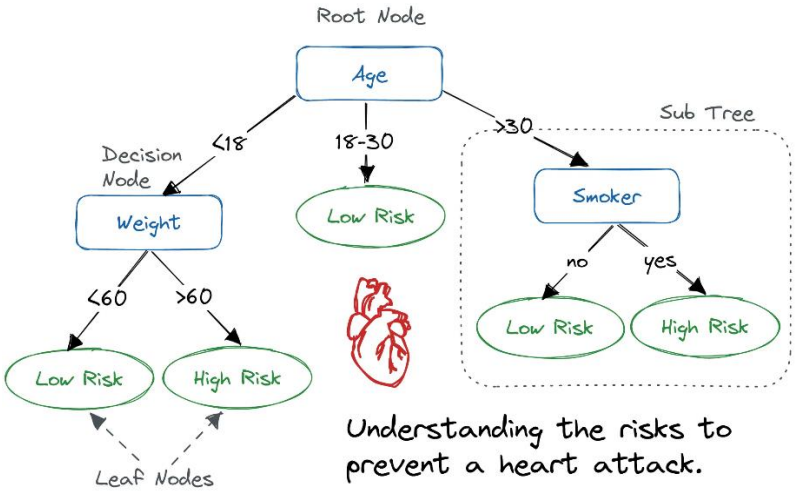



*Figure 2.11 Decision tree model architecture [199]*

Decision trees offer several advantages that explain their popularity in ML, ranging from their transparent structure that enables high interpretability, as decision paths can be directly traced and understood by humans, which makes them particularly valuable in domains requiring explainable models. They can naturally handle both numerical and categorical features, are relatively robust to outliers and missing values, and require minimal data preprocessing. Moreover, their non-parametric nature eliminates assumptions about underlying data distributions, allowing flexible modeling of nonlinear relationships and feature interactions. Decision trees are also computationally efficient for training and inference on moderately sized datasets [197]. These strengths have led to widespread adoption across diverse domains, including healthcare (clinical decision support and patient outcome prediction), finance (credit risk assessment, loan approval, and fraud detection), marketing (customer segmentation and purchasing behavior prediction), and bioinformatics (gene expression analysis and disease subtype classification). However, decision trees also suffer from significant drawbacks, as they are highly prone to overfitting, particularly when allowed to grow deep without constraints, which necessitates regularization strategies such as pruning, depth limitation, or minimum leaf size enforcement. Their greedy, locally optimal

splitting strategy may lead to suboptimal global solutions that fail to capture the true data-generating process. Additionally, decision trees are sensitive to small variations in training data, often resulting in substantially different tree structures, which reduces model stability. As tree complexity increases, interpretability can deteriorate, making deep trees difficult to analyze and explain [197, 198].

### 2.5.1.9. K-Nearest neighbor (k-NN)

This non-parametric, instance-based supervised learning paradigm can be employed for both classification and regression tasks, as it operates under the assumption that similar individuals exhibit homogeneous target values that can be approximated via distance-based techniques. For a given query point, this algorithm identifies the k-closest training samples using a distance metric (e.g., Euclidean, Manhattan distance), subsequently aggregating their labels through either majority voting (classification) or averaging (regression) as depicted in Figure 2.12. Unlike parametric models, K-NN avoids explicit training by retaining the training set and delaying all computations to the inference phase, resulting in a memory-based learning approach. While k-NN demonstrates numerous advantages that include its intuitive implementation and natural adaptability to multi-class problems, which lead to its successful deployments in domains such as recommender systems, medical diagnostics, and image retrieval; its practical utility is constrained by its computational inefficiency when dealing with large datasets, which is paired with its sensitivity to feature relevance, and diminished performance in high-dimensional spaces [200, 201].

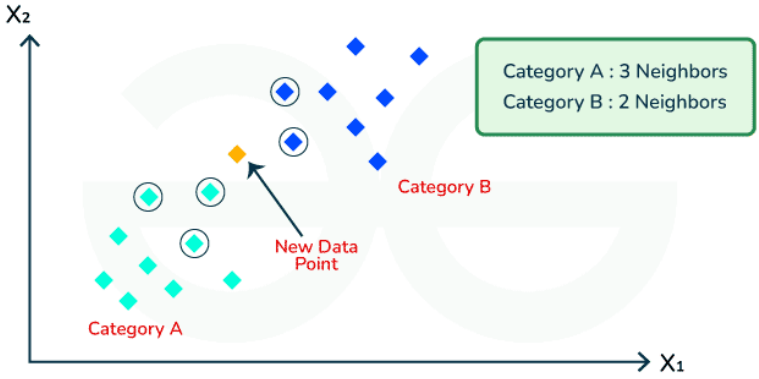



*Figure 2.12 K-NN approach in assigning labels to new instances [204]*

### 2.5.1.10. Naïve bayes classifier

Naïve Bayes classifiers constitute a family of probabilistic supervised learning models grounded in Bayes' theorem and the conditional independence assumption (the naïve assumption), which posits that all input features are mutually independent given the class label. The primary objective of this approach is to estimate the posterior probability of a class given an observation by efficiently modeling the joint probability distribution of features and class labels (Figure 2.13). During training, the algorithm estimates prior class probabilities from the data and computes class-conditional likelihoods using suitable distributional assumptions, such as Gaussian distributions for continuous features or multinomial and Bernoulli distributions for discrete features. Classification is then performed by selecting the class that maximizes the posterior probability for

a new instance. Despite the fact that the conditional independence assumption is often violated in real-world data, Naïve Bayes classifiers remain highly effective in practice, particularly in high-dimensional and sparse settings such as text classification (e.g., spam filtering and sentiment analysis), medical diagnosis, and recommendation systems, where feature dependencies tend to have limited impact on predictive performance. Their advantages include computational efficiency, scalability, robustness to irrelevant features, and minimal training data requirements. However, their performance can deteriorate when strong correlations exist among features or when continuous data significantly deviates from the assumed probability distributions, which may lead to suboptimal probability estimates and reduced classification accuracy [202, 203].

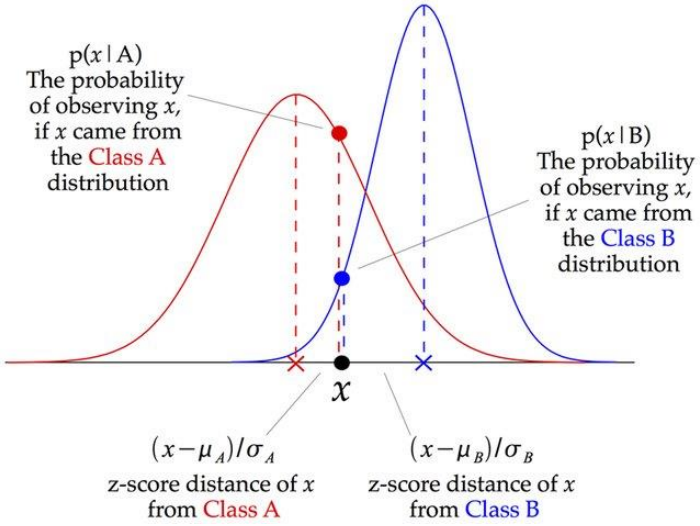



*Figure 2.13 The classification process of a Gaussian-based naïve bayes classifier [205]*

### 2.5.2. Unsupervised machine learning algorithms

#### 2.5.2.1. Principal component analysis (PCA)

PCA is a linear unsupervised dimensionality reduction technique that transforms a set of correlated variables into a new set of uncorrelated (orthogonal) variables called principal components, which are ordered according to the amount of variance they explain in the original dataset. Its objective is to identify orthogonal projection axes (Figure 2.14) that maximize data variance by leveraging eigenvalue decomposition of the covariance matrix or singular value decomposition of the centered data matrix. PCA is favored for noise reduction and feature decorrelation, but it is limited to capturing linear relationships and is therefore ineffective for modeling nonlinear data structures [206].

#### 2.5.2.2. t-Distributed Stochastic Neighbor Embedding (t-SNE)

Similar to PCA, t-SNE is a nonlinear unsupervised dimensionality reduction technique that embeds high-dimensional data into a low-dimensional space, t-SNE accomplished this by converting pairwise similarities into probability distributions and minimizing the Kullback-Leibler divergence between the high and low-dimensional distributions using gradient-based optimization (Figure 2.15). t-SNE is particularly valued for its ability to preserve local neighborhood structures

and reveal visually separable clusters, despite its high computational cost, lack of scalability, and strong sensitivity to hyperparameters such as perplexity and learning rate [208].

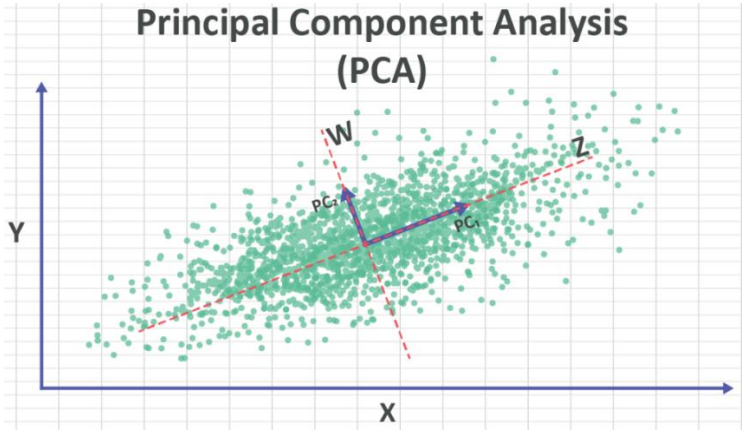



*Figure 2.14 the application of PCA on a multivariate Gaussian distribution [207]*

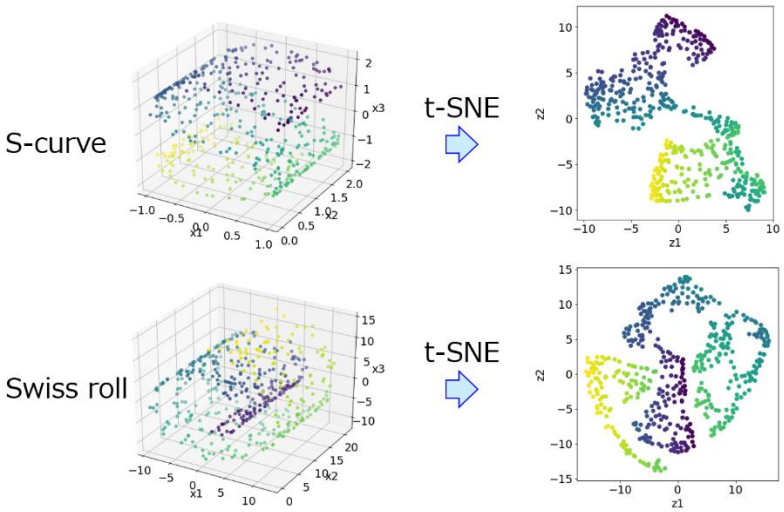



*Figure 2.15 The application of t-SNE on a S-curve and swiss-roll shaped data distributions [209]*

### 2.5.2.3. Generative Adversarial Networks (GANs)

GANs are neural network architectures that consist of two neural components trained simultaneously in opposition: a generator that learns to synthesize realistic samples from random noise, and a discriminator that learns to distinguish between real training samples and generated ones (Figure 2.16). Training is formulated as a minimax optimization problem in which the generator seeks to maximize the discriminator's classification error while the discriminator aims to minimize it, ideally converging to an equilibrium where generated samples are indistinguishable from real data under the discriminator's criteria [161, 162].

### 2.5.2.4. Variational Autoencoders (VAEs)

VAEs are a class of probabilistic generative models that combine autoencoder architectures with variational Bayesian inference to learn continuous latent representations of input data and generate new samples with similar statistical properties as depicted in Figure 2.17. This learning

process is achieved by maximizing the Evidence Lower Bound (ELBO), which jointly enforces reconstruction fidelity and regularization of the latent space toward a predefined prior distribution, typically a standard normal distribution. VAEs produce smooth and structured latent spaces suitable for interpolation and sampling, but often generate blurry outputs due to likelihood-based optimization and restrictive distributional assumptions [211, 212].

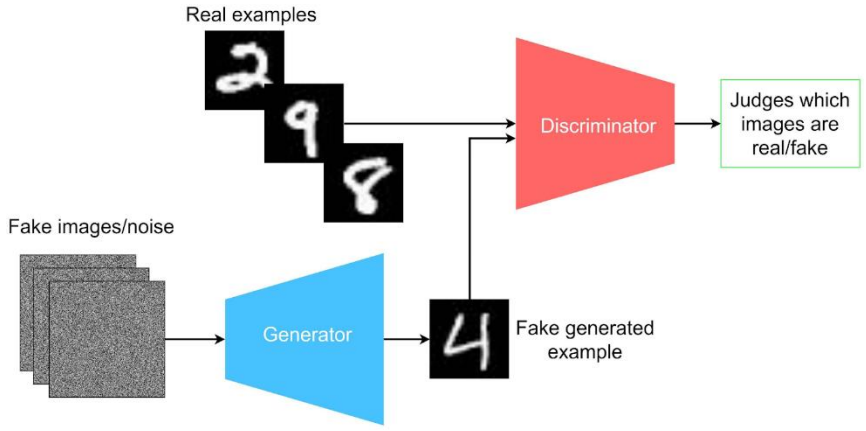



*Figure 2.16 The architecture of a GAN [210]*

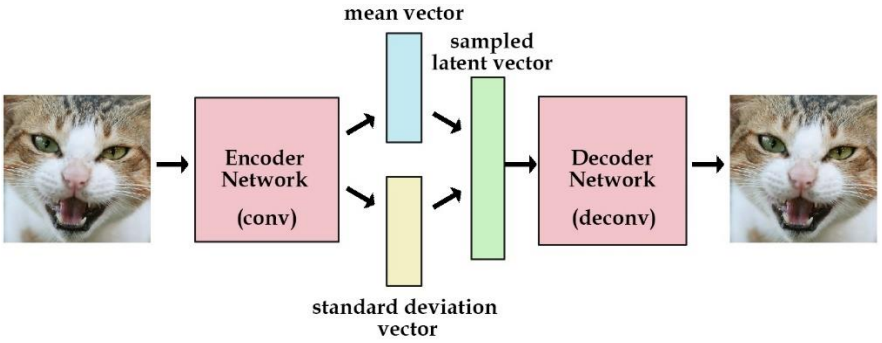



*Figure 2.17 The architecture of a convolutional VEA [213]*

### 2.5.2.5. K-means

A centroid-based unsupervised clustering algorithm designed to partition a dataset into $k$ mutually exclusive clusters by iteratively minimizing the within-cluster sum of squared distances (Figure 2.18). The algorithm alternates between assigning each data point to its nearest centroid using Euclidean distance and updating centroids as the mean of assigned points, converging when centroid positions stabilize or a stopping criterion is met. Despite its efficiency, simplicity, and widespread use, K-means is sensitive to initialization, requires prior specification of $k$, assumes spherical clusters of similar density, and is vulnerable to feature scaling issues and outliers [214, 215].

### 2.5.2.6. Apriori algorithm

A foundational association-rule mining technique designed to identify frequent item-sets and derive association rules from transactional datasets (Figure 2.19). This algorithm leverages the

downward-closure property (all subsets of a frequent item-set must also be frequent) to systematically explore the item-set space in a breadth-first manner, operating through iterative cycles of support counting to evaluate item-set frequency against a minimum threshold and candidate generation via self-joins of frequent $(k-1)$-itemsets followed by a pruning operation. This approach enables the discovery of all frequent item-sets without exhaustive enumeration, thereby facilitating the extraction of association rules that satisfy minimum support and confidence constraints. While Apriori's systematic pruning strategy and interpretability have enabled its widespread use in market basket analysis (supermarket transaction patterns), healthcare (patient diagnosis co-occurrences), and web usage mining (clickstream behavior), it suffers from computational inefficiencies on large or dense datasets due to repeated database scans and exponential growth of candidate item-sets as support thresholds decrease [217, 218].

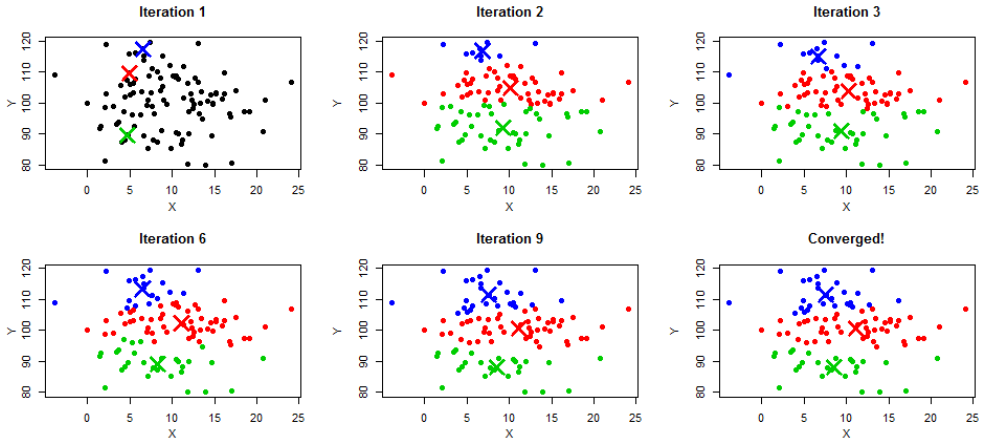



*Figure 2.18 The convergence process of K-means towards optimal clusters [216]*

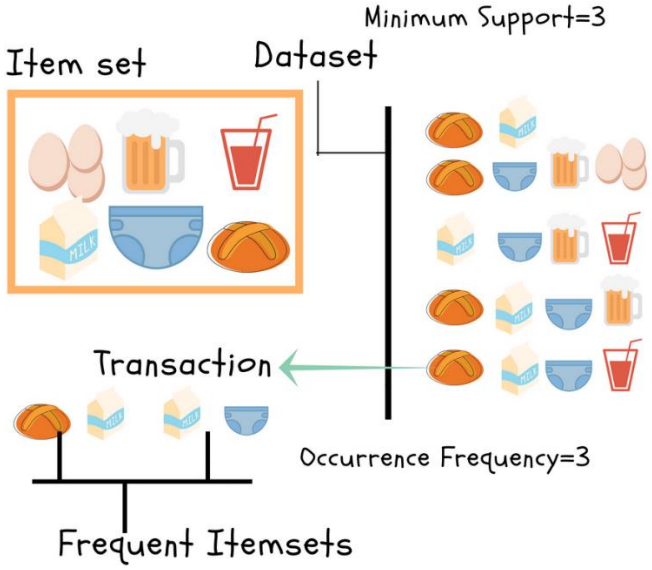



*Figure 2.19 The process of extracting frequent items via apriori algorithm [219]*

### 2.5.3. Deep Neural network (DNN) based algorithms:

A class of ML models that process data through hierarchical layers of specialized computational nodes (MLP, CNN, RNN, attention, etc.), each designed to capture the distinct

patterns of a specific data type (tabular, time-series, textual, image, etc.). The general framework of DNNs consists of an input layer, where each node represents a data instance or one of its features, multiple hidden layers that implement domain-specific transformations (e.g., convolutional operations for spatial data, recurrent connections for sequences, attention mechanisms for contextual relationships) for more efficient feature extraction process, and an output layer that has node(s) that correspond to class probabilities of regression values. Such models are specifically designed to approximate complex functions that enables mapping input features into corresponding output values through the composition of simpler non-linear transformations. More specifically, the training process of a DNN involves minimizing a differentiable loss function that quantifies the error between predicted and actual outputs via backpropagation, where gradients are computed through the chain rule, and model's learnable weights are updated iteratively using a gradient-based optimization technique (e.g., stochastic gradient descent, Adam), enabling robust predictions, pattern recognition, and efficient function approximation in various domains [220, 221]. Typical DNNs architectures include:

#### 2.5.3.1. Multi-layer perceptron (MLP)

This subset of DNNs is one of the most common architectures in ML, inspired by biological neural networks and characterized by a feedforward structure in which information flows unidirectionally from the input layer to the output layer. Its architecture consists of multiple interconnected neurons (Figure 2.20) organized into fully connected (dense) sequential layers as depicted in Figure 2.21. MLPs are often considered relatively shallow architectures, typically comprising one to three hidden layers, and are commonly applied to tasks involving structured tabular data with numerical and/or categorical features. The representational power of MLPs arises from the collective behavior of neurons with learnable weights and biases combined with nonlinear activation functions (e.g., ReLU, sigmoid), enabling the model to learn complex nonlinear mappings between inputs and outputs by applying the activation function to the weighted sum of inputs at each neuron [141, 222].

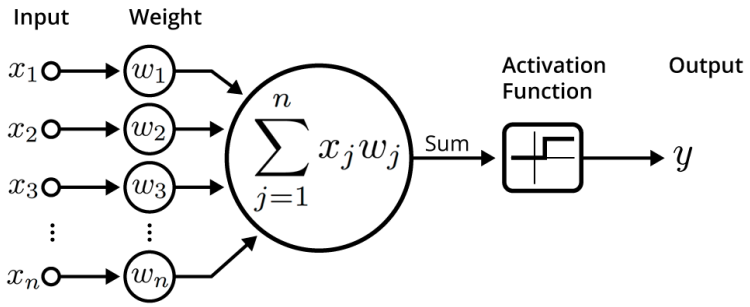



*Figure 2.20 The architecture of a neuron [141]*

MLPs offer notable advantages, including their universal approximation property, which allows them to model complex nonlinear relationships given sufficient capacity and appropriate activation functions. They can generalize to unseen data when properly trained and regularized, and can be adapted to diverse tasks through configurable architectures (e.g., number of hidden layers and units). Moreover, they benefit from well-established optimization algorithms, regularization strategies, and weight initialization techniques. However, they suffer from important limitations,

as their high parametric complexity often requires large labeled datasets for effective training, and increasing architectural depth leads to higher computational cost and greater susceptibility to vanishing gradients and optimization difficulties, often associated with saddle points rather than true local optima. Additionally, their black-box nature limits interpretability, their performance is highly sensitive to hyperparameter choices (e.g., learning rate and initialization strategy), and they lack inductive biases for efficiently handling image, sequential, and textual data compared to specialized architectures such as CNNs and sequence models [224, 225].

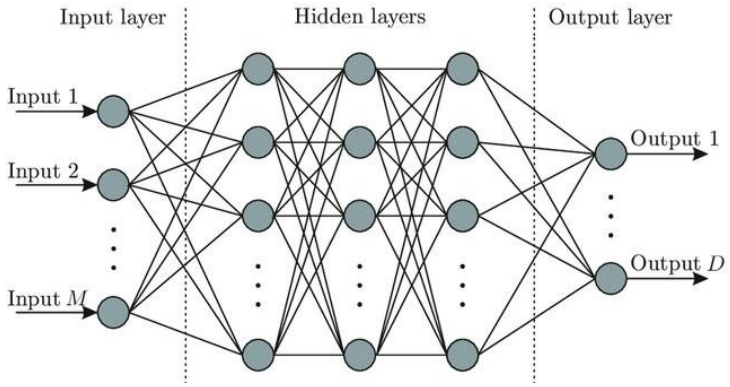



*Figure 2.21 The architecture of a MLP [223]*

### 2.5.3.2. Convolutional neural network (CNN)

This class of DNNs is specifically designed to process grid-structured data (e.g., images) by exploiting spatial hierarchies through a multi-layered framework of convolutional layers, often interleaved with pooling operations and typically followed by fully connected layers, as illustrated in Figure 2.22 [226].

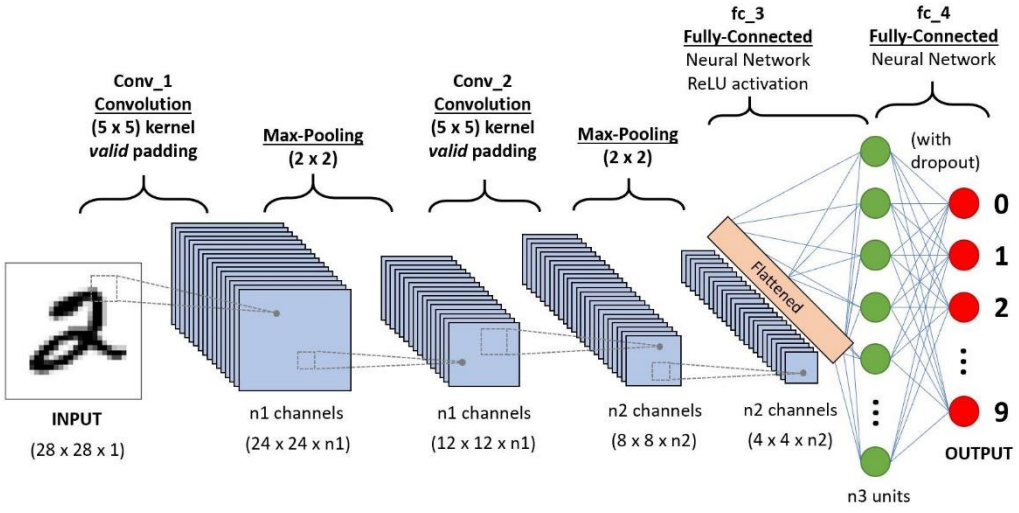



*Figure 2.22 The architecture of a CNN [229]*

To effectively extract latent features from images, convolutional layers slide small, learnable filters (kernels) across the input, where each filter performs an element-wise multiplication with a local receptive field, sums the results, and adds a bias term to produce an activation (Figure 2.23). Applying this operation across all spatial locations yields a feature map that emphasizes salient

local patterns while preserving spatial structure, and using multiple filters enables the capture of diverse features (e.g., horizontal versus vertical edges). Pooling layers, on the other hand, apply fixed-size windows over feature maps to aggregate local responses (e.g., via average or max pooling), thereby reducing spatial resolution and improving robustness to small translations [227]. The final fully connected layers integrate the learned spatial features to perform tasks such as classification or regression. The objective of CNNs is to learn representations that are translation-equivariant in early layers and increasingly invariant at higher levels, enabling the automatic extraction of low- and high-level features with minimal manual feature engineering through end-to-end training that optimizes filter parameters via backpropagation and gradient-based loss minimization [228].

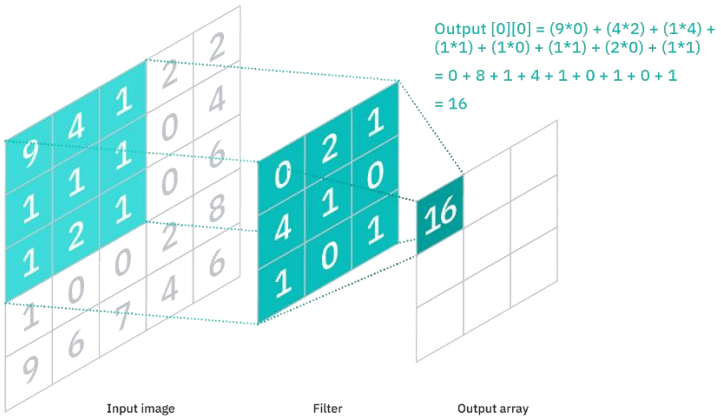



*Figure 2.23 The mathematical application of the convolutional operation [230]*

CNNs are highly regarded for their ability to automatically learn hierarchical feature representations directly from raw input data, reducing the need for manual feature engineering and often improving generalization in image and spatial data tasks. Their weight-sharing mechanism substantially reduces the number of trainable parameters compared to fully connected networks, leading to improved computational and statistical efficiency. Additionally, their local connectivity and translation equivariance, combined with pooling-induced invariance, enable robust pattern detection across spatial locations, making them fundamental to object recognition and detection tasks. However, CNNs typically require large labeled datasets and considerable computational resources for effective training, and their localized receptive fields can limit the efficient modeling of long-range or global contextual dependencies, often motivating the integration of supplementary mechanisms such as attention. Furthermore, CNNs are primarily designed for grid-structured data (e.g., images) and cannot natively operate on non-Euclidean domains (e.g., graphs or manifolds) without architectural extensions or specialized variants [226, 231].

#### 2.5.3.3. Recurrent neural network (RNN)

This type of DNNs is specifically designed to process sequential data (e.g., time series, text, and speech) by modeling temporal dependencies through a recursive architecture (Figure 2.24) that preserves contextual information across time steps via hidden states. At each time step, a recurrent unit computes its hidden state by applying a nonlinear transformation (e.g., tanh or ReLU) to the weighted sum of the current input and the previous hidden state, with the initial hidden state typically initialized as a zero vector or a learned parameter, while sharing weights

across all time steps. This sequential processing allows the network to produce a sequence of hidden states that encode the temporal structure of the input data, enabling subsequent output layers (often fully connected layers) to exploit this information for tasks such as sequence prediction or classification. The primary objective of RNNs is to learn time-dependent representations that capture short-term dependencies and, in principle, long-range dependencies without manual feature engineering, through a training process that employs backpropagation through time (BPTT) and gradient-based optimization to minimize a predefined sequence-level loss function (e.g., cross-entropy) [232, 233]. RNNs offer significant advantages for processing variable-length sequential data through their time-dependent hidden states, which retain contextual information and make them well-suited for tasks such as time-series forecasting, speech recognition, and NLP. Their weight-sharing mechanism across time steps allows efficient modeling of sequential patterns while maintaining a relatively compact parameter space compared to fully connected feedforward networks. However, RNNs are constrained by vanishing and exploding gradients during backpropagation through time (BPTT), which can severely limit their ability to capture long-term dependencies. Their inherently sequential computations hinder parallelization, increasing computational cost and slowing both training and inference. Additionally, RNNs may exhibit instability and require careful hyperparameter tuning, as their autoregressive processing can lead to error accumulation over long sequences [233, 234].

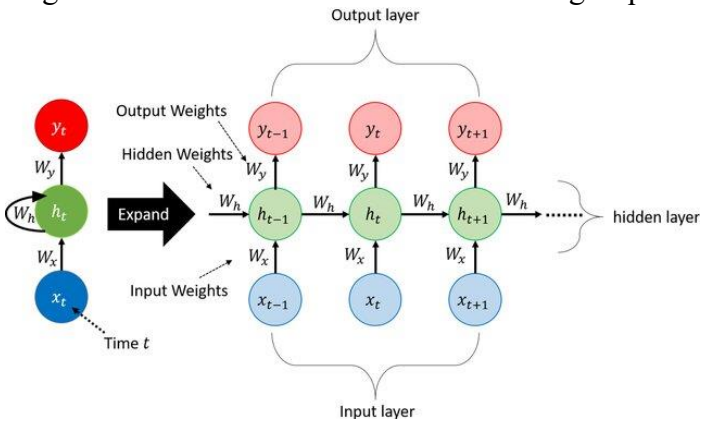



*Figure 2.24 The architecture of a recurrent neural network [235]*

### 2.5.3.4. Gated-recurrent units (GRUs)

This variant of recurrent units is specifically designed to overcome the limitations of standard RNNs by adaptively capturing long-range dependencies in sequential data through gating mechanisms that regulate the flow of information across time steps as illustrated in Figure 2.25. A GRU cell employs two gates: a reset gate, which determines how much of the previous hidden state should be ignored when computing the new candidate activation (allowing the model to discard irrelevant historical information), and an update gate, which controls the trade-off between retaining the previous hidden state and incorporating the new candidate activation (enabling the network to preserve essential long-term dependencies). This gating mechanism allows GRUs to learn compact, informative representations of sequential data, mitigating the vanishing gradient problem while remaining computationally more efficient than standard RNNs or Long Short-Term Memory (LSTM) networks [236, 237]. One of the primary strengths of GRUs is their computational efficiency, as they use fewer parameters than LSTMs, making them faster to train

and less prone to overfitting. GRUs can effectively capture both short-term and long-term dependencies in sequential data through their gating mechanisms, often achieving comparable or superior performance to LSTMs despite their simpler architecture. Additionally, they are easier to implement and require less hyperparameter tuning, making them suitable for rapid experimentation and deployment. However, their reduced gating complexity can limit performance on tasks that demand very long-term memory retention, and they may still struggle with extremely long sequences due to residual gradient attenuation (i.e., gradual weakening of gradient signals during BPTT). Furthermore, while GRUs are generally easier to train than LSTMs, their relative performance is task-dependent, and they lack the fine-grained control over memory contents that LSTMs provide [236, 237].

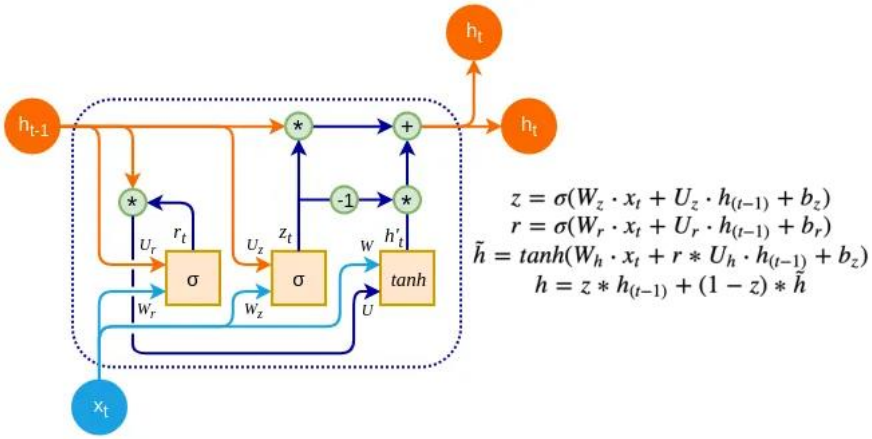



*Figure 2.25 The architrecture of GRU [238]*

### 2.5.3.5. Transformer architecture

This DNN framework is designed for sequence modeling and translation tasks, relying entirely on self-attention mechanisms that allow processing sequential data without recurrent or convolutional operations. Its architecture consists of stacked encoder and decoder layers (Figure 2.26), each containing a multi-head self-attention mechanism, a position-wise feed-forward network, and a normalization layer, with the encoder transforming input sequences into continuous, high-dimensional representations, while the decoder generates output sequences by attending to both the encoder's outputs and its own previous predictions in an autoregressive manner. To compensate for the absence of inherent sequential processing, transformers incorporate positional encodings that inject information about the relative or absolute position of tokens within the input sequence. Transformers are highly regarded for their ability to model long-range dependencies efficiently through attention mechanisms, enabling each token to directly interact with all other tokens in the sequence, which supports parallelizable training and enhances contextual representation learning [239]. Transformers offer several key advantages over traditional sequence modeling approaches, including superior parallelization during training due to their non-recurrent design, the ability to capture long-range dependencies through global self-attention mechanisms, and strong performance across diverse domains such as NLP and computer vision. Moreover, their scalable architecture has enabled the development of increasingly large language models, and their attention mechanisms provide more interpretable and analyzable representations of model behavior. However, transformers also have notable limitations, as their

memory and computational requirements are high, with self-attention operations scaling quadratically with sequence length, and they typically require very large training datasets to achieve optimal performance. Additionally, standard transformer architectures lack innate mechanisms for modeling hierarchical or structured data patterns, which are more naturally captured by recurrent or convolutional operations [240, 241].

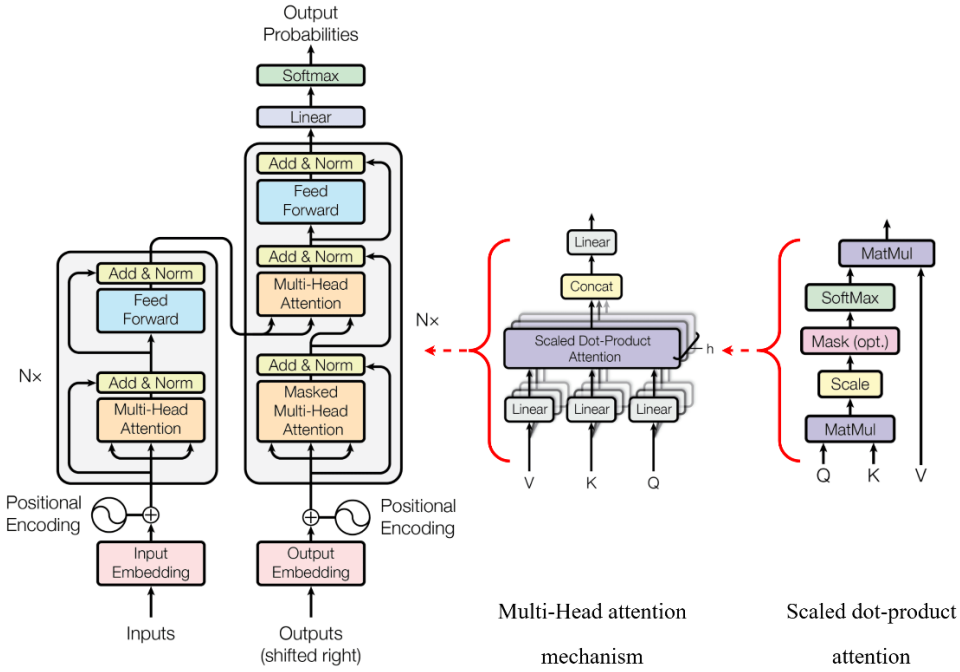



*Figure 2.26 The architecture of a transfomer and its components [239]*

### 2.5.3.6. Autoencoders (AE)

This type of DNNs is designed to learn efficient low-dimensional representations of data by capturing its most salient features through an unsupervised learning paradigm. The architecture of an AE consists of two main components: an encoder, which maps high-dimensional input data into a compressed latent representation via a series of nonlinear transformations, and a decoder, which reconstructs the original input from this latent representation (Figure 2.27). This process is achieved by optimizing the network parameters to minimize a reconstruction loss between the original input and its reconstruction, thereby forcing the AE to retain essential features while discarding irrelevant variations, and encouraging the network to uncover the underlying manifold structure of the input space, which enables tasks such as dimensionality reduction, denoising, and unsupervised feature extraction [242, 243]. AEs offer several key advantages, particularly their ability for unsupervised feature learning, as they can automatically extract meaningful representations from raw input data without requiring labeled examples. They are effective for dimensionality reduction, denoising, and anomaly detection because they encode input data into compact latent spaces that preserve essential structural information. Furthermore, AEs provide architectural flexibility, allowing extensions such as convolutional autoencoders for spatial data and variational autoencoders for probabilistic modeling. However, their limitations include susceptibility to overfitting and memorization of the training data rather than learning generalizable patterns, especially when model complexity is not properly regularized. Their

performance is highly sensitive to architectural design, hyperparameter settings, and the amount of training data. Additionally, features learned in an unsupervised manner may lack interpretability or discriminative power for downstream tasks, and standard AEs may struggle to capture complex hierarchical relationships in data without advanced architectural modifications [243, 245].

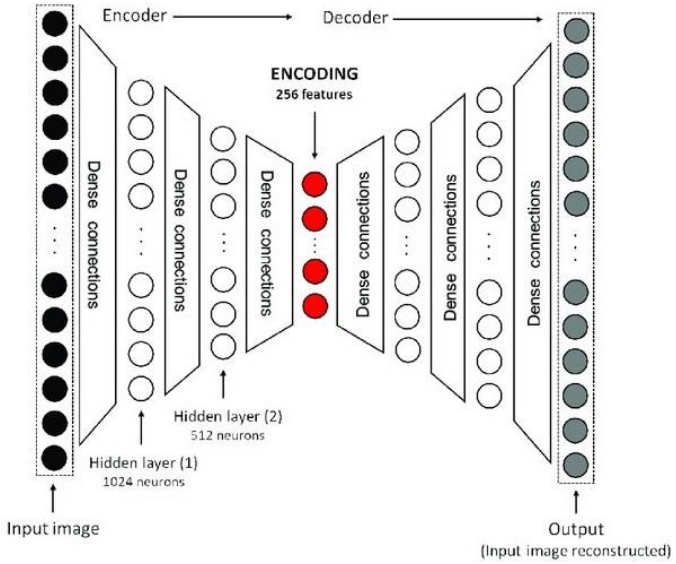



*Figure 2.27 The architecture of an autoencoder [244]*

### 2.5.4. Ensemble methods

#### 2.5.4.1. Random forest (RF)

This ensemble method constructs multiple decision trees using bootstrap aggregation (bagging) and random feature selection during node splitting, where each tree is trained on a distinct bootstrap sample of the original dataset, and splits are based on randomly selected feature subsets that de-correlate individual trees [246], as depicted in Figure 2.28. The overall predictions of a RF are obtained via majority voting for classification or averaging for regression, enhancing predictive accuracy while reducing overfitting. As a non-parametric approach, RFs can model complex nonlinear relationships without assuming any specific data distribution, and they are particularly effective in high-dimensional settings. Their computational efficiency and inherent ability to handle mixed data types (continuous and categorical) enable applications in fields such as bioinformatics, financial analysis, remote sensing, and clinical decision support, although they are less interpretable than single decision trees, computationally demanding for large ensembles, and limited in extrapolating beyond observed data ranges in regression tasks [247, 248].

#### 2.5.4.2. Bootstrap aggregating (bagging)

This ensemble method improves model stability and predictive accuracy by combining multiple base learners trained on diverse bootstrap samples drawn from the dataset. The algorithm generates bootstrap replicates of the training data via sampling with replacement, trains a base learner (e.g., decision tree) on each replicate, and aggregates predictions using majority voting for classification or arithmetic averaging for regression to produce the final output. By leveraging the variance-reduction properties of bootstrap sampling and model averaging, bagging mitigates

overfitting while maintaining the bias characteristics of the base learners, enabling successful applications in financial risk assessment (e.g., credit scoring ensembles), ecological modeling (e.g., habitat suitability prediction), and biomedical classification (e.g., disease subtype identification from omics data). However, the method has limitations, including computational cost that scales linearly with ensemble size, reduced interpretability due to the aggregated ensemble structure, and limited performance improvement when applied to inherently low-variance models such as linear regression. Additionally, its effectiveness depends critically on the diversity and robustness of the constituent base learners [250, 251].

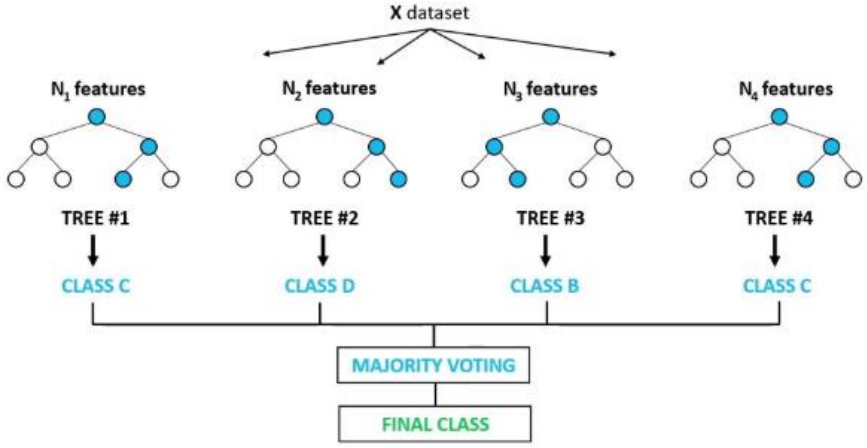



*Figure 2.28 Random forest classifier [249]*

#### 2.5.4.3. Boosting

This ensemble learning technique builds a high-accuracy predictive model by iteratively combining weak base learners (typically high-bias, low-variance models) with an emphasis on correcting errors made by previous iterations. The algorithm trains each successive learner on a reweighted version of the training data that gives higher importance to previously mispredicted instances, and aggregates predictions using weighted majority voting for classification or additive combination for regression. This sequential approach enhances predictive performance by reducing bias while controlling variance and can provide implicit feature selection, supporting successful applications in medical diagnostics (e.g., early disease risk stratification from multimodal clinical data), computer vision (e.g., real-time pedestrian detection), and information retrieval (e.g., search engine relevance ranking). Nevertheless, boosting is sensitive to noise and outliers due to its focus on difficult-to-classify samples, requires significant computational resources for sequential training, and reduces interpretability, despite offering feature importance measures [252, 253].

## 2.6. Applications of machine learning

### 2.6.1. Computer vision

ML algorithms (particularly DNN architectures) have profoundly advanced the field of computer vision by enabling the autonomous extraction and interpretation of complex visual patterns from large-scale image and video data, eliminating the need for manual feature engineering, and improving adaptability across a wide range of real-world visual perception tasks.

Particularly, CNNs have become foundational image classification, object detection, and semantic segmentation tasks, owing to their ability to learn spatial hierarchies of features directly from raw pixels, while other architectures like Residual Networks (ResNets), U-Nets, and Vision Transformers have further elevated the performance and scalability of vision systems by addressing challenges encompassing vanishing gradients, spatial resolution preservation, and long-range dependency modeling. Such advancements have facilitated the deployment of computer vision technologies in facial recognition, medical image analysis (e.g., tumor detection), autonomous driving, and surveillance systems, making automatic visual perception more efficient, accurate, and viable for real-time applications [254].

### 2.6.2. Natural language processing

NLP has been significantly advanced by ML algorithms, owing to their ability to enable the automatic extraction of linguistic patterns, semantic representations, and contextual embeddings from large-scale corpora. Thus reducing reliance on handcrafted linguistic rules, while improving the generalization of NLP systems across languages, domains, and tasks, thereby enhancing their accuracy, scalability, and real-world viability. Classical models such as Naïve Bayes and SVMs were instrumental in early NLP tasks, including spam detection and document classification. However, the advent of deep learning architectures (e.g., RNNs, GRUs, and more recently, Transformer-based models like BERT and GPT) has revolutionized the field by effectively modeling contextual and sequential dependencies in textual data. These advancements have propelled the development of a wide range of NLP applications, including machine translation, sentiment analysis, question answering, text summarization, and conversational agents [255, 256].

### 2.6.3. Cybersecurity

ML models have become integral to the cybersecurity field by enabling the automated detection, classification, and mitigation of security threats, through the analysis of vast and complex networks, systems, and user activity data, consequently reducing reliance on static, signature-based methods, and improving systems overall adaptabilities against evolving attack patterns. Particularly, SVMs and RFs have been widely used for intrusion detection and malware classification, while DNN frameworks, such as CNNs, RNNs, and transformer-based models, have enhanced the capability of capturing spatial, temporal, and contextual patterns in network traffic, logs, and binary codes. These approaches support applications such as anomaly-based intrusion detection, phishing detection, spam filtering, ransomware identification, and user behavior analytics, making cybersecurity systems more robust, scalable, and viable for real-time threat intelligence and proactive defense in dynamic and heterogeneous environments [257, 258].

### 2.6.4. Recommendation systems

Representing the cornerstone of modern recommender systems, ML algorithms enable the automated modeling of user preferences, item characteristics, and interaction patterns, resulting in the efficient provision of personalized content and product suggestions, thereby improving user engagement, retention, and overall decision-making robustness. Traditional approaches such as collaborative filtering (CF), and matrix factorization (MF) have been widely adopted for predicting user-item interactions based on historical data, while DNN architectures (e.g., Neural Collaborative Filtering (NeuMF), Deep Factorization Machines, and Transformer-based models)

have significantly enhanced recommendation performance by capturing complex, nonlinear relationships, contextual dependencies, and temporal dynamics in user behavior, collectively advancing numerous fields encompassing e-commerce, personalized news, music and video, and social media, making recommender systems more adaptive, scalable, and viable for large-scale, real-time, and cross-domain environments [259, 260].

### 2.6.5. Health care

Through their ability to enable the automated extraction of clinically relevant patterns from complex, and multimodal medical data; ML approaches enhance diagnostic accuracy, treatment personalization, and operational efficiency, through both classical models (e.g., logistic regression, decision trees, SVMs) that facilitate early-stage tasks (e.g., disease risk prediction, patient outcome modeling), and by deploying advanced models (e.g., DNNs, Transformer architectures) that enable the efficient analysis of medical imagery, electronic health records, and biomedical signal processing, consequently advancing tasks such as early cancer detection from imaging scans, arrhythmia classification from electrocardiograms, automated pathology report interpretation, and personalized drug recommendation, making healthcare systems more data-driven, scalable, and viable for real-time clinical decision support [261, 262].

### 2.6.6. Finance

The impact of ML techniques on this field has been transformative, enabling the automatic discovery of patterns, trends, and anomalies in large-scale, high-dimensional financial data, thereby enhancing predictive accuracy, risk management, and decision-making efficiency. Specifically, traditional models (e.g., logistic regression, RFs, gradient boosting) are widely implemented for credit scoring, fraud detection, and portfolio optimization tasks, while advanced DNN architectures (e.g., sequential data-oriented neural networks, graph data-oriented neural networks, and transformer-based models) are leveraged to help model the temporal dependencies, relational structures, and market sentiment, extracted from unstructured data (e.g., news articles and social media streams), ultimately advancing the tasks of algorithmic trading, customer segmentation, anti-money laundering systems, and real-time fraud prevention, and making financial services more scalable, adaptive, and viable in dynamic market conditions [263, 264].

### 2.6.7. Manufacturing and industry

In this field, the integration of ML techniques has significantly improved productivity, quality control, and operational efficiency by enabling data-driven process optimization, predictive maintenance, and real-time defect detection. Specifically, SVMs, decision trees, and k-NN are deployed for equipment fault diagnosis, process monitoring, and demand forecasting, whereas DNN-based models are leveraged for handling high-dimensional sensory data, time-series signals, and visual inputs from industrial IoT systems. These capabilities have facilitated successful applications in automated visual inspection, energy consumption optimization, supply chain analytics, and production line automation, making manufacturing systems more adaptive, scalable, and resilient to dynamic market demands [265, 266].

### 2.6.8. Agriculture

The exploitation of ML in this domain have greatly improved precision farming, crop management, and resource optimization through enabling the analysis of multispectral and hyperspectral imagery, sensor data, and environmental variables to support data-driven decision-making. Specifically, decision trees, RFs, and SVMs are leveraged for crop yield prediction, soil quality assessment, and pest detection, while DNN based models support enhancing the capabilities of plant disease identification, weed detection, irrigation scheduling, and climate impact forecasting. These techniques facilitate automated, scalable, and cost-effective solutions dedicated to increase productivity, reduce waste, and promote sustainable agricultural practices, making the sector more resilient to climate change and global food demands [267, 268].

### 2.6.9. Climate science and environmental monitoring

ML algorithms play a crucial role in this field through enabling the efficient modeling of complex earth systems, analyzing vast amounts of remote sensing data, and improving the accuracy of environmental predictions. Most deployed ML approaches involve RFs, SVMs, and k-NN that are implemented for land cover classification, deforestation detection, and air quality assessment, and DNN-based approaches that are dedicated for the more advanced tasks such as climate pattern recognition, extreme weather event forecasting, ocean temperature anomaly detection, and greenhouse gas emission monitoring. These techniques collectively enable the automated, scalable, and high-resolution environmental analysis, support early warning systems, policy-making, and sustainable resource management, thereby enhancing resilience to climate change impacts and environmental degradation [269, 270].

## 2.7. Challenges and limitations of machine learning approaches

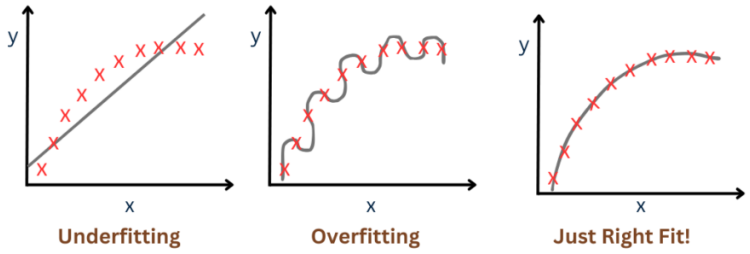
While ML methodologies have achieved remarkable success across numerous disciplines, their widespread adoption remains constrained by persistent theoretical and practical limitations that fundamentally challenge their reliability, generalizability, and scalability. These inherent constraints reveal significant gaps between the theoretical promise of ML paradigms and their real-world applicability, highlighting the need for continued interdisciplinary research to address these foundational challenges. The principal limitations encompass:

### 2.7.1. Computational challenges

#### 2.7.1.1. Underfitting (high bias)

This phenomenon refers to the computational challenge where a model fails to capture the underlying patterns in the training data, typically due to being too simple (Figure 2.29 (left)), having insufficient parameters, or using overly restrictive assumptions, resulting in poor performance on both training and test datasets [139]. This issue often arises from inadequate model complexity, insufficient training time, or the use of overly aggressive regularization, which prevents the model from learning important data relationships. Underfitting negatively impacts ML algorithms by leading to high bias, low predictive accuracy, and an inability to generalize effectively, thereby making the model unsuitable for real-world deployment. The prevention of underfitting involves strategies such as increasing model complexity (e.g., using deeper

architectures or more features), extending training duration, reducing regularization strength, or selecting more appropriate features to better represent the problem space [141].




| | | | |
|---|---|---|---|
| **Bias** | High | Low | Low |
| **Variance** | Low | High | Low |

*Figure 2.29 Understanding underfitting and overfitting [271]*

### 2.7.1.2. Overfitting (high variance)

In contrast to the aforementioned phenomenon, this one depicts the computational challenge where a model learns not only the underlying patterns in the training data but also the noise and random fluctuations, leading to excellent performance on training data but poor generalization to unseen data (Figure 2.29 (middle)). Overfitting typically occurs when the model is either excessively complex with respect to the amount of available data, has too many parameters, or is trained for too long without adequate regularization. Overfitting undermines ML paradigms by causing high variance, reduced robustness, and unreliable predictions in real-world scenarios. Preventing overfitting involves techniques such as regularization (e.g., $L_1/L_2$ penalties, dropout), early stopping during training, cross-validation for model selection, data augmentation, and using simpler models or acquiring more diverse and representative training data [139, 141].

### 2.7.1.3. Data acquisition, quality, and availability

This computational challenge refers to the difficulty of obtaining sufficient, relevant, and high-quality datasets, necessary for effective model training and generalization, since poor data quality (noisy, incomplete, imbalanced, or biased datasets) can lead to inaccurate models that fail to capture the true underlying patterns, while limited availability or restricted access to data (due to privacy, proprietary, or logistical constraints) can hinder the development of robust ones. These issues affect ML models by increasing the risk of overfitting to small or unrepresentative datasets, reducing generalization to unseen data, and potentially embedding societal or measurement biases in predictions. Prevention strategies include employing rigorous data cleaning and preprocessing pipelines, leveraging data augmentation and synthetic data generation, applying feature engineering to maximize information extraction, using transfer learning when large datasets are unavailable, and ensuring ethical and diverse data collection practices to enhance representativeness and reduce bias [139, 272].

### 2.7.1.4. Model complexity and scalability

This challenge revolves around developing approaches that achieve optimal balance between complexity and scalability, through maintaining sufficient representational capacity to capture

intricate data patterns while remaining computationally tractable for large-scale datasets and high-dimensional feature spaces. This dual requirement creates inherent trade-offs between predictive accuracy, computational efficiency, and practical applicability, as dataset sizes and model architectures continue to grow, which can be addressed through regularization techniques, architectural pruning (reduce redundant parameters), the implementation of parallel and distributed computing frameworks, the development of inherently scalable architectures (e.g., sparse neural networks), and the deployment of dimensionality reduction methods (mitigate the curse of dimensionality), which collectively optimize the performance-resource trade-off that fundamentally constrains modern ML systems [273, 274].

#### 2.7.1.5. The curse of dimensionality

The curse of dimensionality encompasses the set of sparsity-related problems that arise when both volume and dimensionality of the input feature space increase, causing an undermining in the statistical significance of training samples, the distortion of distance-based similarity measures, an elevated overfitting risk (due to the expanded hypothesis space), and a prohibitive increase in the computational and storage requirements, collectively degrading ML models generalization capabilities. Addressing these challenges requires researchers to employ dimensionality reduction techniques (e.g., PCA, t-SNE) to project data into more manageable lower-dimensional spaces, the implementation of feature selection methods to identify and retain only the most informative variables, the application of regularization techniques to constrain model complexity, and the utilization of specialized architectures (tree-based ensemble methods, DNNs with embedding layers) that are more robust at handling the peculiarities of high-dimensional data [275, 276].

#### 2.7.1.6. Interpretability and explainability

This challenge stems from the inherent difficulty of comprehending and describing the decision-making processes of complex ML models, especially "black-box" architectures such as DNNs and ensemble methods, where the inability to understand their decision-making process undermines their trustworthiness, accountability, and operational transparency, consequently creating barriers for stakeholders whom require logical reasoning behind model outputs. This lack of transparency particularly hinders deployment in critical domains including healthcare, financial services, and judicial applications, where the reasoning behind estimated predictions carries equal importance to predictive performance. Potential solutions to these limitations involve implementing explainable artificial intelligence (XAI) approaches (e.g., Local Interpretable Model-agnostic Explanations (LIME), attention visualization techniques, gradient-based saliency mapping), or alternatively utilizing structurally transparent models such as decision trees or linear models [277, 278].

#### 2.7.1.7. Hardware and computational limits

Hardware and computational limitations refer to the constraints imposed by processing power, memory capacity, storage, and energy consumption required to train and deploy models, particularly those involving large-scale datasets or complex architectures, which can result in excessively long training times, reduced accuracy due to model/data simplifications, or infeasibility of real-time inference in resource-constrained environments. As model size and

complexity increase, they demand disproportionately more computational resources, which can limit experimentation and raise environmental costs due to high energy consumption. These challenges, however, can be mitigated through strategies such as model compression, pruning, quantization, distributed and parallel computing, utilization of specialized hardware accelerators (e.g., GPUs, TPUs), and algorithmic optimizations that reduce computational overhead without sacrificing performance [279, 280].

### 2.7.2. Real-world challenges

#### 2.7.2.1. Ethical constraints

In ML, ethical challenges involve the responsibility to design, train, and deploy algorithms in ways that uphold fairness, transparency, accountability, and respect for human rights. This includes preventing the amplification of biases, protecting personal privacy, avoiding opaque decision-making, and ensuring algorithms are not misused in contexts such as surveillance or autonomous systems without human oversight, as such failures can erode public trust and lead to discriminatory outcomes. Addressing these concerns requires the integration of fairness-aware algorithms, privacy-preserving techniques, transparent model documentation, robust governance frameworks, and adherence to established AI ethics guidelines and regulatory standards [281, 282].

#### 2.7.2.2. Privacy and security threats

These challenges refer to vulnerabilities that can expose sensitive data or models to unauthorized access, misuse, or manipulation, potentially compromising the confidentiality, integrity, and trustworthiness of ML systems. They commonly include membership inference attacks, model inversion, data poisoning, and adversarial examples, which may result in the leakage of private information, degradation of model performance, or exploitation of learned patterns for malicious purposes. Such risks can undermine algorithm reliability, erode user trust, and limit the adoption of ML models in sensitive applications. Mitigation strategies include the use of privacy-preserving techniques such as differential privacy, secure multi-party computation, federated learning, adversarial training, and robust model validation, combined with strong access controls and continuous security audits to detect and counter both evolving and emerging threats [283, 284].

## 2.8. Conclusion

ML constitutes a powerful and versatile framework for modeling data-driven problems across a wide spectrum of scientific and practical domains. By encompassing diverse learning paradigms (ranging from supervised and unsupervised learning to reinforcement and generative modeling); ML offers flexible solutions for tasks such as prediction, classification, pattern discovery, and knowledge extraction. This chapter has presented the core concepts that define ML, including data types, preprocessing strategies, learning workflows, commonly used algorithms, and modern deep learning architectures. In addition to highlighting the breadth of ML methodologies and applications, this chapter has also emphasized the challenges inherent to real-world ML systems, such as overfitting, scalability, data quality issues, interpretability, and ethical concerns. Addressing these challenges requires careful model design, rigorous evaluation, and responsible deployment practices. Overall, the concepts discussed in this chapter form a critical theoretical

foundation for the advanced ML-based approaches explored in subsequent chapters of this thesis, particularly in the context of complex, data-intensive problems where robust learning and generalization are essential.

# Chapter 3: Recommender Systems in Bioinformatics

Recommender systems have emerged as powerful computational frameworks capable of modeling complex relationships within large, heterogeneous datasets, making them particularly well suited for bioinformatics applications that involve intermolecular interaction prediction, drug discovery, and drug repurposing. By exploiting historical interaction patterns and diverse biological representations, recommender systems offer a scalable and data-driven alternative to traditional experimental and computational approaches, enabling the efficient identification of novel drug-target associations, disease mechanisms, and therapeutic opportunities.

This chapter provides a comprehensive overview of recommender systems in bioinformatics, highlighting their theoretical foundations, methodological adaptations, and practical impact on modern drug discovery and repurposing pipelines. This chapter is organized as follows: Section 2 introduces drug discovery from a bioinformatics perspective, covering the traditional drug development pipeline, its inherent challenges, and the growing role of computational and machine learning-based strategies, with particular emphasis on drug repurposing. Section 3 discusses the concept of intermolecular interactions, their main types, and the importance of computational prediction methods in understanding biological systems and accelerating therapeutic discovery. Section 4 presents the fundamental principles of recommender systems, including their core components, objectives, and taxonomic categories. In Section 5, we explore how bioinformatics problems can be formulated as recommendation tasks, detailing relevant biological data representations and key application domains. Section 6 reviews the most significant contributions of recommender systems in intermolecular interaction prediction tasks, including matrix factorization, deep learning, graph-based, kernel-based, and other recommendation techniques, along with their respective limitations. Concluding with section 7 where we summarize key insights of recommender systems in advancing bioinformatics-driven drug discovery and repurposing strategies.

## 3.1. Introduction

The pursuit of understanding and treating human diseases has driven scientific progress for centuries, ultimately giving rise to the field of drug discovery, which has drastically evolved from the empirical use of natural remedies into a sophisticated, multi-stage pipeline involving target identification, lead optimization, preclinical evaluation, and rigorous clinical testing. Despite these advances, traditional drug discovery remains hindered by prohibitive costs, lengthy development timelines, and high attrition rates, highlighting the urgent need for more efficient strategies such as drug repurposing. At the heart of all biological processes and therapeutic interventions lie intermolecular interactions, the precise binding events between proteins, drugs, nucleic acids, and other biomolecules that dictate cellular functions and disease mechanisms. Accurately characterizing and predicting these interactions has therefore become essential for precision medicine, systems biology, and large-scale therapeutic discovery.

Bioinformatics plays a pivotal role in this effort, offering computational tools to model, infer, and integrate molecular interactions across diverse omics and structural datasets. To manage the growing complexity and scale of such tasks however; bioinformatics has increasingly adopted methodologies from highly compatible domains, such as the recommender system (RS) one. The choice for such approach is based on the strictly-constrained environments characterizing most

biological problems, posing numerous challenges, including data sparsity, cold-start issues for novel compounds or targets, heterogeneous data integration, and the need for interpretable architectures, consequently rendering most ML approaches inapplicable. Originally designed to suggest relevant content to users; RS frameworks operate by inferring missing links between "users" and "items" based on unique features such as historical interactions; a principle that maps naturally onto many biological problems. Owing to their ability to adopt matrix factorization, graph-based, and DL-based techniques, RSs enables the efficient estimation of intermolecular interactions, discovery of disease-gene associations, and acceleration of therapeutic strategies development, where drugs, proteins, genes, and diseases are projected as user/item entities in sparse interaction matrices. By reframing biological questions as recommendation tasks, RS methodologies provide a systematic and scalable approach for exploring molecular interactions, offering new opportunities for therapeutic innovation and personalized medicine.

## 3.2. Drug discovery in bioinformatics

### 3.2.1. Definition and scope

Drug discovery is considered as one of the main cornerstones of biomedical research, aiming to identify novel therapeutic compounds that're capable of alleviating or preventing human diseases caused by molecular dysfunctions that disrupt regular biological processes through genetic mutations, pathogens, or external environmental factors [286]. The primary goal of drug discovery is to design or identify molecules capable of restoring physiological balance, accomplished by specifically targeting aberrant molecular mechanisms in a process that involves interacting with biomolecules such as proteins (e.g., enzymes, receptors, or ion channels) to help modulate their activity and produce a therapeutic effect. At the molecular level, most drugs exert their function through specific binding interactions with target proteins (primarily mediated by non-covalent forces such as hydrogen bonds, hydrophobic effects, and electrostatic interactions) in order to inhibit or accelerate their biological activity, thereby modulating biochemical pathways involved in disease progression [285]. For instance, enzyme inhibitors can block the catalytic activity of a pathogenic or dysregulated protein, while receptor agonists can mimic endogenous ligands to activate signaling pathways that promote normal physiological responses. A comprehensive understanding of these interactions is crucial, as protein-drug binding affinity, selectivity, and kinetics largely determine a compound's therapeutic efficacy and safety profile. Consequently, the importance of drug-target interactions (DTI) extends beyond clinical potency, serving as a foundation for predicting side effects, drug resistance, and metabolic stability. Therefore, modern drug discovery strives to integrate computational, structural, and experimental methodologies to model these interactions at atomic resolution, to enable the rational design of more selective and potent molecules, while bridging disciplines such as chemistry, biology, pharmacology, and computational sciences, to translate molecular insights into effective clinical therapies [286].

### 3.2.2. Traditional drug development pipeline

The transition of a promising chemical compound from a laboratory discovery to a commercially available medicine is a rigorous, multi-stage process known as the drug development pipeline. This highly regulated pathway is designed to comprehensively assess a drug candidate's

safety, efficacy, and quality, ensuring that only beneficial and safe therapeutics reach patients. This pipeline (depicted in Figure 3.1) is broadly divided into discrete, sequential stages, each with a defined objective and a high attrition rate, meaning that candidate compounds with the slightest shortcomings fail to progress to the next phase. It includes:

#### 3.2.2.1. Target identification and validation

This step represents the foundational phase of the drug discovery process, where researchers aim to determine and validate potential targets (proteins, genes, RNA, or any other biomolecule that plays a critical role in disease onset or progression) or pathways whose modulation can alter the course of a disease. The objective of this phase is to elucidate the molecular mechanisms underlying a pathological condition, and to pinpoint components whose inhibition or activation could yield therapeutic benefits, a process that necessitates leveraging clinical observations of patient populations, epidemiological studies linking genetic or environmental factors to diseases, and genetic and biochemical techniques like genome-wide association studies and transcriptomics for more accurate identification of dysregulated biomolecules in diseased states (i.e. potential molecular targets) [287]. Once candidate targets are identified, they undergo rigorous validation using advanced tools such as functional genomics (e.g., CRISPR screens), analysis of protein-protein interactions (PPI), cell-based disease models, and the use of bioactive compounds or antibodies to confirm their causative role in disease biology and therapeutic tractability [288].

#### 3.2.2.2. Lead discovery

The second step in drug developing involves identifying lead compounds (i.e., chemical entities that demonstrate desired pharmacological effects by interacting with the validated biological targets), that serve as templates in latter optimization steps [287]. This phase employs several empirical and computational strategies, including (1) high-throughput screening, where automated systems test vast chemical libraries for biological activity against desired targets, (2) virtual screening, which employs computational modeling and molecular docking simulations, alongside the structural information of targets to predict the results of their interactions with candidate drugs *in silico*, enabling the prioritization of compounds with high binding potential while minimizing experimental workload and costs, and (3) natural product screening, which investigates bioactive molecules derived from plants, microorganisms, and marine organisms for novel mechanisms and scaffolds. The identified compounds that show promising initial activities (referred to as “hits”) are then systematically evaluated and optimized through a Hit-to-Lead (H2L) process, where their chemical structures are refined to improve potency, selectivity, and early pharmacokinetic properties, thereby yielding a promising lead series for subsequent extensive lead optimization [289].

#### 3.2.2.3. Lead optimization

This crucial, iterative phase in drug discovery focuses on systematically modifying a lead compound's chemical structure, to enhance its overall profile as a potential therapeutic agent in a process that involves (1) improving key properties, including binding potency and selectivity for the intended biological target; (2) optimizing pharmacokinetics (i.e., how the body absorbs, distributes, metabolizes, and eliminates an administered drug) such as absorption, distribution,

metabolism, excretion, and toxicity (i.e., ADMET properties); while (3) reducing early toxicity signals [287]. This multidisciplinary process heavily relies on (i) the integrated application of medicinal chemistry, where chemists synthesize and test numerous structural analogs; (ii) structural biology, using techniques like X-ray crystallography to visualize the atomic-level interaction between the drug and its target to guide rational design; and (iii) comprehensive pharmacology, which evaluates the refined compounds in increasingly complex *in vitro* and *in vivo* preclinical models to assess efficacy, safety, and pharmacodynamics properties (i.e., how a drug affects the body on both the biochemical and physiological level). The outcome is the selection of one or more optimized drug candidates that exhibit a balanced and superior combination of efficacy, safety, and developability, ready for formal preclinical development and regulatory submission prior to human clinical trials [287] [290].

#### 3.2.2.4. Preclinical testing

This mandatory and comprehensive stage aim to evaluate the safety and biological activity of a lead candidate *in vitro* and *in vivo* before it can be administered to humans in clinical trials, via employing a tiered experimental approach that begins with (1) *in silico* assays that use computer simulations to model molecular interactions and predict their outcomes and properties, followed by (2) extensive *in vitro* studies on cell cultures and *ex vivo* experiments on living tissues to investigate the compound's mechanisms of action and initial cytotoxicity (i.e., the capacity of a drug to damage or kill living cells) [287]. The most critical data comes from *in vivo* studies conducted in animal models (e.g., rodents, dogs), which are used to characterize the compound's full pharmacokinetic profile, establishing a proof-of-concept for its efficacy in a whole-organism system, and identify potential toxicological liabilities at various doses. The collective goal of these studies is to generate sufficient data on the drug's safety, efficacy, and appropriate starting dosage to support the filing of an Investigational New Drug (IND) application with regulatory authorities, justifying its first-in-human testing [291].

#### 3.2.2.5. Clinical trials

This multi-phase human testing stage is designed to further evaluate the safety, efficacy, and pharmacokinetic profiles of a new drug candidate in a controlled human population. This process is conducted in sequential, rigorously monitored phases that include: Phase-I trials that primarily assess safety, tolerability, and pharmacokinetics in a small cohort of healthy volunteers to determine a safe dosage range; Phase-II trials expand the investigation to a limited patient population to obtain initial evidence of the drug's efficacy and further evaluate its short-term safety; and Phase-III trials that depict large-scale, multi-center studies conducted on hundreds to thousands of patients to definitively confirm therapeutic benefit, monitor adverse effects, and compare the new treatment to the current standard of care [287]. The collective data generated from these phases is essential for regulatory approval, providing the critical evidence required to demonstrate that the drug's benefits outweigh its risks for the intended patient population [292].

#### 3.2.2.6. Regulatory approval

This crucial gatekeeping step encompasses compiling a comprehensive dossier of all non-clinical and clinical data concerning a new drug candidate that’s submitted to a regulatory authority

(e.g., U.S. Food and Drug Administration (FDA), European Medicines Agency (EMA)). These agencies conduct a rigorous, independent review of the submitted evidence and perform an analysis that determines whether the drug is safe and effective for its intended use (has it demonstrated therapeutic benefits that outweigh its known and potential risks). Only upon a positive assessment is a marketing authorization granted, legally permitting the pharmaceutical sponsor to market, sell, and distribute the drug to patients and healthcare providers [287] [293].

#### 3.2.2.7. Post-marketing surveillance

Also known as pharmacovigilance, is the continuous and mandatory monitoring of a drug's safety and efficacy after it has received regulatory approval and is being prescribed to the general population in real-world clinical practice. This phase aims to detect rare, delayed, or long-term adverse drug reactions that could not be observed in earlier stages, alongside assessing the drug's effectiveness in a much larger and more diverse patient population with various comorbidities (i.e., simultaneous presence of multiple diseases or medical conditions in the same individual) and concomitant medications, in order to potentially restrict its use, or in rare cases, withdraw it from the market to protect public health [287] [294].

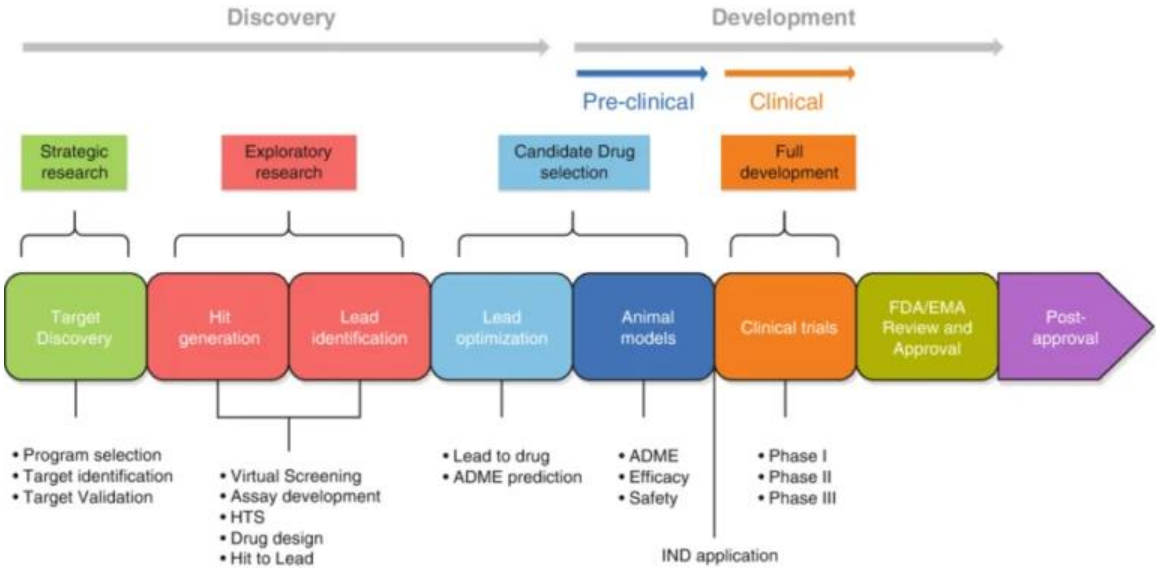



*Figure 3.1 Schematic representation of the drug discovery process [287]*

### 3.2.3. Challenges and limitations of traditional drug development

The traditional drug development pipeline, while rigorous and dedicated to ensure safety and efficacy, is fraught with profound challenges that include [295]:

- ✓ **High cost and time consumption:** Traditional drug development is notoriously expensive and time-intensive, often requiring 10-15 years and billions of dollars to bring a single drug to market, with lengthy preclinical and clinical testing phases, alongside both stringent regulatory requirements and high failure rates (especially in late-stage clinical trials) that further inflate said expenses, consequently limiting innovation, and discouraging investments in therapeutics that’re dedicated for rare and neglected diseases.

- ✓ **High attrition rates and unpredictable efficacy:** A major limitation of traditional drug development is the high attrition rates, as most drug candidates fail to pass the clinical trials due to inadequate efficacy or unforeseen toxicity. With traditional models (e.g., animal testing, two-dimensional cell cultures) frequently failing to replicate the complexity of human physiology, resulting in poor predictability of drug behaviors in clinical settings, underscoring the difficulty of translating preclinical findings to human outcomes.
- ✓ **Limited translational relevance of preclinical models:** Despite being indispensable, animal models and *in vitro* systems often fail to mimic the physiological and genetic properties of human biology (e.g., metabolism, immune responses, and genetic variability). As a result, drugs that appear to be effective in preclinical models may perform poorly in clinical trials, underlining the need for more advanced systems to better bridge the translational gap.
- ✓ **Regulatory constraints:** While necessary for ensuring public safety, the regulatory pathway is complex and demanding, since navigating the requirements of agencies like the FDA and EMA requires significant expertise and time, especially with the evolving guidelines and the need for extensive documentation, which can lead to delays and increased costs, creating formidable barriers for smaller companies and academic spin-offs.
- ✓ **Lack of personalization in drug development:** Conventional drug development often focus on average patient responses rather than inter-individual variability, ultimately contributing to suboptimal efficacy and adverse reactions in specific patient subgroups. Advances in genomics and precision medicine have highlighted the importance of tailoring drugs to molecular and genetic profiles, but such strategies remain underutilized in traditional frameworks.
- ✓ **Ethical and safety concerns:** Ethical considerations, particularly in animal experimentation and human clinical trials, remain a significant limitation, as preclinical studies often raise concerns regarding animal welfare, while human trials must carefully balance innovation with patient safety. Furthermore, unexpected adverse events during or after approval can lead to drug withdrawals, undermining public trust and regulatory confidence.

### 3.2.4. Drug development vs drug repurposing

Drug repurposing, also known as drug repositioning, is a strategic approach in pharmaceutical research that identifies new therapeutic use cases for existing drugs (e.g., approved, discontinued, shelved, or investigational) outside their original medical indication. This strategy deliberately bypasses the need for *de novo* compound discovery by leveraging molecules with already-established human safety profiles, pharmacokinetic data, and often, scalable manufacturing processes, uncovering new disease targets or pathways that the repositioned drug can effectively modulate, which unlocks new therapeutic values for it. While both drug discovery and repurposing share the ultimate goal of developing effective therapeutics (Figure 3.2), they differ significantly in methodology, timeline, and resource investment. In contrast to traditional drug discovery that follows the aforementioned linear pipeline (requires decades and costs billions of dollars); drug repurposing bypasses many of these early-stage hurdles by utilizing compounds with pre-established safety and pharmacokinetic data, therefore shortening development timelines while lowering attrition rates [96]. Additionally, whereas drug discovery emphasizes *de novo*

identification of novel molecular entities for specific targets, repurposing focuses on mapping existing molecules to new biological pathways or disease contexts through computational modeling, omics-based analysis, or clinical observation. Rather than being opposing strategies, drug discovery and repurposing operate synergistically to advance therapeutic options, as the comprehensive data generated during a drug's initial development (e.g., off-target effects, full adverse event profile) serves as a rich knowledge base for more prominent repurposing hypotheses. Moreover, drugs that fail late-stage trials for their initial indication due to lack of efficacy become prime candidates for repurposing to other diseases with shared biological pathways, thereby salvaging immense investments. This symbiotic relationship creates a more resilient and efficient drug development ecosystem, where traditional discovery provides a continuous pipeline of new molecular entities, while repurposing maximizes the therapeutic potential and return on investment from each compound, ultimately accelerating the delivery of new treatments to patients [297].

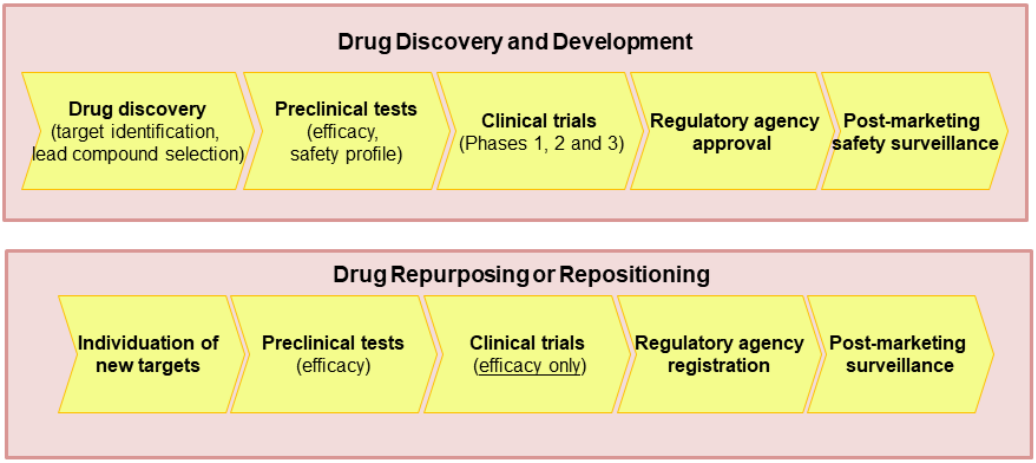



*Figure 3.2 The main differences between drug discovery and drug repurposing [298]*

### 3.2.5. Computational ML methods in drug discovery and repurposing strategies

Computational ML methods have significantly revolutionized *de novo* drug discovery strategies by augmenting and accelerating nearly every stage of their respective pipeline. In the early stages, ML models can exploit and analyze multi-omics data (genomics, proteomics) to help prioritize novel and druggable biological targets and predict potential off-target effects. For lead discovery, they can leverage virtual screening of immense chemical libraries, using structures learned from known active molecules to predict the bioactivity and binding affinity of new candidates with high accuracy, far surpassing traditional methods in speed and scope. Furthermore, in lead optimization, ML-powered quantitative structure-activity relationship (QSAR) models guide medicinal chemistry by predicting how structural modifications would influence crucial properties like potency, selectivity, and pharmacokinetic profiles, thereby systematically designing better drug candidates with reduced experimental cycles. Whereas the more advanced DL architectures (i.e., CNNs, RNNs, GNNs, and transformer architectures) enable representation learning from molecular graphs and protein sequences, allowing an end-to-end prediction of binding affinities and bioactivity profiles, and collectively paving the way toward more predictive, cost-efficient, and precision-oriented drug discovery pipelines [299]. On the other hand, ML methodologies play a pivotal role in drug repurposing strategies through the systematic prediction

of novel drug-disease associations through the analysis of complex, high-dimensional biological and chemical data (e.g., drug chemical structures, protein targets, genomic signatures of diseases, clinical data, and health records) to identify latent patterns and relationships typically unnoticed through manual investigations. Common ML approaches suitable for this task include recommender systems (RS) and their ability to complete sparse drug-target and drug-disease interaction matrices, GNNs that can traverse heterogeneous biological networks linking drugs, proteins, and diseases, and DL approaches that're capable of integrating multiple data types to predict therapeutic efficacy. By generating high-probability associations that link existing drugs with emerging pathways of new diseases, ML techniques significantly accelerate the repurposing pipeline, prioritizing the most promising candidates for experimental validation in wet-lab settings and clinical trials, thereby reducing both the cost and time required to discover new therapeutic uses for old drugs [300].

## 3.3. Intermolecular interactions in bioinformatics

### 3.3.1. Definition

Intermolecular interactions refer to the specific, non-covalent physical or chemical forces that govern the bindings and associations occurring between biological molecules (e.g., proteins, nucleic acids, lipids, and small molecules) that underlie virtually all cellular and physiological processes (Figure 3.3) [301]. These interactions are fundamental to the formation of molecular complexes, as well as the regulation of biological functions (e.g., enzymatic catalysis, signal transduction, gene expression, and immune responses), with bioinformatics being fundamentally concerned with providing the computational tools needed for representing, predicting, and analyzing them, allowing researchers to infer association networks and uncover the molecular mechanisms of diseases and other major cellular processes. The precise computational characterization of these interactions, particularly their strength, specificity, and functional consequences, is crucial in applications like the identification of therapeutic targets, rational drug design, and the overall understanding of the molecular basis of genetic variants [302].

### 3.3.2. Types of intermolecular interactions

In bioinformatics, intermolecular interactions are systematically categorized based on the biological entities involved in them, each possessing distinct structural, functional, and data-characteristics that determine how they are detected, modeled, and exploited for applications like target identification, network analysis, and therapeutic design. Additionally, with computational methods and databases being tailored to the properties of each interaction class (e.g., binding site size, affinity range, experimental observability); distinguishing them is critical for appropriate algorithm selection and interpretation. Most well-known interactions include:

#### 3.3.2.1. Drug-Target Interactions (DTIs)

DTIs refer to the specific binding events occurring between a chemical compound (i.e., drug) and a targeted protein, which results in a modulation in the target's activity (e.g., inhibition, acceleration) and a subsequent therapeutic or adverse physiological effect. DTIs represent the foundation for understanding drug mechanisms of action, predicting side effects, and guiding drug discovery and repurposing efforts through more efficient computational modeling. The strength

and nature of DTIs are experimentally characterized via biochemical and biophysical assays that quantify their binding affinities, often expressed as dissociation constant ($K_d$), inhibition constant ($K_i$), or half-maximal inhibitory ($IC_{50}$) and effective concentrations ($EC_{50}$), typically measured in molar ($M$) units. High-throughput computational techniques like molecular docking, and ML-based models, are commonly employed to study and estimate DTIs, since understanding them is crucial for identifying novel therapeutic targets, minimizing off-target toxicity, and accelerating rational drug design *in silico* [303].

#### 3.3.2.2. Protein-Protein Interactions (PPIs)

These interactions refer to the specific, physical associations and contacts between two or more proteins (driven by a combination of non-covalent forces), that are essential for forming functional complexes, and mediating virtually all cellular processes (e.g., signal transduction, metabolic regulation, and immune responses). Similar to DTIs, PPIs can be quantified by $K_d$ and $K_i$ (lower values indicate stronger binding affinities), alongside other measures such as the association and disassociation rate constants $K_{on}$ and $K_{off}$, and confidence values (i.e., normalized scores reflecting the reliability and reproducibility of experimental evidence supporting the interaction) assigned by databases (e.g., IntAct) such as the MI-score. The computational ML tools and databases employed in the bioinformatics field enable predicting, storing, and analyzing PPI networks, facilitating target identification, the functional annotation of uncharacterized proteins, and the elucidation of complex genotype-to-phenotype relationships [304].

#### 3.3.2.3. Drug-Drug Interactions (DDIs)

DDIs depict the pharmacological, biochemical, or physiological effects that occur when two or more drugs are co-administered, leading to quantitative or qualitative changes in the effects of one drug due to the presence of another. DDIs are systematically represented as structured, quantifiable relationships derived from clinical observations, electronic health records, and curated databases, that are quantified using pharmacokinetic metrics such as the area under the plasma concentration-time curve, or probabilistic confidence scores that’re assigned by computational prediction pipelines. Understanding DDIs is crucial, since drug combinations are widely used for treating complex diseases, and unexpected interactions can cause therapeutic failure, toxicity, or life-threatening adverse reactions. Therefore, necessitating the need for computational ML approaches to help identify hazardous drug combinations, optimize multi-drug therapies, and improve patient safety by modeling interaction likelihoods and their biological pathways across large pharmacological networks [305].

#### 3.3.2.4. Protein-Nucleic Acid Interactions

They refer to the specific binding events that involve proteins and either DNA or RNA molecules, which are fundamental to nearly all genetic and regulatory processes, including transcription, translation, chromatin remodeling, replication, DNA repair, RNA splicing, and post-transcriptional regulation. Quantitatively, protein-nucleic acid interactions are often measured using binding affinity metrics (e.g., $K_d$), and, in high-throughput genomics assays, statistical significance scores (e.g., $p$-value, enrichment score) that reflect the reliability of interaction peaks relative to background signals in genome-wide binding maps. The importance of estimating these

interactions via ML approaches is immense, as accurately deciphering them elucidates regulatory networks, enables the identification of disease-associated mutations that disrupt regulatory elements, enhances the design of gene-editing tools, and supports the development of more effective therapeutic strategies [306].

#### 3.3.2.5. Drug-Disease associations

They refer to the validated or predicted relationships linking pharmacological compounds, and diseases or pathological conditions they influence, either through therapeutic benefit or by inducing adverse effects. These associations are typically derived from diverse data sources (e.g., clinical trial outcomes, electronic health records, gene-disease annotations, and curated databases) and are computationally represented as linkages within heterogeneous biomedical networks. Quantitatively, drug-disease associations are measured using several scoring schemes, including categorical labels (e.g., approved, investigational, contraindicated), evidence-based confidence scores derived from integrated datasets, and statistical measures (e.g., $p$-values, likelihood ratios, and enrichment scores). These associations are critically important for pharmacology, precision medicine, and computational drug repurposing, as they enable the elucidation of disease mechanisms and the identification of novel therapeutic opportunities. Consequently, ML approaches that can estimate these associations by integrating side information (e.g., molecular, phenotypic, and network-level features) are becoming increasingly indispensable [307].

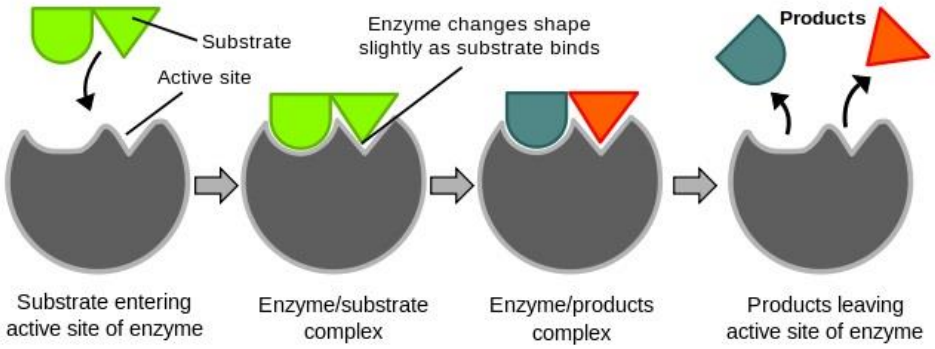



*Figure 3.3 Enzym-substrare interaction [308]*

### 3.3.3. The importance of computational prediction of intermolecular interactions

Computational prediction of intermolecular interactions is essential for modern biomedical research, as it provides the foundational insights necessary to bridge the gap between molecular-scale events and system-level biological functions, a task that is prohibitively expensive, time-consuming, and often infeasible through experimental methods alone. This predictive endeavor is crucial for (1) rationally designing drugs with high specificity and fewer side effects, (2) understanding the molecular etiology of diseases (e.g., what causes cancer, or neurodegenerative disorders), and (3) repurposing existing drugs by revealing novel drug-disease associations. The field of ML drastically facilitates this challenge by enabling the efficient learning of complex, non-linear dependencies that govern molecular bindings directly from vast and heterogeneous datasets. ML paradigms, including RS and various classification and regression approaches, leverage techniques like DL, GNNs, CNNs, and transformer architectures to integrate diverse side

information (e.g., molecular structures, omics profiles, interaction records, and clinical annotations), that enable the prediction of binding affinities, interaction probabilities, and functional effects with unprecedented scale and accuracy (often framed as matrix completion or missing link inference tasks). Consequently, ML is expanding our ability to map intermolecular interaction landscapes with higher resolutions that further advances therapeutic discovery [299].

## 3.4. Recommender systems: principles and foundations

### 3.4.1. Definition and concept of work

RS are a specialized subclass of information filtering systems (i.e., techniques designed to mitigate information overload by selectively presenting items that are estimated to align with users interests) that employ computational techniques and methodologies for enabling the automatic identification and suggestion of items (e.g., products, services, content) to users by analyzing divers features and characteristics (Figure 3.4) [309, 310]. RS operate by analyzing multiple data sources (depending on the deployed RS technique) in order to estimate and provide relevant items to users. This process begins by collecting user-related information such as past behaviors, explicit feedback (e.g., ratings), implicit feedback (e.g., clicks, dwell time), or contextual signals (e.g., time, location), alongside item characteristics (e.g., price, category, brand), which are stored in structured user-item interaction matrices where rows represent users, columns represent items, and entries capture the nature and strength of their respective interactions (e.g., movie ratings, purchase frequency). The deployed recommender then processes this data through pattern recognition methodologies that identify similarities between users (users who liked $X$ also liked $Y$), relationships among items (this product shares features with previously purchased one), and latent behavioral patterns (e.g., frequent co-purchases, genre preferences), which enable the recommender to estimate a user's interest in previously unrated items. Finally, the RS generates recommendations by predicting relevance scores for candidate items, applying ranking algorithms that optimize for accuracy while maintaining diversity and novelty, and filtering results based on appropriateness, trendiness, and other constraints. This multi-stage approach balances predictive performance with user experience quality, adapting dynamically as new interaction data becomes available [309-312].

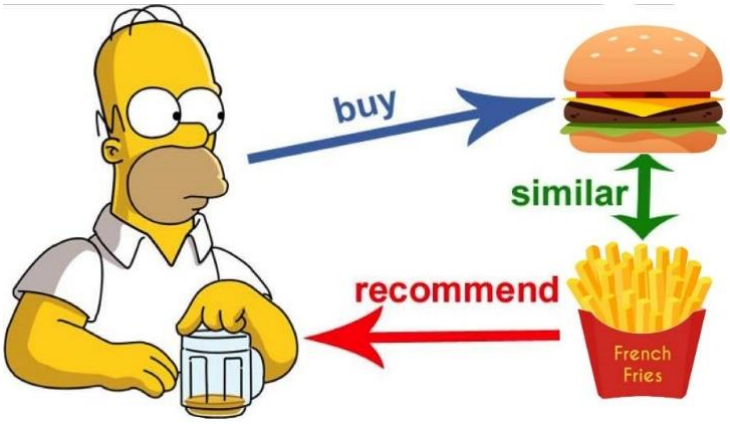



*Figure 3.4 The concept of work of recommender systems*

### 3.4.2. Atoms leveraged by recommender systems

RS derive their cognitive capabilities through the systematic processing of four atomic data primitives [310, 312], which incorporate:

#### 3.4.2.1. User atom

A structured representation of an individual's identity, typically comprising (1) explicit attributes such as demographic characteristics (e.g., age, gender, location), self-reported preferences, and declared interests, (2) implicit behavioral signals (e.g., ratings, clicks, purchases, dwell time), and (3) contextual metadata that captures situational factors (e.g., time, location, used device). These features collectively enable user preference modeling, which in return promotes item recommendation through computational similarity measures and predictive analytics. Mathematically, a user atom is defined as an element $u \in U$, where $U$ denotes the set of all users in the system, in which users serve as one endpoint in the user-item interaction space, enabling algorithms to estimate a utility function $\hat{u} : U \times I \to \mathbb{R}$ that predicts the degree of relevance of an item to a specific user [309, 312].

#### 3.4.2.2. Item atom

A computational unit that depicts a recommendable object, formally defined as $i \in I$, where $I$ denotes the set of all items in the system, in which $i$ depicts a tuple that encodes all object-related information necessary for recommendation tasks, which include (1) the intrinsic content features such as object properties (e.g., product specifications, media genres), (2) relational metadata (e.g., categorical hierarchies, brand affiliation), and (3) contextual attributes encompassing situational factors (e.g., temporal availability, geographical constraints). Together, these components enable the efficient similarity computations among items (e.g., item-item cosine distance), compatibility assessment with user atoms, and dynamic ranking based on situational factors, ultimately forming the fundamental counterpart to user atoms in the user-item space, and directly mitigating the impact of cold-start problem on recommendation tasks [309, 313].

#### 3.4.2.3. Interaction atom:

In RS, this atom denotes the minimal observable unit of engagement between a user $u$ and an item $i$, formally encoded as either: (1) an element $r_{ui}$ in a user-item interaction matrix $R$, or (2) a weighted edge in a bipartite graph where users and items form disjoint node sets. Each interaction atom provides measurable evidence of user preference towards an item through either explicit feedback (i.e., signals directly provided by users, such as numerical ratings or reviews), or implicit signals (i.e., behavioral traces like clicks, purchases, or dwell time), with associated confidence weights reflecting observation reliability, paired with inherent contextual metadata (e.g., temporal stamps, spatial coordinates, device information) to further elucidate the observed behaviors. As the foundational training data for RS; interaction atoms enable preference inference through either matrix completion, link inference, regression, or classification, with their sparsity, distribution, and noise directly affecting the overall performance of the deployed RS [309, 312, 314].

#### 3.4.2.4. Rating matrix atom

This atom depicts the general mathematical representation of user-item interactions, serving as a finite measurement space that partially depicts true user-item preference manifold, and the ground truth leveraged by evaluation metrics. Rating matrix is formally defined as a partially observed matrix $R \in \mathbb{R}^{|U|\times|I|}$, where each element $r_{ui}$ quantifies the strength or preference a user for an item (based on context, it can also be considered as the adjacency matrix for the user-item bipartite graph) [315]. In real-world systems, the sparsity of $R$ (typically less than $5\%$ of entries are observed, due to limited user engagement with the full item set) poses a central challenge for most RS, necessitating specialized approaches that offer prominent generalization capabilities that enable the efficient approximation of missing entries [312, 316].

### 3.4.3. Aims and goals of recommender systems

RS have emerged as essential computational tools for addressing information overload in digital environments, employing algorithmic filtering, predictive ranking, and personalized suggestion mechanisms to optimize user decision-making and improve user experience. Their primary objectives include:

#### 3.4.3.1. Relevance

This foundational objective ensures that recommended items align with user needs, typically through learning the conditional probability $P(i|u)$ that reflects the relevance of item $i$ to the interests of user $u$, given their explicit preferences and implicit behavioral patterns, ultimately maximizing recommendation accuracy that empirically correlates with user trust and engagement. Beyond user satisfaction, relevance drives numerous platform-critical outcomes, encompassing customer retention through repeated positive interactions, conversion rates (i.e., the proportion of recommendations that result in desired outcomes) via targeted suggestions, and decision-making efficiency in high-stakes domains (e.g., clinical treatment recommendations). However, over-optimization for relevance risks filter bubbles (i.e., users are repeatedly exposed to a narrow set of similar content, while ignoring diverse or novel alternatives), and reduced catalog coverage, underscoring the need to balance such objective with diversity and the other objectives [316, 317].

#### 3.4.3.2. Diversity

The pursuit of this complementary objective ensures that recommended items span a broad range of the item space by intentionally promoting heterogeneity across suggestions, typically quantified through intra-list diversity metrics (e.g., average pairwise cosine dissimilarity between item embeddings) or entropy-based measures that capture the variance and balance of recommended content. Through mitigating over-specialization and improving exposure to long-tail items; diversity prevents monotony through enhancing user experience by increasing the exposure to novel and varied content, while supporting platform-level goals including improved catalog coverage, reduced filter bubbles, and increased fairness toward underrepresented items. However, its over-optimization risks diluting recommendation accuracy and predictive relevance, as excessive heterogeneity can introduce weakly aligned items that diminish user satisfaction and trust, necessitating multi-objective optimization frameworks that can balance it with relevance [318, 319].

#### 3.4.3.3. Novelty

This exploratory objective ensures that recommended items expose users to unknown or unexplored content, typically measured by popularity-based novelty (e.g., inverse item frequency) or self-information, which capture the unexpectedness of items relative to a user's historical interactions, therefore expanding their horizons beyond established preference boundaries. Beyond fostering curiosity and sustaining long-term engagement; novelty drives crucial platform-level outcomes, encompassing preventing user experience stagnation, improving ecosystem health by redistributing attention toward long-tail items (i.e., less popular content aligned with niche interests), and strengthening retention and market differentiation via continuously refreshing user experiences. However, over-optimization for novelty risks undermining perceived relevance and trust, as consistently recommending obscure or weakly aligned items can overwhelm users, reduce recommendation accuracy, and ultimately decrease satisfaction, highlighting the need to balance novelty with relevance and diversity [320, 321].

#### 3.4.3.4. Serendipity

This behavioral goal ensures that recommendations are both unexpected (deviating from user expectations) and useful (providing value beyond immediate preferences), formally modeled as the joint optimization of novelty and relevance. Serendipity drives experiential outcomes like discovery-oriented engagement and long-term trust through "pleasant surprise" that increase the perceived value of the system beyond mere accuracy or diversity. Over-optimization for such objective however risks perturbation effects (i.e., excessive deviation from user profiles introduces irrelevant items) that may confuse or frustrate users, thus degrade accuracy and trust, highlighting the necessity of balancing serendipity with relevance, novelty, and diversity [322, 323].

### 3.4.4. Categories of recommender systems

Despite leveraging diverse algorithmic techniques, most RS fundamentally process a shared set of data aspects, encompassing both the behavioral patterns captured within the user-item interaction matrix (e.g., ratings, clicks, purchases), and the explicit attributes assigned to users and items. These core data types serve as the foundational input for a wide range of methods (Figure 3.5), which include:

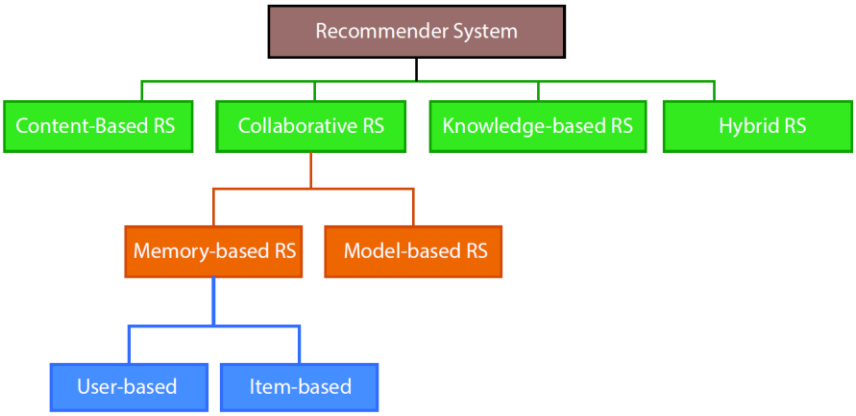



*Figure 3.5 Types of recommender systems [324]*

#### 3.4.4.1. Content-based (CB)

CB approaches aim to generate personalized suggestions by leveraging the intrinsic attributes of items alongside the historical preferences of individual users, while operating under the assumption that user preferences remain stable over items with similar characteristics. Particularly, CB methods start by constructing item profiles via encoding their explicit attributes (e.g., textual description, molecular fingerprints), then generate user profiles by aggregating the profiles of items that a user has previously interacted with, consequently enabling the computation of a new item's relevance to a user by measuring the similarity between their respective profiles (often performed via cosine similarity, TF-IDF weighting, or learned embeddings). CB systems can be broadly categorized into (1) traditional models that rely on hand-engineered keyword-based features (e.g., metadata, tags), and (2) modern approaches that utilize DL to extract latent representations directly from unstructured data (e.g., text, images, molecular textual representations). A key strength of CB approaches lies in their interpretability and their ability to recommend novel or niche items that match a user's specific tastes without requiring data from other users, making them particularly resilient to cold-start problems. Despite being susceptible to over-specialization feedback-loops emerging from the repeated recommending of near-identical items, consequently diminishing their diversity and serendipity [312, 325].

#### 3.4.4.2. Collaborative filtering (CF)

CF techniques operate based on the core premise that homogeneous users exhibit similar preferences (Figure 3.6), an idea often linked to the *"wisdom of the crowd"* concept, which synergies with the objective of CF methods that is the approximation of user-item interaction matrix solely by relying on the historical user behaviors.

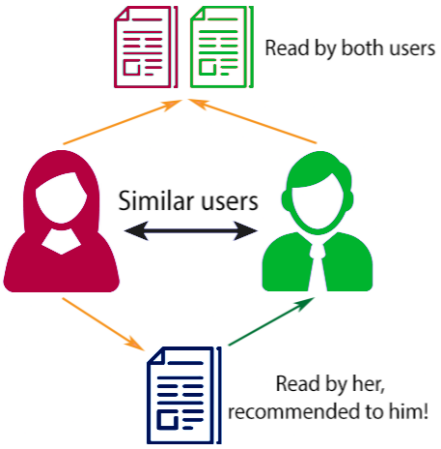



*Figure 3.6 Collaborative filtering technique in recommender systems [327]*

Specifically, CF methods infer user preferences for items based on the collective behavior of the entire user community, accomplished by leveraging either user-user correlations (i.e., identifying users with similar behavioral patterns) or item-item similarities (i.e., identifying items consumed by similar users). Both approaches are implemented through either (1) neighborhood-based (memory-based) methods that compute these relationships directly from the interaction matrix (e.g., k-NN), or (2) model-based approaches (e.g., SVD, matrix factorization, DL-based

frameworks) that build predictive models capable of uncovering underlying, latent relations that collectively explain the observed user-item interactions. A key strength of CF techniques is their ability to function without depending on any explicit attribute data, or any form of feature engineering, while being adaptive to changing user tastes, which helps them discover complex, non-obvious associations that transcend superficial attributes. However, they are more sensitive to data sparsity and the cold-start problems [310, 326].

#### 3.4.4.3. Knowledge-based

Knowledge-based recommenders are founded based on the principle claiming that recommendations are derived from an explicit domain knowledge that maps the relationships between user needs and the item space without relying on historical interactions. Their core mechanism involves an inference process where a user's explicitly stated requirements (obtained via a conversational interface, forms, or constraint specification) are matched against a domain knowledge base (e.g., rule set, case base) using reasoning mechanisms such as constraint satisfaction, case-based reasoning, or utility-based ranking to identify items that precisely fulfill the defined criteria. This approach manifests primarily in two paradigms: (1) constraint-based systems, which solve a set of hard and soft user-defined constraints to find suitable items, and (2) case-based systems, which retrieve items most similar to a user-specified query or a known successful case. The key advantages of knowledge-based systems include their interpretability (the reasoning path can be traced and explained), complete independence from historical data (thus eliminating cold-start and data sparsity problems), and their suitability for complex, infrequently purchased, or highly configurable domains (e.g., financial services, real estate). Nevertheless, a significant knowledge-engineering effort is required to build and maintain the knowledge base, which is static in nature, hence cannot automatically learn or adapt to evolving user tastes, while potentially frustrating them with the constant demand of their specifications upfront [309, 328].

#### 3.4.4.4. Hybrid recommenders

Acknowledging that no single recommendation strategy is universally optimal, hybrid systems synergistically integrate multiple techniques to mitigate individual weaknesses and amplify complementary strengths. They combine multiple recommendation strategies, typically by fusing the input data, algorithmic processes, or final outputs from two or more distinct recommenders (e.g., CB, CF, or knowledge-based systems) through established fusion mechanisms such as weighted hybridization (i.e., combining the weighted scores of several recommenders), switching (i.e., selecting the most suitable model based on context), cascading/pipelining (i.e., using one recommender to refine or pre-process data for another), and feature augmentation (i.e., enriching one model's input with features generated by another). Deploying a hybrid recommender over a single method offers elevated robustness to data sparsity and cold-start problems (by leveraging multiple and complementary data sources), enhanced predictive accuracy and item catalog coverage (through bias compensation across methods), and the ability to balance multiple objectives (e.g., relevance, novelty, diversity) simultaneously. Although, they introduce significant complexity in terms of system design and implementation, higher computational and maintenance costs, a risk of error propagation if one sub-method performs poorly, and reduced interpretability,

as combining multiple models can obscure the reasoning behind final recommendations, which can hinder user trust [309, 329].

## 3.5. The exploitation of RS in bioinformatics

Motivated by the rapid evolution of bioinformatics and the amount of intricate, heterogeneous data across genomics, proteomics, clinical records, and scientific literature; RS emerged as powerful computational tools, owing to their ability to filter, prioritize, and suggest relevant information based on user-defined needs and implicit patterns, which rendered them indispensable in the various tasks that such domain offer.

### 3.5.1. Data representations exploitable via bioinformatics-oriented RS

The practical utility of RS in bioinformatics depends not only on the availability of biological data, but also on how such data is represented for computational modeling. Raw sequences or chemical structures must be transformed into structured forms (e.g., numerical vectors, graphs, or embeddings) that enable RS to measure similarities, uncover hidden patterns, and perform predictive tasks. In this context, the representations of proteins, nucleic acids, genes, and drugs provide the critical bridge between biological knowledge and algorithmic reasoning. Below, we outline the major classes of representations commonly exploited in bioinformatics-oriented RS.

#### 3.5.1.1. Proteins

In bioinformatics-oriented RS, proteins can be leveraged through the different representations they possess, which include:

**A. Sequence-based representations**

At the simplest level, AA sequences of proteins can be leveraged to map them as numerical feature vectors with learnable parameters via: (1) one-hot encodings, where each residue is represented as a binary vector that is projected into a lower-dimensional, dense latent space via an embedding layer (e.g., the sequence "ACLG" becomes the concatenation or sum of the four embedded vectors for Alanine, Cysteine, Leucine, and Glycine), and (2) $k$-mer frequency representations, which capture local sequence context by counting the occurrences of overlapping subsequences such as di- or tri-peptides (e.g., for $k = 2$, the sequence "ACLG" generates the k-mers "AC", "CL", and "LG"). More specialized descriptors exploit physicochemical indices, such as hydrophobicity, charge, or polarity values from the AAindex database, where each AA in a sequence is mapped into its respective numerical value to create a per-position feature vector. Whereas NLP-inspired protein language models such as ESM-2 and ProtT5 focus on learning the contextual embeddings of proteins via self-supervised training techniques performed on millions of protein sequences, producing single, per-residue high-dimensional vectors that encode structural and functional properties more efficiently [330].

**B. Structure-based representations**

Structure-based representations are predicated on the fundamental principle that the 3D-conformation of proteins determine their biochemical functions, as they capture the precise spatial arrangements of atoms, thereby encoding the geometric and physicochemical properties essential

for interactions. They are typically constructed by converting structural data into computational formats such as (1) contact maps, $N \times N$ symmetric matrices (for a protein with $N$ residues) where each entry indicates if the distance between two residues is within a spatial cutoff threshold (favored by CNN-based architectures), (2) graphs, where nodes represent residues or atoms and edges encode spatial proximity or chemical bonds, with node features capturing attributes such as the chemical properties of AA, whereas edge features denoting distances or angles (favored by graph-based frameworks), and (3) binding-site focused representations such as molecular fingerprints or voxelized grids, which summarize the shape, electrostatics, and hydrophobicity of ligand-binding pockets to support applications like virtual screening and target-based drug recommendations [331, 332].

#### 3.5.1.2. Drugs

Similar to proteins, drug molecules possess numerous representations that include:

**A. Chemical structure-oriented representations**

Through these representations, drugs are described by their atomic composition, connectivity, and bonding patterns, typically encoded as either (1) linear notations such as SMILES strings (Simplified Molecular Input Line Entry System) or in InChI formats (international chemical identifier), which represent molecules as text-based sequences of atoms and bonds (e.g., aspirin's SMILES representation is "CC(=O)OC1=CC=CC=C1C(O)=O", and its InChI format is "1S/C9H8O4/c1-6(10)13-8-5-3-2-4-7(8)9(11)12/h2-5H,1H3,(H,11,12)") that can be processed by NLP-inspired architectures like RNNs and transformers; (2) molecular fingerprints, which are fixed-length bit vectors that encode the presence of substructures, topological paths, and physicochemical properties; or (3) molecular graphs, where atoms are represented as nodes (with features like atom type), and bonds as edges (with features like bond order), enabling graph-based frameworks to learn embeddings that capture hierarchical, and chemical features from both local functional groups and global molecular structures. Collectively, these representations enable RSs to quantify structural similarity, infer potential DTIs, and support drug repurposing by mapping chemical features to biological activity profiles [333].

**B. Physicochemical property-oriented representations**

These representations describe compounds through the quantitative features derived from their molecular structures, reflecting intrinsic properties related to chemical reactivity, solubility, and pharmacokinetic potentials. They typically capture key properties such as molecular weight ($MolW$), $logP$ (octanol-water partition coefficient, indicating lipophilicity), topological polar surface area ($TPSA$), counts of hydrogen bond donors and acceptors ($HBD$, $HBA$), number of rotatable bonds, and $pKa$ values, collectively influencing a drug's ADMET characteristics. In RS, these properties are expressed as numerical feature vectors or used to define a feature space, allowing algorithms to model similarities among drugs with shared bioavailability or safety profiles, while correlating them with biological activities and therapeutic outcomes for tasks like toxicity prediction and ADMET-aware drug repurposing [334].

#### 3.5.1.3. Genes

They are represented in multiple ways that can be efficiently leveraged by RS for tasks such as gene-disease association prediction, drug-gene interaction modeling, and personalized medicine. The main categories of gene representations include:

**A. Sequence-based representation**

These forms provide a fundamental abstraction of genes by encoding the linear arrangements of nucleotides (i.e., A, T, C, G) in a DNA sequence, typically achieved through (1) one-hot encodings, where each nucleotide is mapped into a unique binary vector, preserving precise positional information, but lacking any inherent biochemical or evolutionary context; (2) $k$-mer frequency or embedding representations, where sequences are decomposed into overlapping subsequences of $k$-length (e.g., the sequence "ATGA" yields the 3-mers "ATG" and "TGA"), a method that captures local compositional biases and motifs; and (3) genomic language model-inspired approaches (e.g., DNABERT), which leverage transformer architectures pre-trained on large-scale genomic corpora to generate context-aware, numerical embeddings that encapsulate complex biochemical properties and long-range functional dependencies [335, 336].

**B. Expression-based representation**

These representations characterize genes according to their quantitative expression activities across diverse biological conditions, providing a dynamic and context-dependent layer of information uncaptured by static DNA sequence data. Expression data are typically obtained from high-throughput transcriptomic technologies such as RNA sequencing (RNA-seq) or microarrays. Common approaches include: (1) baseline expression profiles, in which each gene is encoded as a numerical vector, where features correspond to measured expression levels across multiple samples or conditions (e.g., raw read counts, normalized counts, transcripts per million), (2) differential expression profiles, which represent genes by statistically significant changes in expressions across distinct experimental conditions (e.g., diseased versus healthy states), thereby emphasizing condition-specific regulatory effects; and (3) co-expression-based representations, obtained by computing pairwise gene-gene similarity measures (e.g., Pearson or Spearman correlation coefficients) across large cohorts of samples, resulting in networks where nodes correspond to genes and edges encode strong co-expression relationships, effectively capturing functional modules and high-level pathway organizations [337, 338].

**C. Functional representation**

This type of representation conceptualizes genes as functional entities characterized by their biological roles, pathways, and phenotypic outcomes, typically derived from (1) curated knowledge databases that encode gene functions through structured, controlled vocabularies describing biological processes, molecular functions, and cellular components, enabling its encoding as a binary or semantic similarity-based vector; (2) pathway-based resources (e.g., KEGG, Reactome) that map genes to specific steps within metabolic and signaling pathways, enabling the computation of functional similarity based on shared pathway membership; and (3) disease association databases (e.g., DisGeNET) that enable genes to be represented by their known

and predicted links to pathologies, thereby situating them within a clinically relevant functional context. Such representations are particularly powerful because they integrate prior biological knowledge, directly supporting the prediction of gene-gene functional relationships, and the identification of novel gene-disease associations and therapeutic targets [339-341].

#### 3.5.1.4. Heterogeneous interaction networks

Heterogeneous interaction networks aim to capture the functional associations of biomolecules (typically obscure in sequences or structures) by modeling them alongside other types of biomolecules or events as nodes within large-scale biological networks, where edges reflect diverse relational data such as intermolecular interactions (e.g., PPIs, DPIs), genetic interactions, co-expression, or associations with pathways and diseases. Through graph-based ML techniques (e.g., node2vec, GraphSAGE), continuous vector representations for proteins, drugs, genes, pathways, and diseases can be generated; encoding both local interaction neighborhoods alongside broader topological roles of within biological systems, which can be crucial in RSs for tasks like intermolecular interaction prediction and gene-disease association, as they directly exploit the relational and behavioral information inherent to biological systems, thereby facilitating the anticipation of novel interactions within a biological interaction matrix [342, 343].

### 3.5.2. Mapping bioinformatics problems as recommendation tasks

This process involves a formal conceptual translation where the goal of discovering novel associations among biological entities is reformulated as the problem of predicting user preferences for items in a recommendation task, leveraging either the explicit characteristics defining individuals or the implicit relationships binding them within biological systems. The key to a successful mapping involves a systematic deconstruction of the biological problem into the core components of an RS (i.e., atoms), mainly incorporating [344, 345]:

#### 3.5.2.1. Defining entities

This step involves identifying the two central entities that the studied biological phenomenon revolves around, where one entity is playing the role of the user, while the other being the item (e.g., in DTIs prediction, proteins are set to be users, while drugs are set to be items, or vice versa).

#### 3.5.2.2. Representing biological data

The selected two entities involved in the studied phenomena ought to be represented by appropriate forms that can be processed by the deployed RS, whether through the explicit attributes of individuals (favored by CB-based approaches), their implicit interactions (favored by CF-based approaches), or a combination of both.

#### 3.5.2.3. Constructing interaction matrix or graph

Following the identification of the involved entities and their most suitable representations is the phase of collecting biological knowledge involving both entities in the form of (1) a structured sparse interaction matrix where rows represent users, columns represent items, and entries $r_{ui}$ depict explicit or implicit feedback values (e.g., known DTI values, genes co-expression data), or (2) a heterogeneous graph, where nodes depict users and items with assigned attributes (e.g.,

protein AA sequences, drug SMILES representations), and edges denote known interactions, their strengths and affinities , or similarities among individuals.

#### 3.5.2.4. Applying an appropriate RS

The proper choice of RS-related atoms facilitates the process of selecting a RS that best solves the problem at hand, enabling the efficient learning of the latent features of individuals alongside the complex patterns within data, while alleviating sparsity through inferring the vast uncharted portion of the interaction matrix (or graph), and transforming biological discovery into a matrix completion or a link inference problem.

#### 3.5.2.5. Validating results

The results provided by the deployed RS ought to be validated via cross-validation techniques, benchmark approaches and datasets, and external experimental evidence, ensuring the robustness and biological relevance of the estimated outcomes before projecting them from *in selico* to *in vivo* or *in vitro* environments.

Collectively, this formalization allows for the direct application of sophisticated CF, CB, and hybrid recommenders across diverse biological tasks, transforming challenges in gene-disease association, intermolecular interaction estimation, and protein function annotation into computationally tractable problems that generate ranked candidate lists of plausible biological hypotheses for experimental validation.

### 3.5.3. The applications of RS in bioinformatics

The applications of RS in the field of bioinformatics are diverse, which include:

#### 3.5.3.1. DTIs and PPIs in selico

In the context of bioinformatics applications, both DTI and PPI prediction tasks are well-suited to be formulated as recommendation problems, as their known interactions can be represented as (1) bipartite or homogeneous graphs where the core objective is to infer missing links (i.e., link prediction), or (2) as interaction matrices that a RS aims to fill (i.e., matrix completion task). Specifically, in DTI prediction tasks, drugs and proteins are treated as two disjoint entity sets, with their experimentally validated interactions form the entries of an interaction matrix (or the edges linking entities in a bipartite graph). In contrast, PPI prediction tasks focus on a single entity set (proteins) that occupies both sides of a homogeneous association graph, where RS frameworks are leveraged to predict novel or context-specific interactions within the complex cellular network. The automation of these problems as recommendation tasks provides significant advantages, encompassing the systematic and scalable discovery of interactions beyond the limitations of costly and time-consuming *in vivo* and *in vitro* experiments, facilitating drug repositioning and pharmacology analysis of DTIs, while enhancing functional module detection and disease mechanism elucidation for PPIs. Furthermore, the integration of heterogeneous biological knowledge (e.g., protein sequences, drug structures) into these recommendation models enables more accurate predictions, thereby advancing precision medicine, rational drug design, and the understanding of molecular networks underlying health and disease [344].

#### 3.5.3.2. Disease-gene association predictions

This type of problems can be translated as a recommendation task via projecting diseases and genes as two distinct entity types within an interaction matrix or a bipartite (heterogeneous) knowledge graph, where a RS seeks to score and rank candidate gene associations for a given disease based on known interactions, functional similarities, and multi-omics evidence. Automating this task using RS paradigms such as CF, graph-based, matrix factorization, and CB approaches offers significant advantages, including the rapid and systematic prioritization of candidate genes from large-scale genomic datasets, and the dynamic integration of heterogeneous information sources such as PPIs, pathway memberships, and phenotypic data. This integrative capability allows RS to accelerate the identification of novel disease-related genes, improve the interpretability of complex disease mechanisms, and guide translational applications, including biomarker discovery and the development of personalized therapeutic strategies. Furthermore, framing disease-gene prediction as a recommendation problem provides scalability for studying rare and complex diseases, where experimental data is inherently sparse, thereby directly supporting precision medicine initiatives through more comprehensive and accurate genotype-phenotype mapping [346].

#### 3.5.3.3. Drug repurposing

When framed as a recommendation task, new therapeutic indications for existing drugs can be deduced via leveraging known drug-disease, drug-target, and disease-gene associations within heterogeneous biomedical knowledge graphs via RSs that model drugs, diseases, and targets as nodes, while exploiting interaction patterns, similarity measures, and graph-based or language model-based embeddings to predict novel, plausible drug-disease links. Automating drug repurposing through RS paradigms offers significant advantages over traditional trial-and-error strategies, including radically reduced development costs, accelerated timelines, and the ability to systematically mine latent therapeutic opportunities from large-scale omics, electronic health records, and clinical trial data. Therefore, advancing translational research by prioritizing candidate drugs for emerging diseases (e.g., COVID-19), and enabling personalized medicine by matching drug profiles with patient-specific molecular and phenotypic characteristics. Additionally, RS-driven repurposing enhances the exploitation of existing pharmacological knowledge, thereby expanding the therapeutic arsenal without the high costs and risks associated with *de novo* drug discovery [347].

#### 3.5.3.4. Personalized therapeutic strategies

This problem can be effectively reconstructed as a recommendation task by tailoring drugs, treatments, and intervention suggestions to individual patients based on their multi-omics profiles and clinical data, thereby moving beyond population-level generalizations toward precise medicine, accomplished through the implementation of a RS that leverages similarities across molecular profiles, disease states, and historical treatment outcomes to infer optimal individualized therapies, in an environment where patient profiles represent users, and therapeutic interventions depict items. Automating this process offers several merits, including the ability to integrate complex and heterogeneous biomedical data, alleviating the reliance on inefficient trial-and-error

treatments, and accelerating the discovery of patient subgroups that exhibit similar molecular responses to specific therapies. Additionally, this approach has the potential to advance the development of adaptive treatment regimens, improve the stratification of patients for clinical trials, and enable more efficient repurposing of existing drugs for niche patient groups, ultimately reducing adverse effects and enhancing therapeutic efficacy and safety [348].

#### 3.5.3.5. Pathway and functional annotation prediction

Modeling this problem as a recommendation task involves projecting proteins and genes as users, and pathways or functional categories as items, with the goal of the implemented RS manifested in scoring and ranking plausible functional annotations based on molecular similarities, network context and proximity, and interaction patterns. In this setup, RS leverage heterogeneous biological data, encompassing PPI networks, co-expression profiles, and sequence-derived evolutionary signals, to infer missing or novel annotations in a scalable and data-driven manner, offering advantages such as alleviating the labor-intensive burden of manual curation, reducing species-specific annotation bias, and enabling continuous knowledge updates as new experimental data become available. The effective implementation of RS in such task accelerates systems biology research by facilitating the discovery of novel pathway memberships, improving the functional characterization of unannotated genes, and enhancing the interpretability of high-throughput omics datasets, ultimately contributing to a more comprehensive understanding of complex disease mechanisms and the identification of novel therapeutic targets [349, 350].

#### 3.5.3.6. Metagenomics

Challenges in this field can be effectively framed as recommendation tasks that aim to predict associations among microbial taxa (i.e., groups of microorganisms classified based on shared characteristics), functional genes, and environmental or host phenotypes through leveraging heterogeneous data derived from large-scale sequencing studies. In this context, RS can infer plausible gene functions from microbial communities, predict host-microbiome interactions, and identify novel microbial taxa linked to specific conditions based on patterns of co-occurrence, genomic similarity, and functional profiles across samples. Thus providing advantages such as handling the extreme sparsity and high dimensionality of metagenomics data, enabling the scalable identification of functionally relevant microbes and genes without reliance on culture-based methods, and supporting the integration of multi-omics data. This approach ultimately advances fields like microbiome-driven diagnostics, and therapeutic microbiome engineering by uncovering novel microbial biomarkers, predicting ecosystem functions, and prioritizing candidate microbes or genes for targeted experimental validation [351, 352].

#### 3.5.3.7. Scientific literature recommendation

This class of recommendation tasks leverages RS frameworks to help researchers efficiently navigate the vast and rapidly growing volumes of biomedical publications by modeling researchers as users, whereas scientific articles, datasets, or preprints as items, with their interactions encompassing past reading histories, citation patterns, and collaborations among authors. Automating this process provides key advantages, including the reduction of information overload,

the enhancement of discovery for relevant yet overlooked studies, and the fostering of critical interdisciplinary connections such as ML, proteomics, and systems biology [353, 103].

## 3.6. Summary of the most important contributions of RS approaches in bioinformatics across intermolecular interaction prediction tasks

Recommender approaches have become increasingly prominent in intermolecular interaction prediction tasks mainly due to their strong ability to exploit latent patterns embedded in large-scale, heterogeneous biological datasets. In practice, the most commonly applied RS models are those that have successfully proven their effectiveness in capturing hidden associations, handling sparse interaction matrices, providing interpretable outcomes, and integrating multimodal biological information, encompassing matrix factorization (MF) techniques, DL-based architectures, graph-based recommenders, alongside kernel-based approaches. The choice of RS methods for intermolecular interaction prediction however depends heavily on the characteristics of the underlying data and the specific biological questions being addressed. For instance, when datasets are highly sparse and incomplete, MF methods tend to perform well due to their ability to learn low-dimensional latent representations. In contrast, graph-based recommenders are more suitable when rich and relational structures exist (e.g., drug-protein-disease heterogeneous networks), as they can leverage topological context and multi-hop connectivity. DL-based models on the other hand excel when recommendation problems involve highly-complex, multimodal biological features. Whereas kernel-based methods are more appropriate when high-quality biological similarity data are available. Less commonly used RS approaches, such as simple popularity models or pure CB filtering tend to underperform in intermolecular interaction prediction tasks as they fail to capture the intricate, nonlinear dependencies characterizing biological systems. Ultimately, selecting an appropriate RS for an intermolecular interaction prediction task requires a careful evaluation of data sparsity, network complexity, available features, and computational constraints, as each approach offers distinct strengths and limitations.

### 3.6.1. Matrix factorization-based techniques

MF models are fundamentally employed in diverse fields due to their proficiency in dimensionality reduction, effectively projecting high-dimensional and sparse user-item interaction matrices into lower-dimensional latent spaces where users and items are represented as dense vectors, thus uncovering latent features that help explain observed interactions and deduce unobserved ones.

This core principle, formally established in the foundational MF framework of Koren et al., [354], has shifted the overall RS paradigm from neighborhood-based methods toward latent factor models, proving themselves to be capable of capturing complex, hidden patterns in user-item interactions, via factorizing the user-item rating matrix into low-dimensional latent vectors, allowing user preferences and item attributes to be represented in a shared latent space, where their inner product estimates the likelihood of their interactions. This approach not only addressed the issue of data sparsity but also enabled the modeling of subtle factors such as user tastes and item characteristics typically uncaptured by simple similarity-based methods. In the field of bioinformatics, these MF principles can be leveraged to predict intermolecular interactions and associations, where biological entities are represented in latent vectors and their interactions

inferred through matrix reconstruction, which further demonstrates the broad applicability of Koren's MF model. The success achieved by the aforementioned approach across numerous domains resulted in the development of diverse variants, with Multiple Similarity Collaborative Matrix Factorization (MSCMF) being among those that tackled DTIs in the bioinformatics field. Originally introduced by Xiaodong et al., [355], MSCMF was distinguishable from other MF-based techniques (as they rely solely on interaction data) by its ability to effectively leverage complementary biological information through explicitly incorporating multiple similarity measures (in the form of matrices) of both drugs and targets into its framework to help guide the learning process of their latent representations towards capturing biological relationships among entities alongside their behavioral patterns, ultimately rendering it capable of surpassing six state-of-the-art approaches (i.e., BLM, PKM, NetLapRLS, RLS-GIP, RLS-GIP-K and KBMF2K) in terms of area under the precision recall curve (AUPR) across four benchmark datasets (i.e., Nuclear Receptor, GPCR, Ion Channel, Enzyme) where it scored 0.673, 0.773, 0.937, and 0.894 respectively, highlighting the effectiveness of integrating heterogeneous biological similarities within MF frameworks form more prominent DTI prediction accuracy. Building on the same foundation, Liu et al. proposed Neighborhood Regularized Logistic Matrix Factorization (NRLMF) [356], an approach that extends traditional MF by tailoring a framework that's capable of performing binary DTI prediction tasks through employing a logistic function that directly models the probability of interactions among drugs and targets. What makes NRLMF stands out is its differential treatment of observed and unobserved interactions (experimentally valid positive interactions are weighted more heavily than unobserved ones), acknowledging the reliability of positive interactions over the uncertainty in negative ones. Additionally, its integration of local structural information (via imposing neighborhood regularization terms on the learned latent representations) ensures that biologically similar entities obtain consistent embeddings, thus helping it overcome data sparsity problems common in DTI datasets, while enhancing the biological plausibility of its predictions via incorporating prior domain knowledge. Moreover, by solely focusing on the nearest neighbors rather than all similarity relations in its neighborhood regularization; NRLMF reduces noise and improves its predictive accuracy to the point of rendering it able to consistently outperform other neighborhood, kernel, and MF-based state-of-the-art approaches (i.e., NetLapRLS, KBMF2K, BLM-NII, WNN-GIP, and CMF) in terms of area under the curve (AUC) (scored 0.95, 0.969, 0.989, and 0.987) and AUPR (scored 0.728, 0.749, 0.906, and 0.892) scores across four benchmark datasets (i.e., Nuclear Receptor, GPCR, Ion Channel, Enzyme), with its predictions being further validated against updated biological databases such as ChEMBL, DrugBank, KEGG, and Matador. Additionally, in the context of computational drug repositioning for COVID-19, Xu et al. [357] developed Weight Regularization Matrix Factorization (WRMF), offering a novel framework that integrates similarity information with weight regularization to help improve the reliability of estimated DTIs. Unlike conventional MF approaches that treat all interactions equally, WRMF incorporates similarity-constrained weights (extracted form molecular homogeneity and similarity networks) that ensure the strong influence of higher biological similarity values over learned latent representations, alongside a regularization term that augments the influence of observed interactions over unobserved ones. This design alongside the substitution of singular value decomposition (SVD) with the gradient descent algorithm allows the model to better capture the underlying pharmacological and genomic

relationships, while reducing the risk of overfitting to noisy or sparse data, consequently rendering it capable of outperforming five state-of-the-art MF methods (i.e., IMC, CMF, MBiRW, BNNR, NIMCGCN), while providing robust predictions that are concerned with repositioning existing drugs against SARS-CoV-2, which aligned with compounds later reported in external COVID-19 drug databases.

Aside from bioinformatics-oriented MF approaches; other researcher focused developing universal methods, such the collective matrix factorization (CMF) approach introduced by Singh et al. [358] that extends traditional MF by learning latent representations through simultaneously factorizing multiple matrices that share common entities and the various independent relations binding them, enabling it to seamlessly integrate heterogeneous cross-domain knowledge, while making it particularly suited for complex bioinformatics applications where multi-relational data is crucial for predicting novel biomedical associations. The unique nature of CMF enables it to effectively mitigate data sparsity problems in any single matrix through leveraging the auxiliary information embedded within other related matrices (e.g., drug-drug similarity matrices, protein-protein similarity or interaction matrices). Such robustness rendered CMF capable of outperforming two state-of-the-art approaches (i.e., pLSI-pHITS, Pop) in terms of precision scores across both dense and sparse user-movie rating datasets, underscoring its flexibility and generalizability in bridging RS methodologies with complex data integration challenges. In contrast to the aforementioned efforts, Rendle et al. [359] focused on redefining the optimization criterion while developing their Bayesian-based approach named BPR-$O_{PT}$, a maximum posterior estimator that is theoretically grounded in the principle of maximizing the posterior probability that a user prefers a consumed item over a non-consumed one. And to efficiently optimize such objective, the authors introduced the LearnBPR algorithm, a stochastic gradient descent variant that employs bootstrap sampling of training triples, significantly enhancing convergence speed and scalability compared to standard gradient-based optimization techniques. Empirical validation of BRP-$O_{PT}$ across the Rossmann and Netflix DVD rental datasets demonstrated state-of-the-art performance capabilities, consistently surpassing other learning methods (i.e., SVD-MF, WRMF, Cosine-kNN) in terms of AUC score values for implicit feedback, underscoring the influence of the optimization criterion in the overall performance of the implemented recommender approach. On the other hand, Kabbur et al. [360] aimed to extend factored item-based methods to top-N recommendation trough introducing FISM (i.e., Factored Item Similarity based Method), an item-centric paradigm that effectively estimates a user's preference for an item via factorizing an item-item similarity matrix, while incorporating it in its optimization process. Specifically, FISM decomposes the large item-item similarity matrix into two lower-dimensional item latent factor matrices, drastically reducing the number of parameters, mitigating overfitting, and enabling the efficient capture of nuanced, transitive item relationships that are not apparent in raw data, all while considering users as weighted aggregations of the latent factors of all items they interacted with. This unique perspective enables FISM to outperform five state-of-the-art MF methods (i.e., itemKNN, PureSVD, BPRMF, BPRkNN, SLIM) across three benchmark datasets (i.e., MovieLens 100K, Netflix, Yahoo) in terms of hit rate (HR) and average reciprocal hit-rank metric values, establishing itself as a robust and efficient framework for generating precise top-N recommendations across both general purpose and domain-specific datasets. Finally, in order to overcome the limitations of the SVD-based MF approaches, BLS-SVD++ was proposed by Wang

et al. [361] as an enhancement to the traditional SVD++ algorithm, which integrates a backtracking line search (BLS) mechanism into the optimization process in order to improve both convergence speed and prediction accuracy. Specifically, BLS-SVD++ leverages the BLS techniques in order to adaptively adjust its learning rate in each iteration based on gradient information, ensuring more stable and efficient optimization without manual parameter tuning. By improving the optimization foundation of SVD++, BLS-SVD++ is able to outperform other SVD-based variants (i.e., SVD, SVD++, RSVD) across three benchmark datasets (i.e., MovieLens 1M, MovieLens 10M, FilmTrust) in terms of root mean square error (RMSE), mean average error (MAE), and Recall values, while requiring fewer iterations to converge, underscoring the predictive capabilities of MF approaches in scarce recommendation environments.

Collectively, MF-based approaches provide remarkable adaptability in learning the implicit behaviors of users and items, enabling them to provide robust recommendations with reasonable flexibility and scalability across large datasets, despite being limited by:

- ✓ **Linearity assumption:** They primarily rely on dot products to capture interactions in a linear latent space, which is often insufficient for modeling the complex, non-linear relationships characterizing real-world user-item interactions, especially in the bioinformatics field.
- ✓ **Hyperparameter sensitivity:** Their performance is highly dependent on the precise tuning of hyperparameters, particularly the dimensionality of the latent space and regularization coefficients, which requires computationally expensive procedures like cross-validation, which renders them unfeasible for large-scale datasets.
- ✓ **Cold-start problem:** Like most CF methods, they perform poorly for new users or items with no interaction records, as there is insufficient data to learn an accurate latent representation from (in bioinformatics-oriented recommendation tasks, they struggle to provide robust recommendations for orphan and mutated proteins, as well as novel drugs).
- ✓ **Interpretability:** The latent factors they learn are abstract and lack intuitive meaning, making it difficult to elaborate the decision-making process behind their provided recommendations (which features of a drug led to a predicted interaction with a target).

### 3.6.2. Deep learning-based models

DL-based RS approaches have revolutionized recommendation tasks by leveraging multi-layered neural architectures to automatically learn complex, non-linear representations of users and items, bypassing the linear constraints of MF approaches, while providing the possibility to efficiently exploit the inherent attributes of both users and items for more prominent recommendations.

The Neural Matrix Factorization (NeuMF) model developed by He et al. [362] pioneered this shift by unifying the strengths of MF techniques and neural networks within a single end-to-end neural architecture that's capable of overcoming the former's linearity assumption through the latter's ability to learn the complex, non-linear interaction functions, thereby capturing high-order patterns that linear models fail to grasp. NeuMF consists of two complementary components: a generalized matrix factorization (GMF) that preserves the linear inner product structure for more efficient latent similarity modeling, and a MLP that non-linearly transforms embeddings through

multiple hidden layers to capture intricate user-item relations, with the outputs of both components being aggregated as a joint representation that balances generalization and flexibility, consequently improving the overall accuracy across implicit feedback datasets, while providing a scalable framework that can adapt to different levels of interaction complexity, as demonstrated with its ability to outperform numerous state-of-the-art approaches (i.e., ItemPop, ItemKNN, eALS, and BPR) across both MovieLens (0.73, 0.447) and Pinterest (0.877, 0.552) datasets, where it demonstrated superior HR and NDCG (i.e., normalized discounted cumulative gain) score values. The success achieved by NeuMF inspired numerous researchers to delve more into the process of exploiting DL techniques in recommendation tasks, with the outer product-based Neural Collaborative Filtering (ConvNCF) model [363] being among the first approaches to redefine how pairwise interactions are modeled, replacing the standard inner product with an outer one, and creating structured 2D interaction maps that preserve the embedding-level relations between the vector representations of users and items, while providing the opportunity to leverage CNN architectures as pattern detectors, supporting the capture of meaningful, localized feature correlations usually lost in the compressed scalar outputs of inner products, alongside providing a more expressive and discriminative model of user-item affinity. The effective application of CNNs on the generated interaction maps enabled ConvNCF to demonstrate superior performance over strong baselines (i.e., ItemPop, MF-BPR, NeuMF, and JRL) in top-N recommendation tasks across both Yelp and Gowalla datasets in terms of HR (i.e., 0.7936, 0.3086) and NDCG (i.e., 0.5826, 0.16) score values, establishing a novel paradigm that moves beyond simple vector combining operations to a more detailed, feature-wise interaction analysis for collaborative filtering. Following the same path, Han et al. [364] proposed Stacked Embedding Convolutional Neural Collaborative Filtering (SECNCF) model that extends ConvNCF through introducing pedrails that enable stacking the feature vectors of users and items to form unique interaction representations that can also be processed by CNNs, allowing the model to learn rich, complex, and high-order sets of latent characteristics across multiple subspaces simultaneously. This architectural advancement enables SECNCF to consistently outperform other state-of-the-art approaches (i.e. ItemPop, JRL, ConvNCF, GMF, MLP, and NeuMF) in terms of HR (i.e., 0.6454, 0.6968, 0.7352) and NDCG (i.e., 0.4643, 0.511, 0.4979) score values in top-N recommendation tasks across ACV, AKS, and AMT benchmark datasets respectively, underscoring the efficiency of learning stacked embeddings as a means to significantly boost the predictive power of deep learning-based RS. Moreover, Sedhain et al. [365] developed an AE-based CF paradigm named AutoRec, which treats recommendation tasks as non-linear, item-based or user-based vector reconstruction tasks (rather than regular matrix factorization problems) that are performed via a single-layered AE framework. AutoRec starts by projecting partially observed interaction vectors (either for a user or an item) through a bottleneck layer to help learn their respective compressed latent representations, which are then non-linearly reconstructed to predict missing entries, effectively capturing complex, high-order correlations and dependencies within the interaction network that linear models fail to grasp. This design allows AutoRec to outperform baselines such as RBM-CF, BiasedMF, and LLORMA in terms of RMSE score values (i.e., 0.831, 0.782, 0.823) across MovieLens-1M, MovieLens-10M, and Netflix benchmark datasets, validating its ability to achieve state-of-the-art accuracy while remaining computationally efficient, thus establishing AEs as foundational frameworks for modern RS. Similar to the aforementioned work, Wu et al. [366]

introduced CDAE (i.e., Collaborative Denoising Auto-Encoder), an architecture designed specifically to learn robust user-item representations from incomplete and noisy interaction data through corrupting user-item latent vectors with random noise before feeding them to its encoder. This process compels CDAE to learn latent representations that can efficiently reconstruct the original uncorrupted vectors, which improves generalization and resilience to missing or noisy data. Additionally, this approach explicitly incorporates a user-specific latent vector into its hidden representation, which enables the model to personalize recommendations more effectively by combining global item features with individualized user preferences. This integration of denoising regularization and user embeddings allows CDAE to capture higher-order, non-linear patterns spread across the implicit feedback data, while mitigating overfitting, resulting in its consistent outperformance of POP, ITEMCF, MF, BPR, and FISM baseline models across MovieLens-10M, Netflix, and Yelp benchmark datasets in terms of mean average precision score values.

Aside from the universally applicable DL-based RS, Öztürk et al. [367] focused on developing a DTI-oriented RS named DeepDTA, an end-to-end DL architecture that directly processes the raw sequential representations of molecules (i.e., SMILES, AA sequences), enabling the automatic learning of task-relevant molecular features directly from data, while eliminating the need for handcrafted molecular descriptors or structural information that other approaches heavily relied on. DeepDTA consists of two parallel branches (each consisting of three consecutive CNN layers followed by a max-pool layer) that independently process targets and drugs to extract localized structural motifs, followed by a MLP block that processes the concatenations of both vector representations to help model complex, non-linear interactions between involved biological entities, thereby enabling the accurate estimation of interactions' continuous binding affinity values. This scalable and generalizable framework allows DeepDTA to capture complex, high-order relationships among molecules that traditional approaches fail to grasp, while rendering it capable of achieving superior predictive accuracy in binding affinity prediction tasks over both Kron-RLS and SimBoost baseline models in terms of concordance index (CI), mean squared error (MSE), and AUPR score values across both Davis (0.878, 0.261, 0.714) and KIBA (0.863, 0.194, 0.788) benchmark data sets respectively. Additionally, inspired by the aforementioned approach, Karimi et al. [368] introduced DeepAffinity, an interpretable DL-based approach that's capable of simultaneously capturing sequential dependencies and local structural motifs from raw 1D molecular representations of data (i.e., SMILES and structural property sequence (SPS) representations of both drugs and proteins). DeepAffinity accomplishes affinity estimation tasks via an architecture that consists of two parallel blocks (with each stacking GRUs, attention mechanisms, and CNNs) that're followed by a fully-connected layer that processes their concatenated outputs, enabling the efficient exploitation of domain-related knowledge in order to provide more accurate estimations concerning DTI affinities, while providing meaningful insights regarding the estimated outcomes. After an unsupervised pre-training phase, followed by a fine-tunning one that involves four benchmark datasets (i.e., Estrogen Receptors, Ion channels, GPCR, Tyrosine kinase), DeepAffinity is capable of outperforming several baseline approaches (i.e. ridge regression, lasso regression, and random forest) in terms of RMSE score values. Similar to the aforementioned approach, Öztürk et al. [369] were inspired by DeepDTA to build WideDTA, a refined DL architecture specifically tailored for word-based drug-target binding affinity prediction that expands the representational scope of molecular data by incorporating multiple textual

representations beyond raw sequences (i.e., AA sequences and domain/motif features (PDM) for targets, and SMILES sequences and maximum common substructures (LMCS) for drugs) for more enriched input space. Each of these four input types is separately encoded through one of four CNN branches (each composed of two consecutive 1D-convolutional layers followed by a max pooling layer) that extract localized structural motifs, after which their latent representations are concatenated and passed through an MLP block to predict continuous binding affinity values. This broader representation learning framework enables WideDTA to capture both fine-grained sequence-level dependencies and higher-level semantic patterns that are often overlooked in both character-based and purely sequence-based models. As a result, the model achieves improved predictive accuracy over DeepDTA, KronRLS, and SimBoost state-of-the-art baselines across both Davis and KIBA benchmark datasets in terms of CI (0.886, 0.875) and MSE (0.262, 0.179) score values respectively, underscoring the utility of combining multiple complementary molecular descriptors in DTI prediction tasks.

On the other hand, the success of NLP techniques across various domains, pushed Wan et al. [370] to develop DeepCPI, a large-scale DL framework that combines unsupervised learning techniques (through exploiting latent semantic analysis and Word2vec to learn prominent representations of proteins and drugs from large corpora of 1D sequential representations) with MLPs for more accurate estimations, while ignoring both structural features and handcrafted similarity measures. Such scalable framework enables DeepCPI to capture both local structural motifs and global dependency patterns, allowing it to flexibly learn high-order associations that other approaches fail to identify, while rendering it capable of demonstrating superior predictive accuracy across multiple benchmark datasets (i.e., ChEMBL, BindingDB) in terms of area under receiver operating characteristic curve (AUROC) and area under the precision curve (AUPC) score values over approaches like DTINet, NetLapRLS, DTINet, random forest, SLNN, and DNN with conventional features. DeepCPI was also capable of demonstrating strong generalization to large-scale virtual drug screening tasks, where it was able to prioritize biologically meaningful interactions and highlight potential repositioning candidates, underscoring its predictive power, scalability, and applicability to real-world drug discovery pipelines. Moreover, unlike other researchers, Abdel-Basset et al. [371] aimed to integrate hybrid feature representations of biological entities to help capture their behavioral properties more efficiently, resulting in the development of DeepH-DTA, a DL-based DTI prediction model tailored for COVID-19 drug repurposing. This approach incorporates three distinct feature-learning pathways to comprehensively represent DTIs; one processes protein sequences through squeezed-excited dense convolutional networks; another processes drug topological representations via improved heterogeneous graph attention networks that help capture complex structural relationships; and a final one that extracts spatio-sequential features from the drugs' SMILES string using a bidirectional convolutional LSTMs, with all three branches' outputs being concatenated before passing through an MLP block to help estimate an interaction's overall score. This hybrid design enables DeepH-DTA to capture complex non-linear relationships between compounds and proteins, while rendering it capable of outperforming several state-of-the-art approaches (i.e., KronRLS, SimBoost, DeepDTA, MT-DTI(wo-FT), MT-DTI, DeepCPI, WideDTA, GANsDTA, Attention-DTA, GAT, GIN, and DeepGS) in terms of CI, MSE, R squared, and AUPC scores values across both Davis (0.924, 0.195, 0.725, 0.801) and KIBA (0.927, 0.111, 0.799, 0.861)

benchmark datasets. Additionally, DeepH-DTA was able to successfully estimate the binding affinities between SARS-CoV-2 and certain FDA-approved drugs, demonstrating its practical utility in urgent biomedical contexts. Lastly, and based on transformer architectures, Huan et al. [372] developed MolTrans, a DL framework for DTI prediction that leverages both large-scale unlabeled data and supervised training techniques to capture fine-grained biochemical interactions at the sub-structural level. Unlike prior models that rely on global representations, MolTrans decomposes each drug and protein into sub-structures (chemical fragments for compounds, and $k$-mer segments for proteins) that are processed via transformer encoders, with the resulting representations being leveraged to construct an interaction map that quantifies the intensity of interactions. Additionally, to further refine the generated pairwise associations, the model applies a CNN layer to help captures higher-order interaction patterns in the forms of tensors that are processed via a decoder in order to estimate the probability outputs of DTIs. This unique framework enables MolTrans to derive contextualized representations that account for long-range dependencies across sub-structures, while rendering it capable of surpassing baseline approaches such as linear regression, DNN, GNN-CPI, DeepCPI, DeepDTA, and DeepConv-DTI in terms of AUROC, AUPR, sensitivity, and specificity score values across BIOSNAP (0.895, 0.901, 0.775, 0.851), DAVIS (0.907, 0.404, 0.8, 0.876), and Kiba (0.914, 0.622, 0.797, 0.896) benchmark datasets, offering a biologically meaningful framework that aligns computational predictions with molecular binding mechanisms.

Overall, DL-based RSs provide powerful modeling capabilities by overcoming the linearity assumption of traditional MF approaches through the ability to learn complex, non-linear interactions, thereby achieving state-of-the-art performance across implicit feedback, binding affinity prediction, and drug-target interaction tasks. Nevertheless, they remain limited by:

- ✓ **Substantial computational requirements:** Training DL-based models takes long periods of time, as they demand significant GPU and CPU resources, especially when processing large-scale biological datasets or during conducting hyperparameter optimization, ultimately rendering them less practical for large-scale or real-time recommendation and screening tasks.
- ✓ **Data hunger and sparsity sensitivity:** DL-based models require large volumes of high-quality interaction data to generalize well, which is not the case in bioinformatics datasets that are often sparse, noisy, and imbalanced, which may cause overfitting and degraded performances.
- ✓ **Hyperparameter and architectural sensitivity:** The more complex DL-based architectures get, the more sensitive they become to the choice of hyperparameters (e.g., network depth, layer sizes, learning rates, regularization coefficients), since suboptimal settings often lead to unstable or poor results, thereby necessitating extensive tuning strategies through computationally expensive procedures like cross-validation.
- ✓ **Interpretability challenges:** Despite their elevated accuracy, most DL approaches operate as black boxes, providing little insight regarding their decision-making processes, making it difficult to identify impactful features or substructures that drive outcomes, which may limit their trust in biomedical applications, though some approaches (i.e., DeepAffinity, MolTrans)

attempted to address this issue through attention mechanisms that highlight important molecular substructures.

- ✓ **Cold-start and generalization issues:** Universally-applicable DL-based RS struggle with unseen targets and drugs (e.g., orphan proteins, novel drugs), as they do not possess any historical interactions that can be leveraged to generate meaningful representations.
- ✓ **Generalization constraints:** The training techniques leveraged by DL-based approaches render them specialized for witnessed data, while powerful for their specific tasks, often lack transferability across different biological prediction tasks or datasets without significant retraining and architectural modifications.

### 3.6.3. Graph-based recommenders

With an impact similar to that of DL-based approaches, graph-based methods have fundamentally advanced recommendation tasks by explicitly leveraging the properties of graphs to help capture topological, and environmental dependencies among both biological entities and the sub-elements composing them, consequently overcoming the limitations of sequence and matrix-based methods that fail to grasp the structural patterns governing intermolecular interaction outcomes.

Diverging from sequence-dependent models, Luo et al. [373] proposed DTINet, a network integration approach that predicts DTIs by inferring the topological features of drugs and targets through a unified, heterogeneous biological network that incorporates diverse information sources (i.e., DTIs, DDIs, drug-disease associations, drug-side-effect associations, drug-drug similarities, protein-disease associations, PPIs, and protein-protein similarities) while ignoring both sequential and structural features. Specifically, DTINet first applies random "walk with restart" to extract diffusion-based topological features from each network, followed by a dimensionality reduction step using "diffusion component analysis" to learn low-dimensional, noise-robust embeddings that preserve the most informative relationships. These embeddings are then projected into a shared latent space that allows DTINet to infer novel drug-target associations through geometric proximity in the embedding manifold, effectively translating heterogeneous biological knowledge into a unified representation space. Such design enables DTINet to output superior predictive accuracy over competing baselines (i.e., BLM-NII, NetLapRLS, HNM, and CMF) in terms of AUROC and AUPR scores, with several of its top-ranked novel predictions being experimentally validated, confirming its ability to generalize beyond known associations, alongside demonstrating its practical utility in drug discovery and repositioning within complex biological systems. Furthermore, while focusing on exploiting user-item implicit signals, Wang et al. [374] introduced NGCF (i.e., Neural Graph Collaborative Filtering), a universal model that explicitly models the high-order connectivity patterns between users and items via propagating their collaborative signals through the structure of their respective bipartite graph. Specifically, NGCF iteratively updates node representations via performing non-linear message construction and aggregation operations, which helps encode both node features and connection strengths across multiple layers, thus capturing more complex topological patterns. This design enables NGCF to distill collaborative signals directly from the interaction graph, moving beyond the limitations of modeling users and items in isolation, while achieving a more expressive representation that

reflects the nuanced structure of user-item relationships, consequently enabling it outperform several baselines approaches (i.e., MF, NeuMF, CMN, HOP-Rec, PinSage, GC-MC) in terms of Recall and NDCG scores across Gowalla (0.1569, 0.1327), Yelp (0.0579, 0.0477), and Amazon-Book (0.0337, 0.0261) benchmark datasets, establishing a new paradigm for graph-based recommenders and inspiring a generation of subsequent graph neural network models. Additionally, by focusing on trying to efficiently leverage the heterogeneous biological network information, Wan et al. [375] developed NeoDTI, a framework that utilizes neighborhood information aggregation techniques to directly learns topology-preserving, node-level embeddings from a heterogeneous network that comprises multiple interaction and similarity types. Specifically, NeoDTI updates the representation of each node by non-linearly aggregating feature information of its diverse neighbors and the edges connecting them. This process allows NeoDTI to capture multi-relational dependencies, while effectively incorporating biological context across different data modalities into one unified embedding space, which preserves the structural properties of the original network, due the imposed constraints that force the reconstruction of the original individual networks. This learning paradigm enables NeoDTI to achieve superior predictive accuracy over numerous state-of-the-art approaches (i.e., DTINet, HNM, MSCMF, NetLapRLS, DT-Hybrid, BLMNII) in terms of AUPR score values, while successfully highlighting known and novel interactions that were later confirmed through literature evidence. Graph-CNN [376] on the other hand focused on learning the latent representations of protein pockets in a two-staged, structure-aware pipeline that bridges unsupervised structural embeddings with task-specific interaction modeling. Specifically, Graph-CNN trains a graph AE on a set of protein pockets to produce low-dimensional embeddings that preserve both local and global pocket properties, which is accomplished via a greedy two-stage encoder-decoder design that involves compressing local neighborhood information via degree-specific graph convolutional filters, and transforming per-node vectors into node fingerprints whose average yields a single graph-level embedding. Then, two parallel graph-convolutional networks (one for pocket graphs initialized from the AE weights, and one for 2D molecular graphs) learn refined pocket and molecule features, which are then concatenated into a joint target-ligand representation that is processed by an MLP block to predict their respective binding probability. This innovative design enables the model to outperform baselines such as 3DCNN, Vina, andom forest, and NN across both DUD-E (Database of Useful Decoys: Enhanced) and MUV (Maximum Unbiased Validation) benchmark datasets in terms of AUC (0.886, 0.563) score values, underscoring its efficiency at extracting task-specific features for binding affinity prediction from native pocket and ligand graph representations.

In contrast to the aforementioned approach, Lim et al. [377] focused on learning the latent representations of target-drug complexes as whole, introducing a 3D structure-embedded Graph Neural Network (3D-GNN) framework that explicitly incorporates the 3D-binding conformations of protein-ligand complexes for more accurate DTI estimations. Specifically, the model first encodes the 3D geometry of a complex into two distinct adjacency matrices, one only capturing purely covalent bonds, while the other encompassing both covalent and noncovalent interactions, allowing the network to differentiate between intrinsic atomic connectivity and spatially dependent interaction forces that govern molecular recognition. Such structural information is exploited via a distance-aware graph attention mechanism (GAT) with gate augmentation, which adaptively weighs atomic contributions based on 3D distances, consequently stabilizing feature propagation

during training, while improving model convergence and performance. Experimental validation involving 3D-GNN demonstrates its superior performance over several docking and DL-based models in terms of AUROC, adjusted LogAUC, and PRAUC scores across both DUD-E and PDBbind datasets, while confirming its ability to reproducing the natural population distribution of active and inactive molecules. Jiang et al. [378] on the other hand attempted to avoid leveraging expensive graph representations of proteins' 3D-structures via introducing DGraphDTA, a structure-aware framework that integrates GNNs with protein contact maps, enabling the explicit modeling of both molecular and structural dependencies within drug-target pairs, via representing protein graphs using contact maps (obtained via pconsc4 algorithm) derived from their 3D structures. The graphs of both drugs and proteins are then processed using Graph Convolutional Networks (GCNs) that learn hierarchical, and topological features that capture both local chemical environments and global spatial dependencies. By introducing contact map-based protein modeling, DGraphDTA bridges the gap between 1D and 3D information, while significantly outperforming numerous state-of-the-art approaches (i.e., KronRLS, SimBoost, DeepDTA, WideDTA, and GraphDTA) in terms of CI, MSE, and Pearson's correlations metrics across both Davis (0.904, 0.202, 0.867) and KIBA (0.904, 0.126, 0.903) benchmark datasets, underscoring the potentials of contact maps in bridging sequence data with spatial information to advance computational drug discovery. Similar to the aforementioned approach, Nguyen et al. [379] proposed GraphDTA, a drug-oriented DL-based framework that explicitly models the topological structures of drugs as molecular graphs processed via one of four different graph-based approaches (being GCN, GAT, GIN, or GAT-GCN), allowing the model to learn spatially aware embeddings that preserve structural dependencies inherent in molecular chemistry. In parallel, targets are represented as a 1D AA sequences that are processed by three successive 1D-convolutional layers followed by max pooling to capture hierarchical sequence features, with the resulting feature vectors being processed via an MLP block that predicts binding affinity values in a continuous regression setting. Beyond methodological innovation, GraphDTA demonstrated state-of-the-art performance on two benchmark datasets (i.e., Davis, KIBA) over prior baselines (i.e., DeepDTA, WideDTA, KronRLS, and SimBoost) in terms of CI (0.893, 0.882) and MSE (0.229, 0.147) score values, alongside being ranked among the top 10 performers (out of 530) in the IDG-DREAM Drug-Kinase Binding Prediction Challenge. Furthermore, its latent space analysis revealed that hidden node activations correlated with interpretable chemical properties, highlighting GraphDTA's ability to automatically infer chemically meaningful patterns without explicit feature engineering. Son et al. [380] on the other hand challenged the difficulty of collecting and leveraging the 3D structural information of protein-ligand complexes and introduced GraphBAR, a GCN framework that exploits the graph form of protein-ligand complexes (without relying on voxelized grids) that are comprised of multiple adjacency matrices, and a feature matrix that captures the atomic and molecular properties of the binding pocket. The overall architecture of GraphBAR consists of several parallel graph convolutional blocks, each based on different adjacency matrix and composed of three GCN layers interleaved with fully connected and dropout layers that use degree-normalized adjacency matrices for stable aggregation of neighborhood information, which are succeeded by graph gather layer that combines node embeddings into a single global feature vector, with the outputs from all blocks being concatenated and fed into additional dense and dropout layers to predict binding affinities. Such framework enables

GraphBAR to achieve superior efficiency and predictive accuracy compared to the Pafnucy in terms of RMSE, MAE, and Pearson's correlation coefficient across four datasets that are extracted from PDBbind. Moreover, by leveraging data augmentation with docking simulations, the model mitigates limitations arising from small structure-based datasets, leading to consistent performance improvements across benchmarks.

In an attempt to incorporate the predictive capabilities of language models alongside GNNs in PPI tasks, Jha et al. [381] introduced a PPI_GNN, a hybrid graph-language framework that integrates GCN and GAT with pre-trained protein language models (i.e., SeqVec and ProtBert) to jointly capture structural, and sequential information of proteins. Each node (i.e. AA residue) in a graph protein is described via context-aware embeddings that're generated by a language model, allowing the framework to encode biochemical semantics without requiring expert-defined features. These features are then propagated through either a GCN, which aggregates local neighborhood information via spectral convolutions, or a GAT, which employs an attention mechanism to assign adaptive weights to neighboring residues, thereby modeling heterogeneous inter-residue dependencies more flexibly. Empirical results on human and S. cerevisiae PPI datasets revealed promising accuracy, sensitivity, specificity, precision, F-score, Matthew correlation coefficient, AUPRC, and AUROC score values, underscoring the potential residing in combining structural graphs with deep contextual sequence embeddings in bridging the gap between sequence-level representation learning and structure-informed interaction prediction. Furthermore, Zhang et al. [382] proposed DrugMAN, a DTI prediction approach that employs GAT-based integration algorithm adapted from the BIONIC framework to learn network-specific low-dimensional embeddings for drugs and targets from four drug networks and seven gene/protein networks. Specifically, each network is encoded via a block of three sequential GATs that hierarchically aggregate up to third-order neighborhood information, the resulting representations are then combined through a weighted and stochastically masked summation, projected into a latent space (optimized by network reconstruction to ensure high-quality, information-rich representations), and processed via a mutual attention mechanism that models pairwise dependencies and generate a unified joint representation for each drug-target pair. Such unique framework enables DrugMAN to demonstrate consistent results that outperform those of several baseline approaches (i.e., SVM, random forest, DeepPurpose, DTINet, and NeoDTI) in terms of AUROC, AUPRC, and F1-score values across diverse scenarios (i.e., warm start, target-cold, drug-cold, both-cold), highlighting both its efficiency at extracting complementary information from heterogeneous networks, and its generalization capabilities in drug discovery and repurposing-related tasks. Drawing inspiration from NLP models, Yazdani-Jahromi et al. [383] developed AttentionSiteDTI, an interpretable graph-based framework that reformulates DTI prediction tasks as sentence-level relation classification problems, where each protein binding pocket or drug molecule is treated as a word. The model first identifies protein binding sites using a pocket-detection algorithm, represents them alongside ligands as graphs, and processes them via Topology Adaptive Graph Convolutional Networks (TAGCN). Treated as contextual sequences, the resulting embeddings are aggregated via an attention pooling mechanism, before passing through a self-attention bidirectional LSTM module followed by an MLP block to help estimate the interaction likelihood. This design provides interpretable attention maps that highlight the specific residues and molecular regions driving the interaction, consequently rendering it capable

of outperforming several state-of-the-art approaches (i.e., KNN, L2, NN-score, random forest, Vina, Smina, AtomNet, 3D-CNN, GCNs, CPI-GNN, DrugVQA, TransformerCPI, PocketGCN, GNN, and GraphDTA) in terms of AUC and F1-score values across DUD-E, Human, and BindingDB benchmark datasets. Whereas in-lab experimental validation confirmed a strong correspondence between computationally predicted and experimentally observed interactions for several candidate compounds against Spike (ACE2), underscoring its real-world applicability as a reliable pre-screening tool for drug discovery and repurposing. Finally, while focusing on developing an interpretable attention-based approach, Bae et al. [384] proposed GraphATT-DTA, which models local-to-global DTIs trough processing the graph forms of drugs SMILES via GNNs (e.g., GCN, GAT, GIN, MPNN, DMPNN), and the sequential forms of proteins via 1D-CNNs. To learn interaction-specific relationships, GraphATT-DTA employs an attention-based interaction module that constructs a relation matrix that enables quantifying the pairwise importance between atomic substructures and protein subsequences through atom-wise and subsequence-wise Softmax operations. This yields significance-weighted representations that reflect bidirectional attentional influence, allowing each entity to adapt its representation based on the other, consequently rendering GraphATT-DTA capable of demonstrating superior performance over several baselines (i.e., DeepDTA, GraphDTA, Deep-Affinity, ML-DTI, HyperAttentionDTI, and FusionDTA) in terms of CI and MSE scores across Davis (0.904, 0.204) and BindingDB (0.651, 1.582) benchmark datasets. Moreover, its attention mechanism provides interpretable insights regarding drug substructures and protein regions that influence binding strength, offering a transparent and biologically meaningful predictive framework that bridges molecular representation learning with explainable drug discovery.

In general, graph-based RS have significantly advanced intermolecular interaction prediction tasks through the efficient modeling of biological entities via graph structures that enable capturing high-order relational dependencies among drugs, proteins, and heterogeneous biological entities, thereby achieving state-of-the-art performance by leveraging the inherent topology of user-item networks and complex biological systems. Nevertheless, they remain limited by:

- ✓ **High computational costs:** The propagation and aggregation operations across graph neighborhoods require iterative message-passing and large-scale matrix computations (require processing the entire graph structure that is represented as an adjacency matrix), making training and inference computationally expensive, particularly for dense molecular graphs or large heterogeneous biological networks.
- ✓ **Over-smoothing and over-squashing effects:** Deeper GNN architectures tend to blur node representations as layers increase, leading to the loss of discriminative power between distinct molecular substructures or protein regions, thus reducing the model's ability to capture fine-grained biochemical relationships.
- ✓ **Data sparsity and noise propagation:** Graph-based methods heavily depend on well-connected network structures. However, bioinformatics graphs (e.g., drug-protein or drug-disease networks) are often incomplete, sparse, and noisy, which causes unreliable message propagation and feature dilution.

- ✓ **Limited interpretability and biological transparency:** Although some models (e.g., AttentionSiteDTI, GraphATT-DTA) leverage attention mechanisms to highlight key interaction regions, most Graph-based frameworks function as complex black boxes, offering limited interpretability on booth node and edge levels, thus hindering biological validation and trust in their predictions.
- ✓ **Cold-start and inductive generalization issues:** Their heavy reliance on neighborhood information causes them struggle with entities that have few or no connections (e.g., new drugs or orphan proteins), as there is insufficient neighborhood data to generate meaningful representations from, making them ill-suited for tasks that involve novel entities that are absent from the training graph.
- ✓ **Hyperparameter and architectural sensitivity:** The predictive performance of graph-based recommenders is highly dependent on architectural design (e.g., number of layers, neighborhood size, attention heads) and regularization settings, which vary significantly across datasets, causing instability and necessitating extensive hyperparameter tuning to avoid underfitting or overfitting.
- ✓ **Architectural redundancy and limited methodological innovation:** Most graph-based RS exhibit low architectural novelty, as they predominantly rely on existing GNN modules (e.g., GCN, GAT, or MPNN) with only minor structural rearrangements or superficial modifications, thereby limiting genuine methodological creativity and hindering the development of fundamentally new paradigms for biological interaction modeling.

### 3.6.4. Kernel-based estimators

Kernel-based recommenders have significantly contributed in advancing recommendation tasks via leveraging kernel functions to implicitly map data into high-dimensional feature spaces, enabling the capture of complex, non-linear similarities between users and items without requiring explicit feature engineering, thereby providing a powerful mathematical frameworks for modeling intricate relationships in both commercial and biological domains.

Kernelized Bayesian Matrix Factorization with Twin Multiple Kernel Learning (KBMF2MKL) approach [385] pioneered the shift towards kernel methods, with its ability to integrate multiple heterogeneous side information sources through kernel functions by leveraging a fully conjugate Bayesian formulation, which allows for efficient variational inference, making it computationally feasible while maintaining the benefits of full Bayesian uncertainty estimation. Specifically, the model uses kernel-based nonlinear dimensionality reduction on both users and items to extract kernel-specific latent components, which are then linearly combined through multiple kernel-learning to form composite representations (automatically weigh the informativeness of each information source), that are subsequently factorized to predict the interaction matrix. This design enables KBMF2MKKL to demonstrate strong performance in multilabel classification tasks via achieving the lowest Hamming loss on 10 out of 14 benchmark datasets compared to five state-of-the-art baselines, underscoring the efficiency of kernel-based learning in improving predictive accuracy and uncertainty quantification in complex biological and general recommendation tasks. Building on this foundation, Nascimento et al. [386]

introduced KronRLS-MKL, a Multiple Kernel Learning extension of KronRLS (i.e., Kronecker Regularized Least Squares) that integrates multiple heterogeneous information sources to address DTI problems as link prediction tasks. Specifically, KronRLS-MKL combines several base kernels from both drug and target domains through a weighted linear combination that maintains the complementary information across various sources, elevates interpretability through assigned weights (reveal most contributing features), while controlling overfitting through a $L_2$ regularization term, all in a framework that's optimized in a two-step iterative process that alternates between solving for regression coefficients and updating kernel weights. Empirical analyses conducted across four benchmark datasets (i.e. Enzyme, Ion channel, GPCR, and Nuclear receptors), while using ten distinct kernels for drugs and ten for targets demonstrated that KronRLS-MKL achieved higher performance over numerous baselines in terms of AUPR score values, demonstrating the efficiency of its interpretable framework in integrating heterogeneous molecular data in computational drug discovery. Other researcher such a Han et al. [387] developed a universally applicable approach named GLocal-K, a Global-Local Kernel-based matrix completion framework that generalizes sparse, high-dimensional user-item rating matrices in a low-dimensional latent space using only rating data. The model operates in two major stages: (1) a pre-training phase, where an AE is trained using a local kernelized weight matrix constructed via a 2D-RBF kernel; and (2) a fine-tuning stage, where the pre-trained AE is refined using a global kernel-based rating matrix generated through a convolution-based global kernel. This two-tier kernel strategy allows GLocal-K to jointly leverage localized similarity structures during pre-training and global relational patterns during fine-tuning, consequently rendering it capable of outperforming 11 baseline approaches (including models augmented with side information) in terms of RMSE values across ML-100K (0.89), ML-1M (0.822), and Douban (0.721) benchmark datasets, demonstrating the efficiency of localized similarity structures and global relational patterns in producing compact yet information-rich feature representations.

Collectively, kernel-based RS provide mathematically grounded frameworks for modeling complex similarity relationships through kernel tricks, enabling effective integration of heterogeneous data sources through kernel fusion while offering certain degrees of interpretability. Nevertheless, they remain limited by:

- ✓ **Computational complexity:** Kernel methods typically require $O(n^2)$ or $O(n^3)$ operations for matrix construction and inversion, making them computationally prohibitive for large-scale datasets with millions of users and items, especially when multiple kernels are jointly optimized.
- ✓ **Kernel selection sensitivity:** Their performances heavily depends on the appropriate kernel selection and parameter tuning, which requires domain expertise and extensive cross-validation without clear guidelines for optimal choices.
- ✓ **Limited feature learning:** Although kernel methods introduce non-linear mapping, they are often restricted to fixed kernel spaces defined a priori, preventing them from learning adaptive representations directly from data, and constraining their ability to capture intricate and hierarchical interactions in complex biological systems.

- ✓ **Sparse data challenges:** Most kernel-based recommenders struggle when faced with sparse interaction matrices (which is the case in bioinformatics-related recommendation scenarios), which lead to unreliable similarity estimations concerning unseen (cold-start) entities such as novel drugs, or orphan targets.
- ✓ **Limited generalization across heterogeneous domains:** Although multiple kernel learning frameworks aim to integrate heterogeneous data, the learned kernel combinations are often dataset-specific and fail to generalize well to new biological contexts or unseen modalities without re-derivation of kernel function and retraining.
- ✓ **Limited research momentum and stagnation:** Due to their constrained representational capacity and the challenges associated with kernel selection, scalability, and the integration of high-dimensional entity-related data, kernel-based RS have witnessed a steady decline in research attention, as the community increasingly shifts toward DL-based paradigms that offer greater flexibility, adaptability, and generalization potentials.

### 3.6.5. Other methods

Other RS approaches have made modest theoretical and practical contributions to recommendation tasks by strategically leveraging divers algorithmic paradigms (e.g., similarity learning, probabilistic inference, ensemble learning), hybridization techniques, and data modalities to overcome opposed limitations.

In the topic of probabilistic approaches, Gaussian process (GP) based models such as Hofmann's [388] Gaussian generalization to the Probabilistic Latent Semantic Analysis (GpLSA) framework provides the ability to handle continuous-valued user responses, achieved through modeling observed user ratings as if they are arising from a mixture of latent user communities or interest groups, where each user can probabilistically participates in multiple communities, and each community is characterized by a Gaussian distribution over normalized item ratings. Such unique perspective enables GpLSA to outperform several baseline CF techniques in terms of RMSE score values across EachMovie benchmark dataset, establishing GpLSA as a cornerstone in the evolution of probabilistic recommender models, and laying the foundation for subsequent Gaussian and Bayesian-based approaches across domains. Mnih et al. [389] on the other hand introduced Probabilistic Matrix Factorization (PMF), which is capable of formulating recommendation tasks as probabilistic inference problems over latent feature spaces of users and items that are learnable through maximizing a log-posterior function. In this context, PMF assumes that each observed rating arises from the inner product of user-item vector representations that are affected by gaussian noise, with zero-mean spherical gaussian priors over said vectors to regularize the model and prevent overfitting. This design enables PMF to achieve superior predictive accuracy and computational efficiency over several baselines in terms of RMSE score values across Netflix benchmark dataset, establishing it as a robust, scalable, and non-linear recommender architecture. Whereas Kim et al. [390] proposed the Tucker Gaussian Process (TGP) model, a generalization of GP approaches that enables the efficient and expressive modeling of multi-way interactions for regression and CF tasks. Specifically, TGP decomposes the latent function tensor using Tucker decomposition, where individuals are represented by low-rank embeddings, with a core tensor capturing their multiplicative interactions through a GP prior. This formulation allows

TGP to inherit the non-parametric flexibility of GPs, while overcoming their traditional computational bottlenecks by exploiting the multilinear structure of Tucker-decomposition to achieve scalability and compactness, consequently rendering it capable of outperforming standard GP and MF baselines in terms of RMSE score values across California House Price, Irish wind, and MovieLens-100k benchmark datasets, and further demonstrating its ability to generalize across sparse and high-dimensional interaction spaces. Lastly, Yang et al. [391] proposed a Multi-output Gaussian Process (MW-GP) framework that explicitly models the correlations between users and items through a shared covariance structure. Specifically, MW-GP formulates the recommendation task as a multi-output regression problem where the latent functions governing user preferences are drawn from a GP prior with a structured kernel that captures both the similarity between users/items and the dependencies between different rating dimensions or interaction types. This paradigm enables MW-GP to not only generate accurate estimates, but also output well-calibrated predictive variances, ultimately surpassing baselines such as CF-NADE, GPLVM, Sparse FC, GC, CWOCF, LLORMA, Biased MF, SVD++, and Bayesian GPLVM in terms of MAE and RMSE score values across ML-1M (0.643, 0.821), ML-10M (0.579, 0.753), and Jester (2.944, 3.916) benchmark datasets, underscoring its efficiency as a reliable and informative system for decision-making under uncertainty.

Aside from GP approaches, He et al. [392] proposed SimBoost, a hybrid framework that integrates similarity-based features with gradient boosting machines. Specifically, this approach constructs a comprehensive feature space composed of three categories: drug-specific and target-specific features derived from similarity matrices, pairwise network-based features capturing topological relationships among drugs and proteins, and interaction-based features encoding local neighborhood statistics from known drug-target pairs. These features collectively enable SimBoost to account for both molecular similarity and interaction topology, while rendering it capable of outperforming both MF, and KronRLS baselines in terms of RSME, AUC, AUPR, and CI score values across Davis (0.247, 0.956, 0.785, 0.884), KIBA (0.204, 0.907, 0.782, 0.847), and Metz (0.166, 0.958, 0.629, 0.851) benchmark datasets, highlighting its superior ability in capturing complex drug-target interaction patterns. Pal et al. [393] on the other hand proposed KPMF (Kernelized PMF), a hybrid approach that incorporates auxiliary relationship information through kernelized graph priors. Specifically, KPMF constructs separate similarity graphs for users and items, extracts their Laplacians, and then exploits them as kernelized priors on the user-item latent factor matrices in order to enforce proximity among homogeneous entities in the latent space. This design allows KPMF to leverage the structural information encoded within user and item graphs to guide the learning of latent factors, thus renders it capable of surpassing numerous baselines (i.e., FunkSVD, SVD, SVD++, PMF, BPMF, KPMF-Soc, and MKMF) in terms of RMSE score values across CiaoDVD (0.9162), FilmTrust (0.7371), Epinions (1.0899), ML-100k (0.9232), and ML-1m (0.8554) benchmark datasets, establishing KPMF as a pivotal model in terms of bridging probabilistic inference and kernel methods. Finally, while focusing on random forest-based approaches, Chu et al. [394] proposed DTI-CDF, a Cascade Deep Forest framework that constructs concatenated forms of representations for target-drug pairs based on their comprehensive sets of similarity measures. The resulting feature vectors are then processed by a multi-layered cascade forest structure, where each layer consists of multiple random forest classifiers whose predictions are concatenated with the original input features, enabling iterative

feature transformation and representation learning that progressively refines the discriminative information for interaction prediction. This design allows DTA-CDF to outperformance several baselines (i.e., DNN, random forest, XGB, DDR) in terms of AUPR, AUC, F-score values across Enzymes, Ion Channels, GPCR, and Nuclear Receptors benchmark datasets, offering enhanced predictive accuracy and practical applicability in computational drug discovery and repositioning.

In general, such RS provide unique frameworks with flexible integration capabilities that leverage either uncertainty quantification techniques (in the case of probabilistic approaches), or the strengths of multiple algorithmic paradigms (in the case of hybrid approaches) that enable the incorporation of diverse heterogeneous data sources, thereby enabling flexible learning and improved prediction robustness in both commercial and biological recommendation tasks. Nevertheless, they remain limited by:

- ✓ **Elevated computational complexity:** GP-based models exhibit cubic time complexity (due to covariance matrix inversion), which makes them intractable for large-scale datasets, while both hybrid and ensemble models compound this computational burden due to their integration of multiple components.
- ✓ **Limited expressiveness for complex interactions:** Despite their efficient leverage of handcrafted and kernel-derived features, they tend to struggle in capturing high-complex, non-linear interaction patterns that DL architectures excel at modeling, potentially leading to inferior predictive accuracy on tasks with intricate relationship hierarchies.
- ✓ **Data sparsity and cold-start vulnerability:** GP-based and hybrid RS are sensitive to sparse interaction matrices and novel entities (e.g., orphan proteins or novel drugs), as their inference mechanisms rely on existing similarity structures and observed data to define priors, leading to poor generalization in low-data regimes.
- ✓ **Scalability challenges in high dimensions:** Probabilistic, hybrid, and ensemble approaches tend to get increasingly difficult to train and regularize effectively without suitable dimensionality reduction, especially when processing high-dimensional biological feature spaces (e.g., genomic sequences, FP vectors).
- ✓ **Hyperparameter sensitivity:** These models inherently possess numerous hyperparameters (e.g., forest depth, kernel parameters and choice) that result in a high-dimensional optimization spaces, which require extensive tuning and validation to achieve optimal performance across different datasets and tasks.
- ✓ **Model complexity and interpretability trade-off:** Despite providing probabilistic outputs and uncertainty estimates, the underlying latent spaces and kernel projections of probabilistic approaches are often difficult to interpret biologically, thereby hindering their capacity to reveal mechanistic insights about molecular interactions or binding determinants.
- ✓ **Stagnation and innovation plateau:** The methodological advancement in this family has slowed over time, as many approaches kept reusing pre-existing modules with minor architectural variations, showing limited conceptual creativity or scalability improvements, which has led researchers to gradually abandon them in favor of DL frameworks.

## 3.7. Conclusion

RS have proven to be a powerful tool in bioinformatics, particularly in drug discovery, intermolecular interaction prediction, and drug repurposing, addressing many challenges faced by traditional drug development processes through leveraging advanced ML techniques in order to enable more efficient identification of intermolecular interactions, and disease-gene associations. The ability of RS to integrate large biological datasets accelerates the discovery of potential therapies and personalized treatment strategies, offering a significant advantage over conventional methods, as they play a crucial role in drug repurposing, where they enable the repositioning of existing drugs for new therapeutic applications, thus maximizing the utility of already approved compounds. However, limitations such as data sparsity, scalability issues, and the complexity of model interpretability remain challenges that need to be addressed. Despite these hurdles, the integration of RS into bioinformatics continues to drive innovation, offering promising opportunities for more precise, efficient, and individualized medical solutions.

# Chapter 4: Proposed Recommender Systems for Intermolecular Interaction Prediction, Drug Repurposing, and de novo Drug Design

Intermolecular interaction prediction tasks play a pivotal role in modern drug discovery, enabling the identification of drug candidates that can bind to biological targets in order to modulate disease-related pathways. Beyond simply estimating whether a drug interacts with a given target, this task requires uncovering deeper patterns that link the sequential and structural characteristics of molecules with their functional and therapeutic behaviors. Accurate prediction of such interactions helps researchers prioritize promising compounds, uncover new indications, and guide drug repurposing efforts.

This chapter presents the core contribution of our thesis: the development of bioinformatics-oriented recommender system approaches that share the common goal of intermolecular interaction prediction, while supporting both drug repurposing and novel drug design strategies, accomplished through leveraging either sequence or structure-based molecular representations to learn meaningful associations between drugs, targets, and diseases. This chapter is structured as follows: Section 4.1 provides a general introduction that outlines the objectives and scope of the chapter. Section 4.2 defines the problematic and establishes the research context, detailing the problem formulation, the biological relevance of the selected targets, the underlying ML paradigms adopted to approach them, and the criteria used for training, evaluation, and biological validation. Section 4.3 presents the first proposed RS approach and its greedy variant, which are based on convolutional and recurrent neural networks, featuring their architectural design, experimental assessment, and applications in drug repurposing scenarios involving even orphan proteins. Section 4.4 introduces the second RS approach and its translator counterpart, extending conventional RS frameworks by jointly leveraging sequential and structural molecular representations through attention-based and translation mechanisms to support advanced drug discovery and repurposing tasks. Finishing by Section 4.5 where we conclude and summarize our efforts.

## 4.1. Introduction

The growing availability of immunology-related bioinformatics data has played a pivotal role in advancing our understanding of human diseases and their molecular foundations. Critical biological entities such as cytokines, cyclin-dependent kinases, and G-protein-coupled receptors became crucial in elucidating diverse pathological processes, as their intricate interactions with drugs and other biomolecules directly influence the progression and treatment of major disease threats, including cancer, HIV infection, and COVID-19. By studying the molecular communications between these entities (e.g., how cytokines regulate immune pathways, how CDKs drive the cell cycle, and how GPCRs orchestrate cellular signaling); researchers gain essential insights into disease mechanisms and therapeutic vulnerabilities. The sheer volume, diversity, and complexity of accumulated biological data however has created an overwhelming challenge for traditional drug discovery and repurposing pipelines, as classical experimental screening methods and knowledge-driven computational approaches struggle to fully utilize today's massive quantities of molecular sequences, structures, and interaction records. Consequently, identifying novel drug candidates, uncovering previously unknown associations, or predicting interactions for structurally unique or orphan molecules is still an extremely difficult task, since data overflow phenomena has widened the gap between what biological data offers and what conventional methodologies can exploit, thereby slowing the translation of raw information

into meaningful therapeutic insights. To overcome this challenge, researchers resorted to ML paradigms and their ability to enable systems to automatically learn representations, infer patterns, and generalize learned knowledge. Among ML-based strategies, recommender systems have shown particular promise in reformulating intermolecular interaction prediction problems as recommendation tasks by treating proteins, drugs, and other molecules analogously to users and items. By learning from historical interactions and incorporating structural, sequential, or topological information; RS approaches provide effective frameworks for predicting intermolecular interactions, offering scalable, and flexible solutions capable of addressing data sparsity, heterogeneity, and the need for personalized or molecule-specific predictions.

Building on these foundations, this chapter introduces four novel approaches designed to address the limitations of existing interaction prediction models. The first model being RSCRDL, a Recommender System based on Convolutional Recurrent Deep Learning for DTI prediction tasks, which views drugs and proteins as textual entities, enabling the extraction of implicit features directly from their biological sequences without relying on predefined global descriptors, thereby enabling generalization to unseen or orphan molecules. The second approach extends RSCRDL with a hybrid, and greedier mechanism that's capable of integrating explicit molecular Features of individuals (therefore its name RSCRDL_F) to enhance prediction performance when additional structural or chemical information is available. Together, these models aim not only to outperform general-purpose RS baselines but also to broaden the possibilities for interaction prediction in realistic, data-rich, and sparsity-affected biological environments. The third and fourth proposed approaches on the other hand are two complementary transformer-based models that aim to address the dual challenges of predicting unobserved interactions, and generating structurally valid novel drug candidates: a Recommender System that exploits both Sequential and Structural molecular features via its Transformer architecture (RSSSTr), and its translator counterpart (TSSTr). Both models leverage the sequential representations of molecules, alongside their structural fingerprint (FP) information encoded through our novel binary FP indexation strategy, to help provide deeper understanding of biochemical properties far beyond what purely sequence-based representations or global identifiers can capture. More specifically, RSSSTr focuses on matrix completion through estimating unobserved DTIs even for mutated proteins, and both newly developed and unseen compounds. While TSSTr performs matrix expansion via generating novel candidate drug molecules that comply with major drug-likeness constraints. Working cooperatively, the translator expands the chemical search space with biologically plausible compounds, while the recommender validates their therapeutic relevance, forming a scalable framework that bridges predictive interaction modeling with automated drug design. Together, these synergistic models demonstrate a transformative direction for computational drug discovery: from merely retrieving known associations to actively proposing new treatment possibilities guided by interaction-aware learning.

## 4.2. Problem definition and research context

### 4.2.1. Problem formulation

Despite the notable advances in ML-based prediction of molecular properties, the prevailing paradigms for modeling intermolecular interactions face major limitations when translated into

real-world therapeutic discovery. This thesis addresses these limitations by reformulating the interaction prediction challenge as a recommendation task, projecting it as a matrix completion problem where the goal is to infer missing entries in a vast and sparse interaction matrix. While this reframing enables the automated inference of potential associations; existing RS approaches remain hindered by several constraints. First, most RS models still depend heavily on global handcrafted descriptors or index-based encodings, which fail to capture the rich biochemical and structural semantics embedded within molecular sequences and fingerprint representations, therefor limiting their ability to model the true nature of binding mechanisms. Second, the practical characteristics of biological data introduce three major challenges: (i) the extreme sparsity of known interactions, (ii) the emerging cold-start scenarios caused by newly integrated or low-active entities, and (iii) the presence of orphan proteins and novel drug candidates that lack prior interaction histories entirely. Overcoming these obstacles requires moving beyond models that merely interpolate within observed interaction data, towards frameworks that are capable of robustly generalizing to unseen proteins and *de novo* compounds via directly learning from their intrinsic sequential and structural representations. Moreover, the rapid emergence of resistant pathogens, and the finite coverage of existing chemical libraries highlight a critical paradigm shift: computational systems must not only complete interaction matrices, but also expand them. This calls for frameworks that integrate generative molecular design with interaction relevance assessment, allowing the creation of novel therapeutic candidates and the validation of their biological roles within expanded biochemical spaces.

### 4.2.2. The impact of the of the selected biological targets

The selection of specific biological targets is critical for developing robust computational models in bioinformatics, as it determines the relevance and generalizability of the resulting predictions. This study focuses on a curated set of targets where each one represents a distinct and vital aspect of human therapeutic intervention. Their inclusion is driven by a combination of strong therapeutic demand, diverse molecular interaction mechanisms, and comprehensive data availability, which include:

#### 4.2.2.1. Cytokines

A broad category of small secreted proteins produced primarily by immune cells, functioning as essential messengers that coordinate innate and adaptive immune responses. They regulate key biological processes including inflammation, immune cell activation and trafficking, hematopoiesis, and tissue repair. Their dysregulation (whether excessive or insufficient) can drive severe pathological conditions, with the overproduction of pro-inflammatory cytokines causing cytokines storms, which are highly damaging hyper-immune reactions associated with acute respiratory distress, multi-organ failure, and high mortality (as observed in severe cases of viral infections such as COVID-19). Conversely, impaired cytokine signaling can weaken immune surveillance and promote chronic infections or malignant progressions. These dual roles position cytokines as both critical biomarkers and high-valued therapeutic targets in immunological and inflammatory diseases [395].

#### 4.2.2.2. Cyclin-Dependent Kinases (CDKs)

They constitute a conserved family of serine/threonine protein kinases that act as central checkpoints for eukaryotic cell-cycle progression, transcriptional regulation, and DNA damage responses. Their activation requires them to bind to specific cyclin partners in order to regulate transitions between cell-cycle phases. Dysregulation of CDK activities (whether through overexpression, loss of inhibitory control, or oncogenic mutations) results in uncontrolled cell proliferation, impaired cell-cycle arrest, and increased genomic instability. Consequently, CDKs are regarded as major drivers of tumorigenesis across a wide range of cancers, while being treated as high-valued yet complex targets in modern anti-cancer drug development [396].

#### 4.2.2.3. G-Protein-Coupled Receptors (GPCRs)

They represent the largest and most functionally diverse class of membrane-embedded receptors, acting as key molecular transducers that convert extracellular stimuli (e.g., neurotransmitters, hormones, and chemokines) into intracellular signaling responses. They regulate a wide range of essential physiological processes, including vision, immune regulation, cardiovascular function, and metabolic homeostasis. Dysregulation of GPCR signaling, whether through genetic mutations, aberrant expression, or ligand imbalance, is implicated in a broad spectrum of disorders such as hypertension, type II diabetes, neurodegenerative diseases, psychiatric conditions, inflammatory pathologies, and cancer. In addition, their accessibility at the cell surface makes GPCRs prime targets for both therapeutic intervention and pathogen exploitation, as several viruses and bacterial toxins hijack GPCR pathways to facilitate host cell entry or subvert immune signaling. Their pivotal biological role and pharmacological relevance have resulted in GPCRs being the target of more than one-third of FDA-approved drugs, solidifying their importance in modern therapeutic strategies [397].

#### 4.2.2.4. Human Immunodeficiency Virus (HIV)

A retrovirus that selectively infects and destroys $CD4^+$ T lymphocytes (i.e., T helper cells), the principal regulators of the adaptive immune system. Its pathogenicity stems from its ability to integrate into the host genome, promoting persistent infection and immune evasion. The progressive depletion of $CD4^+$ T cells ultimately leads to acquired immunodeficiency syndrome (AIDS), a state of severe immunosuppression characterized by heightened susceptibility to opportunistic infections, malignancies, and life-threatening systemic complications. Despite the success of antiretroviral therapy in suppressing viral replication, HIV remains a major global health burden requiring lifelong treatments, with ongoing challenges including drug resistance and the absence of a definitive cure [398].

#### 4.2.2.5. Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)

The beta-coronavirus responsible for the global COVID-19 pandemic. It primarily infiltrates host cells via ACE2 (Angiotensin-Converting Enzyme 2) receptors, triggering strong inflammatory and immune responses. The clinical risks associated with it range from mild respiratory illness to severe outcomes including acute respiratory distress syndrome (ARDS), multi-organ failure, and death. Beyond acute infection, SARS-CoV-2 can induce a spectrum of

persistent complications known as “Long COVID”, affecting cardiovascular, neurological, and immune systems. Its high transmissibility, and significant mutation rate results in the emergence of diverse immune-evasive variants, continuing to challenge the efficacy of vaccines and therapeutics, while sustaining its profound threat to global health security [399].

Collectively, these targets hold major importance in immune regulation, infectious diseases, and cancer progression, which are three of the most pressing global health concerns. Their heterogeneous biological roles and distinct binding mechanisms create an ideal benchmark for evaluating the generalization capabilities of RS approaches across intermolecular interaction prediction problems.

### 4.2.3. Understanding the selected ML paradigms

The selection of appropriate ML architectures is driven by their intrinsic abilities to capture both the sequential and structural characteristics of biological data, which fundamentally dictate molecular functions, and, consequently, interaction outcomes. Equally important is their capacity to generalize effectively to unseen or orphan molecules. In this work, we focus on two paradigms: the combined use of CNNs and RNNs in the first two approaches, and the adoption of Transformer architectures in the latter two models. These choices are deliberate, reflecting each paradigm’s computational strengths and their suitability for modeling complex protein and drug sequence representations.

#### 4.2.3.1. CNN-RNN synergy

The integration of CNNs and RNNs within a single framework offers significant advantages for intermolecular interaction prediction tasks. CNNs efficiently extract local patterns and structural motifs from molecular sequences, and help identify critical substructures in drugs and conserved domains in proteins that govern binding specificity. Additionally, CNNs reduce the dimensionality and complexity of the input sequences by distilling essential features into more compact representations, thus alleviating the burden on RNNs, which in return focus more on modeling prominent sequential dependencies without necessitating excessively complex architectures. By combining CNNs and RNNs, frameworks deploying them simultaneously capture hierarchical spatial features and sequential contextual relationships, enhancing their generalization to unseen proteins or drugs, improving their robustness against sparse interaction data, and enabling them to predict novel interactions more accurately [400, 401].

#### 4.2.3.2. Transformer architectures

Transformer architectures offer distinct advantages for intermolecular interaction prediction tasks through enabling an efficient parallel-processing paradigm for both sequential and structural representations of molecules (which is our case). Furthermore, unlike sequential models that process data step by step, transformers leverage self-attention mechanisms to capture long-range dependencies and contextual relationships across entire sequences, enabling the detection of subtle interaction-relevant features. Their ability to simultaneously attend to multiple sequence positions and structural elements allows for a comprehensive understanding of protein domains, binding sites, and drug substructures. Moreover, transformers excel at handling large, heterogeneous datasets, and can generalize to unseen or orphan molecules without relying on any historical

interaction records. This capacity makes them particularly well-suited for predicting novel interactions and expanding the chemical-biological interaction space, thereby supporting the design of new therapeutic candidates [368, 372].

By enabling accurate learning from molecular sequences and fingerprints, these DL paradigms support both deductive (repurposing-oriented) and generative (novel-design-oriented) therapeutic objectives, collectively supporting strategies that are faster, more adaptive, and biologically more accurate.

### 4.2.4. Overview of the chosen criteria and hyperparameters

The effectiveness of our proposed approaches depends not only on their architectures and hyperparameters, but also on the criteria used to train, evaluate, and biologically validate them. Given the complexity of biological systems and the high stakes of therapeutic decision-making, it is essential to adopt a well-defined set of hyperparameters and criteria to ensure methodological rigor, predictive reliability, and biological relevance, which in our case include:

#### 4.2.4.1. Training criteria and hyperparameters

Training criteria and hyperparameters are selected to account for the inherent sparsity and imbalance of interaction data, alongside the inherent complexity of the deployed frameworks, while promoting robust feature learning from sequential and structural molecular representations. They ensure that the proposed models learn meaningful latent patterns rather than overfitting to noisy or incomplete interaction records.

**A. Development environment**

The development environment adopted in this work is PyTorch [402], chosen over other higher-level frameworks such as Keras [403] due to its flexibility, transparency, and suitability for research-oriented DL frameworks in bioinformatics. PyTorch's dynamic computational graph enables intuitive model construction, fine-grained control over forward and backward passes, and easier debugging, which is critical when designing custom architectures that integrate sequential and structural molecular representations, such as CNN-RNN hybrids and transformer-based models. This flexibility is particularly advantageous for implementing non-standard training procedures, custom loss functions, and experimental modules that are less straightforward to realize in more abstracted frameworks like Keras. Moreover, PyTorch has become the *de facto* standard in the research community, with strong ecosystem support for state-of-the-art models, pretrained components, and rapid prototyping, as evidenced by its adoption in major libraries and benchmarks in computational biology and chemistry. Its seamless GPU acceleration, native support for automatic differentiation, and close alignment with Python programming practices further enhance productivity and reproducibility, making it a robust and future-proof choice for developing advanced DL models for intermolecular interaction prediction and drug discovery tasks [402].

**B. CNN's kernel size**

Also referred to as the filter size, this parameter defines the spatial dimensionality of the sliding window applied by convolutional layers to an input tensor. By specifying how many adjacent elements are processed jointly at each convolutional operation, the kernel size directly determines the scale of patterns that the network can capture. Smaller kernels emphasize fine-grained local motifs and, when stacked across multiple layers, enable efficient hierarchical feature extraction with fewer parameters. In contrast, larger kernels capture broader contextual or structural patterns in a single operation but incur higher computational costs and may introduce over-smoothing effects. Consequently, the choice of kernel size influences representational capacity, computational efficiency, and the balance between local specificity and global contextual awareness in learned feature representations [222].

### C. Max pooling layers

Mox pooling layers depict downsampling operations used by CNNs to reduce the spatial dimensionality of feature maps while preserving the most salient activations (Figure 4.1). By only retaining the maximum activation within each region, max pooling introduces a degree of translation invariance, noise suppression, complexity reduction, and overfitting mitigation by decreasing the number of parameters in subsequent layers. In sequence-based or 1D convolutional settings, the same principle applies along the sequence dimension, enabling the model to emphasize the most informative local motifs irrespective of their exact positions [227].

### D. Number of GRU units

Also known as the hidden size or hidden dimensionality, this hyperparameter refers to the number of neurons in the hidden-state vector of a GRU layer, which determines the dimensionality of the representation used to encode temporal dependencies in a sequence. Increasing the number of GRU units enhances the model's capacity to capture long-range and complex sequential dependencies, but also raises computational cost and the risk of overfitting [236].

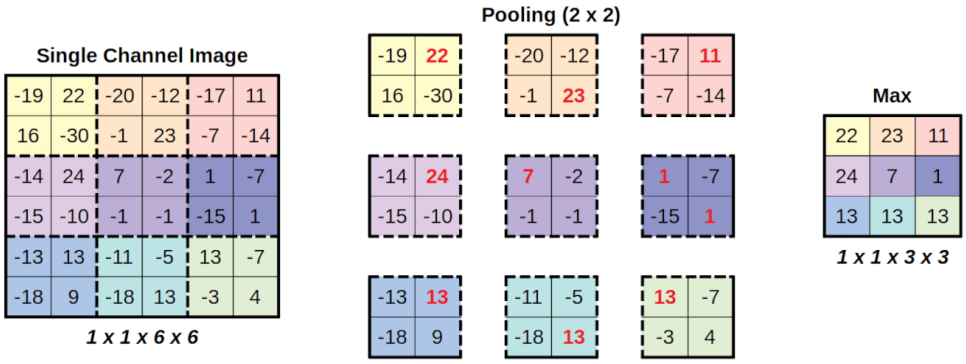



*Figure 4.1 Downsamplng operation of an image channel via the maxpooling layer [404]*

### E. Dropout layers/values

A regularization technique used in neural networks to prevent overfitting by randomly "dropping" a fraction of neurons during training. Dropout effectively reduces co-adaptation between neurons and improves the network's generalization capability [222].

## F. Activation functions

Mathematical functions applied to the outputs of neurons to introduce non-linearity into neural architectures, enabling the approximation of complex, non-linear mappings between inputs and outputs. Without them, all DL-based networks would effectively behave as linear models, regardless of their depth, severely limiting their representational capacity and expressive power [222].

- ✓ **Rectified Linear Unit (ReLU) activation function:** A non-linear activation function widely used in DL architectures due to its simplicity and effectiveness in training deep networks (Figure 4.2). ReLU operates by setting all negative inputs to zero while keeping positive inputs unchanged Eq (4.1), introducing sparsity, while improving computational efficiency, and reducing interdependencies between neurons. Additionally, ReLU mitigates the vanishing gradient problem, allowing for faster convergence during training [405].

$$ReLU(x) = max(0, x) = \begin{cases} x \ \ if \ \ x > 0, \\ 0 \ \ otherwise \end{cases} \qquad \text{Eq (4.1)}$$

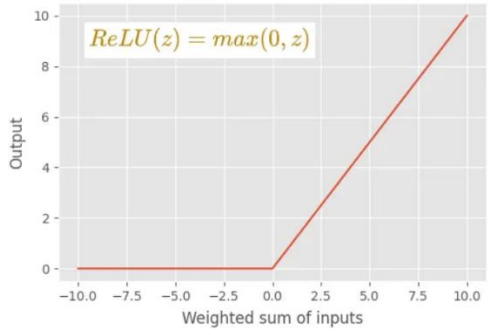



*Figure 4.2 ReLU ctivation function curve [405]*

- ✓ **Sigmoid activation function:** A smooth, non-linear activation function (Figure 4.3) that maps any real-valued input to the bounded interval [0,1] Eq (4.2). Owing to this probabilistic interpretation, the Sigmoid function is commonly employed in neural networks for binary classification, and output layers that require probability estimation. Despite these advantages, its gradient diminishes rapidly for large positive or negative input values, which can induce the vanishing gradient problem in deep architectures, thereby slowing convergence and hindering effective optimization during training [406].

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad \text{Eq (4.2)}$$

## G. Xavier initialization method

Also known as Glorot initialization, this weight initialization method is designed to maintain a stable variance of activations and gradients across layers in DL networks, thereby mitigating vanishing or exploding gradient problems during training. It initializes weights by drawing them from a distribution with zero mean and a variance that depends on the number of input and output

units of a layer Eq (4.3). For a layer with $n_{in}$ input units and $n_{out}$ output units, Xavier initialization is commonly defined as [225]:

$$Var(W)\frac{2}{n_{in}+n_{out}} \qquad \text{Eq (4.3)}$$

Accordingly, weights can be sampled either from a uniform distribution, or a normal distribution. This initialization is particularly well-suited for networks using Sigmoid or hyperbolic tangent activation functions, and has become a foundational technique in DL due to its effectiveness in accelerating convergence and improving training stability, especially in transfer architectures.

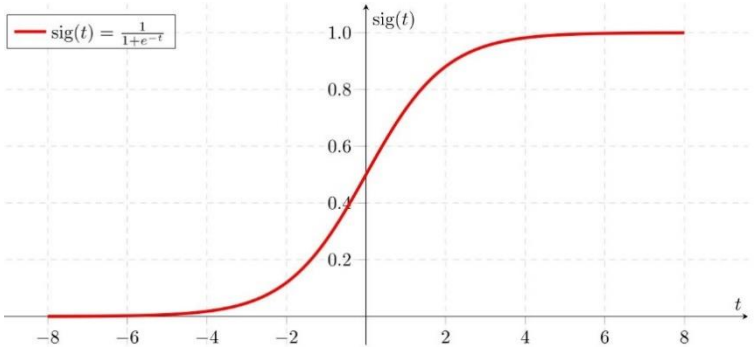



*Figure 4.3 Sigmoid activation funtion curve [406]*

## H. Training, validation, and test splits of data

Similar to most ML models, the process of training and assessing RS involves partitioning a dataset into three disjoint subsets (i.e., training, validation, and test), to ensure both robust learning and unbiased performance assessment. The training set is used to learn model parameters, while the validation set serves as a proxy for unseen data during development, enabling systematic hyperparameter tuning, early stopping, and model selection. In particular, validation performance provides a principled criterion for deciding which version of the model should be retained and later exploited during the inference phase on the test set. Whereas the test set is strictly held out and used only once to estimate the final generalization performance, thereby preventing information leakage and overly optimistic results. This three-way split is especially important for RS tasks, where data sparsity and cold-start challenges can easily lead to underfitting or overfitting if model development is not carefully managed [407].

## I. Number of epochs

The number of epochs is a training hyperparameter that specifies how many complete passes the learning algorithm performs over the entire training dataset. In other words, one epoch corresponds to a full iteration in which each training sample contributes once (typically through mini-batches) to the computation of the loss function and the subsequent parameter updates via a gradient-based optimization algorithm. Increasing the number of epochs generally allows the model to further minimize the empirical risk and capture latent data structures at the risk of overfitting. As a result, the optimal number of epochs is commonly determined using validation-based strategies such as early stopping, which terminates training when performance on a

validation set no longer improves, thereby balancing convergence and generalization performance [141, 222].

### J. Batch size

Batch size specifies the number of training samples processed simultaneously by a model to compute a single gradient update during the training phase. Smaller batch sizes introduce higher stochasticity (i.e., randomness) in gradient estimation, which can act as an implicit regularization technique, potentially improving generalization and facilitating escape from sharp local minima, albeit at the cost of noisier updates and slower convergence. In contrast, larger batch sizes yield more accurate and stable gradient estimates, and enable the efficient utilization of modern parallel hardware, at the cost of increased memory resources, and inferior generalization when excessively large [222].

### K. Padding

A preprocessing ML technique that adds additional artificial elements (most commonly zeros, but also reflected, replicated, or learned values) to the boundaries of input data to control dimensionality and preserve information during model operations. Padding is widely employed in both convolutional and sequence-based models to ensure that operations such as convolutions, pooling, or sliding windows can be applied uniformly across all input positions without unintended loss of boundary information. In sequential and textual data (e.g., natural language tokens, protein sequences, or SMILES strings), padding is primarily used to unify sequence lengths within a batch, thereby enabling efficient parallel computation and stable gradient updates during training [221].

### L. Tokenization process

A preprocessing step in ML, dedicated for sequence-based modeling, in which raw input data (e.g., text, protein sequences, or chemical SMILES strings) is decomposed into smaller, discrete units called tokens, which can efficiently be mapped to distinct numerical representations. Depending on the application, tokens may correspond to characters, sub-words, words, or biologically/chemically meaningful motifs. Tokenization allows models to process standardized, numerically encodable elements, facilitating embeddings, feature extraction, and the learning of meaningful patterns within the sequences [225].

### M. Loss functions

In ML, a loss function (also called a cost function or objective function) is a mathematical formula that quantifies the discrepancy between a model's predicted outputs and the corresponding true target values for a given dataset. Formally, it assigns a non-negative scalar value to each prediction, with larger values indicating greater errors. Loss functions are essential for guiding the optimization process during training, as their gradients with respect to model parameters are used by gradient-based algorithms to iteratively update and improve the model. The choice of loss function directly impacts model performance, convergence behavior, and suitability for a specific task [141, 222].

- ✓ **Binary cross entropy (BCE):** Also known as log loss, this loss function is widely used for binary classification tasks, where the target variable belongs to one of two mutually exclusive classes. BCE quantifies the dissimilarity between true labels $y$ and predicted probabilities $\hat{y}$ outputted by the model via applying Eq (4.4).

$$\mathcal{L}_{BCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad \text{Eq (4.4)}$$

  BCE is particularly well-suited for probabilistic models, as it penalizes confident but incorrect predictions more heavily, and encourages predicted probabilities to align closely with the true class distributions. In the context of DTI prediction using recommender systems, BCE enables models to differentiate between positive and negative interactions more efficiently, even within sparse interaction matrices. By directly optimizing for the probability of an interaction occurring, BCE supports accurate prediction of both known and novel interactions, which is critical for addressing cold-start problems, in a generalized, inductive framework [222].
- ✓ **Cross entropy loss function (CE):** Also known as the Softmax loss, this cost function is widely used for multi-class or probabilistic prediction tasks, where the objective is to estimate the likelihood of each class or token in a discrete distribution. Formally, for a target one-hot encoded vector $y \in \{0,1\}^C$ and predicted probability vector $\hat{y} \in [0,1]^C$ over $C$ classes, the loss for a dataset of $N$ instances is defined as:

$$\mathcal{L}_{CE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}[y_{i,j} \log(\hat{y}_{i,j})] \quad \text{Eq (4.5)}$$

  In the context of sequence-generating models, CE loss is applied token-wise across generated sequences, encouraging the model to assign higher probabilities to the correct next token (e.g., AA or SMILES characters). By optimizing CE loss, the model learns to produce novel drug candidates that preserve the statistical and structural properties observed in the training data, supporting downstream tasks such as matrix expansion and interaction validation [222, 239].

## N. Label smoothing

A regularization technique used in categorical or sequence-based prediction tasks to prevent a model from becoming overconfident in its predictions. Instead of assigning a probability of 1 to the correct class and 0 to all others, label smoothing redistributes a small fraction of the probability mass to the incorrect classes. The resulting smoothed target vector is then used in the CE loss function instead of the original one-hot labels. This technique has been shown to improve generalization, reduce overfitting, mitigate model overconfidence, and stabilize the training of deep networks, particularly in sequence generation tasks such as transformer-based molecule generation [239].

## O. Adam optimizer (Adaptive Moment Estimation)

A gradient-based optimization algorithm that's widely used for training neural networks, as it combines the advantages of two classical methods: AdaGrad (i.e., Adaptive Gradient Algorithm),

which adapts the learning rate for each parameter individually based on historical gradients, and RSMProp (Root Mean Square Propagation), which normalizes updates by the moving average of recent squared gradients. Adam maintains exponentially decaying averages of past gradients (first moment $m_t$) and past squared gradients (second moment $v_t$), and updates model parameters using bias-corrected estimates of these moments. Formally, for a parameter $\theta_t$ at iteration $t$, learning rate $\alpha$, decay rates $\beta_1$ and $\beta_2$, and gradient $\frac{\partial \mathcal{L}}{\partial \theta_t}$, the updates are:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\frac{\partial \mathcal{L}}{\partial \theta_t} \quad \text{Eq (4.6)}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\left(\frac{\partial \mathcal{L}}{\partial \theta_t}\right)^2 \quad \text{Eq (4.7)}$$

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t}, \qquad \widehat{v}_t = \frac{v_t}{1-\beta_2^t} \quad \text{Eq (4.8)}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \varepsilon} \quad \text{Eq (4.9)}$$

Adam is popular due to its efficient handling of sparse gradients, fast convergence, and reduced sensitivity to hyperparameter tuning [408].

#### 4.2.4.2. Evaluation criteria

The selected evaluation metrics are specifically selected to assess the classification reliability and robustness of trained models across various recommendation scenarios (e.g., data sparsity and colds-start).

**A. Confusion matrix**

A confusion matrix is a tabular representation used in ML to summarize the performance of a classification model by comparing its predicted labels against true labels. It provides a detailed breakdown of correct and incorrect predictions across all classes, enabling the computation of various evaluation metrics such as accuracy, precision, recall, specificity, sensitivity, F1-score, and balanced accuracy. In the binary classification case, the confusion matrix consists of four components:

- ✓ **True Positives ($T_P$):** correctly predicted positive instances
- ✓ **True Negatives ($T_N$):** correctly predicted negative instances
- ✓ **False Positives ($F_P$):** negative instances incorrectly predicted as positive ones
- ✓ **False Negatives ($F_N$):** positive instances incorrectly predicted as negative ones

Confusion matrices are valuable in biomedical and bioinformatics applications, where misclassification errors may have severe practical or biological consequences [409].

**B. Precision score**

A classification metric that measures the proportion of predicted positive instances that are truly positive. It quantifies a model's ability to avoid false positive predictions, and is formally defined as:

$$Precision = \frac{T_P}{T_P + F_P} \qquad \text{Eq (4.10)}$$

Precision is particularly critical in applications where false positives carry a high cost, such as DTI prediction tasks, where incorrectly predicting an interaction may lead to wasted experimental validation or misleading biological conclusions [409, 410].

### C. Recall score

Also known as sensitivity or true positive rate, this metric quantifies a model's ability to correctly identify all relevant positive instances. It is defined as the proportion of true positives among all actual positive samples, measuring how effectively a classifier captures positive cases disregarding false ones. Formally, recall is given by

$$Recall = \frac{T_P}{T_P + F_N} \qquad \text{Eq (4.11)}$$

Recall is particularly important in applications where missing positive instances is costly, such as medical diagnosis, and interaction prediction tasks, as it emphasizes completeness over prediction's precision [409, 410].

### D. F1-score

A harmonic mean-based metric that combines precision and recall into a single scalar value, providing a balanced measure of a model's accuracy in binary or multi-class classification tasks. It is particularly useful when dealing with imbalanced datasets, as it penalizes extreme trade-offs between precision and recall. F1-score is defined as:

$$F1\text{-}score = 2 \cdot \frac{Precision * Recall}{Precision + Recall} \qquad \text{Eq (4.12)}$$

By using the harmonic mean rather than the arithmetic mean, the F1-score ensures that high values are only achievable when both precision and recall are simultaneously high. Consequently, it is widely adopted in RSs, bioinformatics, and information retrieval tasks where class imbalance and false negatives are critical concerns [409, 411].

### E. Balanced accuracy score

This metric measures a model's ability to correctly predict each individual class with equal importance. It is defined as the arithmetic mean of recall (true positive rate) computed independently for each class, thereby compensating for skewed class distributions that can bias standard accuracy. In a binary classification scenario, balanced accuracy is given by:

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{T_P}{T_P + F_N} + \frac{T_N}{T_N + F_P}\right) \qquad \text{Eq (4.13)}$$

Balanced accuracy provides a more reliable assessment of predictive performance in scenarios such as intermolecular interaction prediction, where positive and negative interactions are often highly imbalanced [412].

### F. Cohen's kappa score ($\kappa$)

A statistical measure used in ML to quantify the level of agreement between a model's predicted labels and the true labels while explicitly accounting for agreement that may occur by chance. Unlike simple accuracy, $\kappa$ provides a more robust evaluation in the presence of class imbalance, making it particularly suitable for binary classification problems with skewed label distributions. For a binary classification task, Cohen's kappa formula can be written as:

$$\kappa = \frac{2(T_P \cdot T_N - F_N \cdot F_P)}{(T_P + F_P)(F_P + T_N) + (T_P + F_N)(F_N + T_N)} \qquad \text{Eq (4.14)}$$

The value of $\kappa$ ranges from $-1$ to $1$, where $1$ indicates perfect agreement, $0$ corresponds to agreement equivalent to random chance, and negative values indicate agreements that are worse than chance [413].

### G. Inter-cluster criteria

In clustering and unsupervised learning, this metric quantifies the separation or dissimilarity between distinct clusters within a dataset. It evaluates how far apart the centroids of different clusters are, providing a measure of cluster distinctiveness. A higher inter-cluster distance indicates better separation between clusters, which generally corresponds to improved clustering quality [414]. Formally, for clusters $j$, $k$ with the centroids $z_j$, $z_k$, the inter-cluster distance can be computed as:

$$Inter\text{-}cluster = \sum_{\forall\, k,j=1..C\,,k\neq j} \| z_k - z_j \|^2 \qquad \text{Eq (4.15)}$$

### H. Intra cluster criteria

In contrast to the aforementioned metric, intra-cluster quantifies the compactness or cohesion of data points within a single cluster. It evaluates how closely the members of a cluster are grouped around its centroid, providing a measure of cluster tightness. Lower intra-cluster values indicate higher cohesions, which is generally desirable for high-quality clustering [414]. Formally, for data points $o_i$ of cluster $k$ that has the centroid $z_k$, the intra-cluster distance can be computed as:

$$intra\text{-}cluster = \sum_{k=1}^{C} \sum_{i=1}^{N} \| o_i - z_k \|^2 \qquad \text{Eq (4.16)}$$

### I. Correlation distance metric

This distance metric measures the dissimilarity between two vectors based on their correlation rather than their absolute values. Unlike Euclidean distance, which is sensitive to magnitude, correlation distance captures the degree to which two vectors vary together, making it particularly useful for comparing patterns or trends in data [415]. Formally, for two vectors $x$ and $y$, with the mean values $\bar{x}$, and $\bar{y}$, the correlation distance $d_{corr}$ is defined as:

$$d_{corr}(x,y) = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad \text{Eq (4.17)}$$

## J. Chebyshev distance metric

Also known as the maximum metric, or $L_\infty$ norm, this metric measures the dissimilarity between two vectors in a multidimensional space, while considering only the largest absolute difference across all dimensions. Formally, for two vectors $x, y \in \mathbb{R}^n$, the Chebyshev distance is expressed as

$$d_{Chebyshev}(x,y) = \max_{i=1,\cdots,n} |x_i - y_i| \quad \text{Eq (4.18)}$$

This metric is particularly useful in applications where the most significant deviation along any single dimension dominates the notion of distance (e.g., chess board movement modeling, anomaly detection). Its emphasis on the largest component difference makes it robust to cumulative small variations while highlighting critical outliers [416].

## K. METEOR score (Metric for Evaluation of Translation with Explicit ORdering)

An automatic evaluation metric originally developed for machine translation that can also be applied in molecular sequence generation tasks. Unlike simple exact-match metrics, METEOR score evaluates the quality of a generated sequence by aligning it to a reference sequence based on exact matches, stemmed matches, synonym matches, and paraphrase matches, while incorporating both unigram precision and recall into its computations. It further penalizes fragmented matches by considering the number of chunks in which matched tokens appear, rewarding sequences that maintain correct word order or structural order. Formally, the METEOR score is computed as:

$$METEOR = F_{mean}(1 - p_{frag}) \quad \text{Eq (4.19)}$$

$$P = \frac{m}{w_t} \quad \text{Eq (4.20)}$$

$$R = \frac{m}{w_r} \quad \text{Eq (4.21)}$$

$$F_{mean} = \frac{10PR}{R + 9P} \quad \text{Eq (4.22)}$$

$$p_{frag} = 0.5\left(\frac{c}{u_m}\right)^3 \quad \text{Eq (4.23)}$$

Where $P$ is the unigram precision, $R$ is the unigram recall, $m$ is the number of unigrams present in both candidate translation and reference translation, $w_t$ and $w_r$ are the number of unigrams in the candidate and reference translations respectively, $F_{mean}$ is the harmonic mean, $p$ is the penalty, $c$ is the number of chunks, and $u_m$ is the number of unigrams that have been mapped [417].

#### 4.2.4.3. Biological criteria

These criteria incorporate domain-specific tools and constraints that help evaluate whether a model's outputs are consistent with known molecular behavior, therapeutic feasibility, and biological mechanisms. Such validation is particularly critical when extending predictions to orphan proteins, novel compounds, or previously unobserved interaction spaces.

**A. Molecular fingerprints (FP)**

Molecular fingerprints are fixed-length vector representations (binary or numerical) that encode the presence, absence, or frequency of chemical features within a molecule, enabling efficient similarity computation, clustering, and ML-based analysis. They transform a molecular graph (defined by atoms as nodes and bonds as edges) into a numerical representation that captures substructures, connectivity patterns, and, in some cases, physicochemical properties. Molecular fingerprints are broadly categorized into two main types: (i) Dictionary-based FPs, which rely on predefined vocabularies of chemical substructures or functional groups (e.g., MACCS keys), offering high interpretability and reproducibility, despite being constrained by the coverage and expressiveness of their fixed dictionaries. (ii) Hash-based FPs, which algorithmically generate molecular substructures and map them into fixed-length vectors using hash functions that permit collisions (e.g., Extended-Connectivity FP), providing greater flexibility and scalability in capturing diverse structural patterns, albeit at the expense of reduced interpretability due to hash collisions and the absence of explicit feature semantics [418].

- ✓ **Extended Connectivity Fingerprint with a bond-diameter of 6 ($ECFP_6$):** ECFPs are a class of circular, hash-based molecular descriptors generated using the Morgan algorithm, where local atomic environments are iteratively expanded to capture neighborhood information around each atom. $ECFP_6$ specifically refers to an ECFP with a bond diameter of 6, corresponding to a radius of 3 bonds, meaning that for each atom, its chemical environment is characterized by considering all neighboring atoms and bonds up to three bonds away. The algorithm begins by assigning initial identifiers to atoms based on properties such as atomic number, valence, and connectivity, then iteratively updates these identifiers by aggregating information from neighboring atoms at increasing radii. ECFP6 is widely used in cheminformatics and drug discovery because it provides a strong balance between local structural sensitivity and computational efficiency, and has been shown to perform particularly well in tasks such as molecular similarity search, activity prediction, and DTI modeling [419].

**B. Biological similarity measuring metrics**

These metrics are used to quantitatively assess the degree of resemblance between biological entities based on their sequential, structural, or functional characteristics. As such, they play a

critical role in bioinformatics and computational biology, particularly for validating and interpreting the outcomes of ML-based models in tasks such as interaction prediction, functional annotation, and candidate prioritization [420].

- ✓ **Tanimoto coefficient (Tc):** Also known as the Jaccard similarity index, this similarity metric is widely used in cheminformatics and computational biology for quantifying the resemblance between the feature vectors (typically FP vectors) of biological entities. Given two fingerprint vectors $A$ and $B$, Tc is defined as:

$$Tc(A,B) = \frac{c}{a+b-c} \qquad \text{Eq (4.24)}$$

  Where $c$ is the number of bits set to 1 in both vectors, and $a$ and $b$ are the number of bits set to 1 in each vector respectively. This metric ranges from 0 (no shared features) to 1 (identical representations). Due to its robustness, interpretability, and empirical success, the Tc is a standard metric for validating learned representations and comparing predicted drug candidates against known compounds in ML-driven DTI studies [421].

- ✓ **E-value score:** In the context of protein-protein similarity analysis, the E-value provides a BLAST-specific, biologically grounded criterion to evaluate the significance of sequence alignments, which is particularly useful for validating predicted interactions and integrating ML-based similarity measures with established bioinformatics methods. Specifically, it represents the expected number of alignments with a score equal to or greater than the observed alignment score that would occur purely by chance when searching a database of a given size, with lower E-values corresponding to more statistically significant alignments. Formally, it is defined as:

$$E\text{-}value = K \cdot m \cdot n \cdot e^{-\lambda S} \qquad \text{Eq (4.25)}$$

  Where $S$ is the raw alignment score, $m$ and $n$ are the effective lengths of the query and database sequences, and $\lambda$ and $K$ are statistical parameters determined by the scoring system used in BLAST [422].

## C. Biological computational tools

These tools are software frameworks or algorithmic platforms designed to facilitate the analysis, modeling, and interpretation of complex biological data. They leverage computational methods to process diverse types of biological information, including nucleotide or protein sequences, molecular structures, and chemical properties, enabling tasks such as similarity assessment, functional annotation, interaction prediction, and molecular property estimation. By formalizing biological problems into computationally tractable models, these tools provide reproducible, scalable, and systematic approaches for exploring large-scale datasets, supporting hypothesis generation, and guiding experimental validation in fields such as genomics, proteomics, cheminformatics, and systems biology [423].

- ✓ **BLAST (Basic Local Alignment Search Tool):** BLAST is a widely used bioinformatics algorithm for comparing biological sequences (e.g., DNA, RNA, or proteins) against a sequence database to identify regions of local similarity. By detecting homologous sequences; BLAST facilitates the inference of potential functional, structural, or evolutionary relationships, as it balances computational efficiency and sensitivity through heuristic search strategies while providing statistical measures, such as E-value score, to assess the significance of alignments. BLAST has become a cornerstone of sequence-based analyses, including protein function annotation, similarity-based clustering, and the validation of predicted molecular interactions in computational biology and drug discovery studies [10, 424].
- ✓ **RDKit:** An open-source cheminformatics toolkit designed to facilitate the representation, manipulation, and analysis of chemical structures and molecular data. It provides a wide range of functionalities, including molecular fingerprint generation, substructure searching, descriptor calculation, molecular visualization, and the handling of SMILES and other chemical formats. RDKit enables the integration of chemical information into computational pipelines, supporting tasks such as drug discovery, virtual screening, and ML-based molecular property prediction. Its versatility and efficiency make it a widely adopted platform for both academic research and industrial applications in cheminformatics [425].

Together, the training, evaluation, and biological criteria form a unified framework for ensuring that the proposed models are not only computationally effective, but also scientifically valid and biologically meaningful. By aligning optimization objectives with evaluation rigor and biological interpretability, this criteria-driven approach supports reliable model comparison, informed conclusions, and translational relevance in the context of computational drug discovery and repurposing.

## 4.3. First proposition: Convolutional and recurrent deep learning-based approaches for DTI prediction as matrix completion task

### 4.3.1. Recommender System based on Convolution Recurrent Deep Learning (RSCRDL)

In this section, we introduce our first pair of proposed approaches (i.e., RSCRDL and its greedy, feature-augmented variant RSCRDL_F) which view intermolecular interaction prediction tasks through an NLP perspective. As depicted in Figure 4.4, RSCRDL is constructed around the premise that biological sequences are much like textual data, encoding rich implicit information that governs molecular behaviors. Motivated by the limitations of traditional RSs that rely primarily on global or handcrafted features, RSCRDL instead processes the textual representations of proteins and drugs to directly extract latent, interaction-relevant biochemical features. To this end, the model incorporates parallel convolutional and recurrent layers, enabling the simultaneous capture of local structural motifs and broader sequential dependencies; an approach analogous to deep NLP architectures used in sentence modeling. This design allows the framework to learn an implicit interaction function that generalizes robustly to unseen or non-reactive molecules, effectively addressing cold-start scenarios by leveraging the intrinsic information embedded within AA and SMILES representations. Furthermore, our models demonstrate that sequence-based learning can rival (and in many cases surpass) general-purpose, state-of-the-art RS

approaches, underscoring the inadequacy of relying solely on global descriptors when richer biological information is available. Building upon RSCRDL's foundation, the RSCRDL_F variant incorporates additional CB aspects through a greedy feature-integration mechanism dedicated to enhance the overall predictive confidence. Aside from performing typical matrix completion tasks, we also explore the applicability of both our approaches on orphan protein sequences, evaluating their ability to infer drug candidates for targets with no recorded interactions, highlighting their potential in data-sparse biomedical settings.

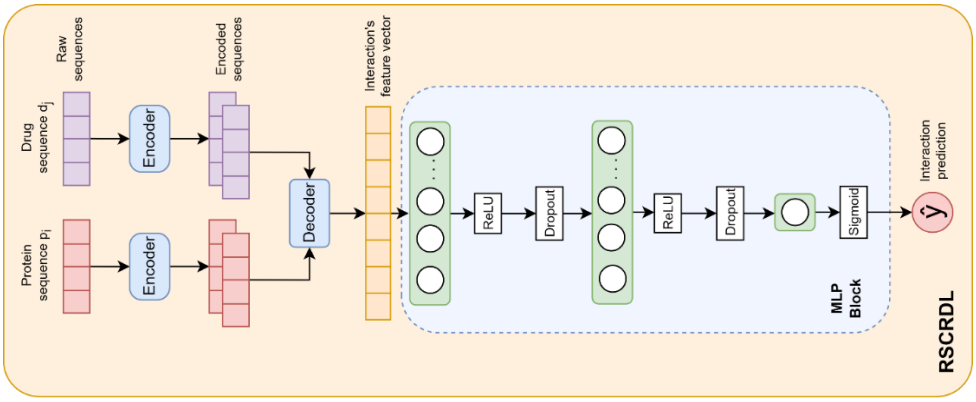



*Figure 4.4 The architecture of RSCRDL*

Conventional RS models (e.g., NeuMF, Glocal-K, and CGI [362, 387, 426]) rely on global features (e.g., indices within lookup tables) assigned to users and items based on the information extracted from the interaction matrix $R$, that encompasses their behaviors. Although their techniques may grant them the ability to learn local relations existing among residents of $R$ without any external knowledge; they may affect their generalization capabilities negatively, especially for low-profiled individuals, and new ones. In contrast, RSCRDL adopts a problem-specific approach that alleviates these shortcomings by extracting the latent features of users and items directly from their textual representations. Instead of depending on arbitrary global latent factors; RSCRDL employs encoder-decoder architectures to process tokenized protein sequences and drug SMILES strings, enabling it to derive biologically and chemically meaningful features that originate from domain-relevant sources in a manner similar to that of the CB models. More specifically, for proteins, the model learns the latent feature vectors corresponding to the AA tokens composing each sequence. And with all proteins being composed of the same set of AA residuals; any update to regarding an AA's embedded representation affects all protein sequences containing that same AA. This merit grants RSCRDL broader insight across all protein sequences without necessitating any for form of pretraining, enabling the robust processing of unobserved molecular entities, while eliminating the cold-start problem (the same process applied on drug SMILES). After the encoding process, the resulting molecular feature vectors are concatenated to form a joint interaction representation. This combined vector is passed to the model's decoder, which computes a refined interaction-specific representation, before processing it via an MLP block to determine the interaction's class. Figure 4.4 depicts the overall architecture of RSCRDL, and its three phases to capture the latent features of both proteins and drugs.

### A. Encoding phase

During the forward propagation of an interaction instance, the textual representations of the two involved molecules are individually passed to their corresponding encoder modules, denoted as $Enc()$. Within each encoder, the input sequence is processed at a token level through five parallel branches designed to extract diverse implicit features. The outputs of these branches are subsequently aggregated via summation to produce a single feature vector encapsulating all extracted information for that specific sequence, as illustrated in Figure 4.5. Specifically, each sequence undergoes preprocessing steps that include tokenization and padding, then projected into a latent embedding space through an embedding layer. Let $U_P$ represent the protein sequence entering the protein encoder $Enc_P(\ \ )$, and let $a_i \in R^{h_emb}$ denote the $h$-dimensional vector representation of the $i^{th}$ AA generated by the embedding function $Emb()$. The transformed representation of the protein sequence $U_{P_emb}$, after projecting all its constituent AA tokens, is given by:

$$U_{P_emb} = a_1 \oplus a_2 \oplus \cdots \oplus a_n \quad \text{Where } a_i = Emb(AA_i) \qquad \text{Eq (4.26)}$$

Here, $U_{P_emb} \in \mathbb{R}^{n \times h_emb}$ is the embedded representation of the protein, $\oplus$ denotes the concatenation operation, $n$ is the padded length of the protein sequence (note that padding tokens do not contribute to backpropagation), $Emb()$ is the embedding layer, and $AA_i$ represents the $i^{th}$ AA in the input protein sequence. At this stage, each protein sequence can be conceptualized as a one-dimensional image of length $n$ with $h_{emb}$ channels, enabling the application of 1D convolutional operations for structural feature extraction. Following extensive empirical experimentation, we settled at four parallel convolutional layers with distinct kernel sizes, as specified below:

$$U^i_{P_conv} = pool\left(f\left(conv\left(U_{P_emb}, k_i\right)\right)\right) \quad for\ k_i \in K \text{ and } K = \{2,3,5,7\} \qquad \text{Eq (4.27)}$$

In this formulation, $U^i_{P_conv} \in \mathbb{R}^{n_conv_i \times h_conv}$ is the resulting feature vector, with $n_conv_i$ representing the transformed sequence's length, $h_conv$ indicating the number of output channels, $pool$ denoting the $1D$ max-pooling operation, $f$ representing the $ReLU$ activation function, and $conv$ signifying the $1D$ convolution operation. After the simultaneous processing through all four convolutional layers, the sequence is further processed through GRU layers. Additionally, a parallel GRU layer processes the raw embedded sequence representation $U_{P_emb}$ directly to prevent information loss through maximizing the extraction of sequential features. Consequently, $U_{P_emb}$ and all four $U^i_{P_conv}$ feature vectors are processed independently according to:

$$U^j_{P_GRU} = GRU(U^k_P) \qquad \text{Eq (4.28)}$$

Where $U^k_P \in \{U_{P_emb}, U^i_{P_conv}\}$, and $U^j_{P_GRU} \in \mathbb{R}^{n_GRU \times h_GRU}$ is the corresponding output feature vector, with $n_GRU$ denoting the number of GRU units and $h_GRU$ representing the dimensionality of the latent space. Finally, all five feature vectors generated by the GRU layers are summed—a design choice validated through empirical testing over concatenation—and passed

to the encoder's final GRU layer to further refine the implicit feature extraction. This yields the sequence's ultimate vector representation $U_{P_i} \in \mathbb{R}^{n_GRU \times h_GRU}$. An identical procedure is followed for the drug sequence $U_{D_j}$ using its corresponding drug encoder $Enc_D(\quad)$.

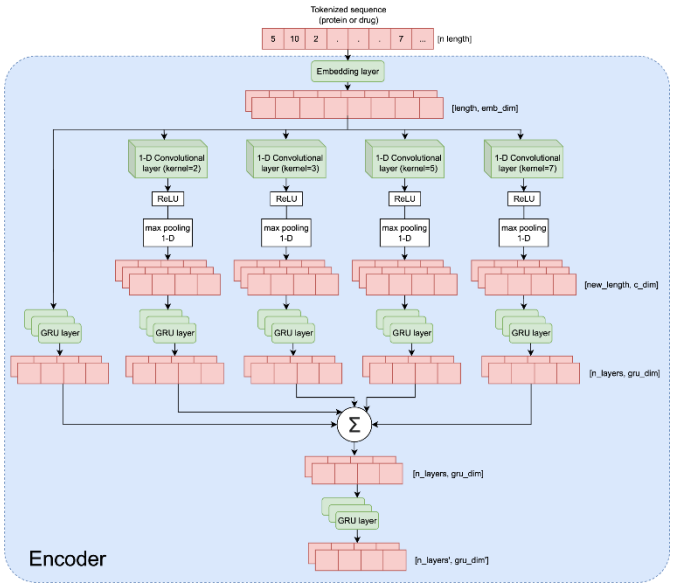



*Figure 4.5 RSCRDL encoder's architecture*

## B. Decoding phase

Once the encoding phase has independently computed the feature vectors for both molecules involved in a given interaction, the decoding process is initiated. This phase starts with the concatenation of the two feature vectors, that pass through the decoder module $Dec()$ to generate the interaction's final feature vector, as illustrated in Figure 4.6. During decoding, the concatenated vector passes through a GRU layer, which is subsequently followed by a Dropout layer. The resulting representation is then flattened according to the following equations:

$$U_{interaction_{(i,j)}} = Dec(U_{Pi} \oplus U_{Dj}) \qquad \text{Eq (4.29)}$$

$$Dec(U_{Pi} \oplus U_{Dj}) = Flatten(Dropout(GRU(U_{Pi} \oplus U_{Dj}))) \qquad \text{Eq (4.30)}$$

In these formulations, $(U_{Pi} \oplus U_{Dj}) \in \mathbb{R}^{n_GRU \times 2 * h_GRU}$ represents the concatenated feature vectors of the protein and drug generated by their respective encoders, while $U_{interaction_{(i,j)}} \in \mathbb{R}^{n_GRU' \times h_GRU'}$ denotes the flattened vector representation of the interaction between protein $Pi$ and drug $Dj$.

## C. MLP block

Following the decoding phase, the resulting interaction vector is passed through two fully connected layers, each employing a $ReLU$ activation function and followed by a Dropout layer for regularization. Finally, the processed vector is fed into an output layer with a $Sigmoid$ activation function, which produces the predicted interaction class.

### 4.3.2. The RSCRDL_F variant

While developing RSCRDL, our primary objective was integrating complementary RS techniques to create a more robust predictive approach. Recognizing that drugs possess a variety of accessible explicit features; we further extended our approach to construct RSCRDL_F. This greedier variant enables the incorporation of 18 additional molecular drug-related features by concatenating them directly with the interaction's flattened feature vector prior to the MLP block. By doing so, RSCRDL_F leverages the advantageous aspects of CB recommendation approaches, thereby enhancing its capacity of understanding and predicting intermolecular behaviors.

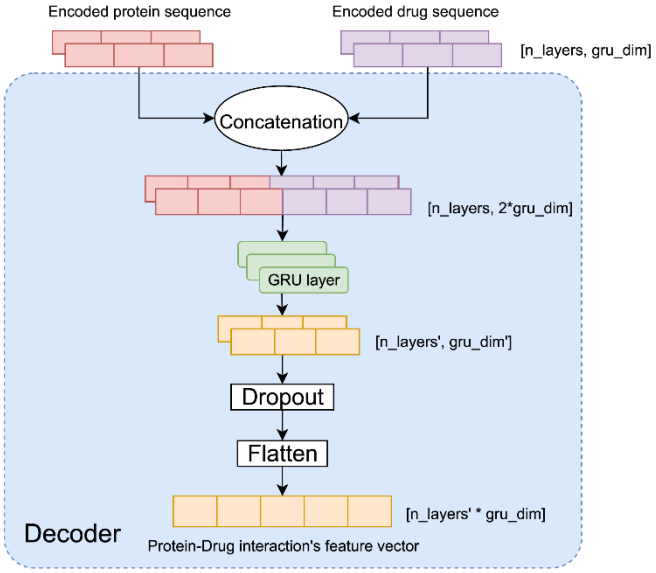



*Figure 4.6 RSCRDL decoder's architecture*

### 4.3.3. Evaluation and analysis

#### 4.3.3.1. Datasets

To assess the performance of our proposed approaches, we conducted experiments on two distinct datasets derived from the ChEMBL database. Both datasets capture interactions between protein sequences and drug compounds, yet each is constructed according to a different biological criterion, as summarized in Table 4.1. The first dataset focuses on protein sequences belonging to the cytokine family and their interactions with drug compounds, with interaction labeling governed by the $IC50$ criterion. Specifically, interactions exhibiting $IC50$ values that are less than or equal to $100nM$ were classified as positive (i.e., biologically relevant binding), while all remaining interactions were designated as negative. The second dataset centers around protein sequences from the CDK family and their corresponding drug interactions. In this case, interaction classification was determined using the percentage inhibition criterion. Interactions demonstrating 50% inhibition or greater were labeled as positive, whereas the remainder were classified as negative.

#### 4.3.3.2. Experimental setup

All layers of both models share identical sets of hyperparameters. The encoder's embedding layers project the initial drug and target sequences into a 60-dimensional latent space, followed by

four parallel $1D$ convolutional layers that preserve the same dimensionality of $60$, with each layer employing a distinct filter size as specified in Eq. (4.27). The GRU layers each contain $8$ units, with a hidden size of $30$ and a dropout value of $0.5$—a configuration determined empirically. Finally, the MLP block comprises two hidden layers with $256$ units each, both utilizing dropout layers set to $0.5$.

*Table 4.1 Characteristics of the deployed datasets*

| Dataset Properties | Cytokine | CDK |
|---|---|---|
| Number of interactions (samples) | 5846 | 4079 |
| Number of proteins | 60 | 35 |
| Number of drugs | 5054 | 1476 |
| Number of positive samples | 2903 | 1866 |
| Number of negative samples | 2943 | 2213 |
| Interaction matrix sparsity | 98.072 % | 92.104 % |

#### 4.3.3.3. Training process

We implemented and executed RSCRDL and its variant RSCRDL_F using PyTorch 1.10.1. Model performance was monitored and optimized via the BCE loss function and the Adam optimizer, respectively. The learning rate was initialized at $10^{-3}$ and decayed progressively over the training course. Training was conducted for $50$ epochs on the cytokine dataset and $250$ epochs on the CDK dataset; these divergent values were intentionally selected to evaluate all tested approaches under both short and long training durations. The batch size was uniformly set to $32$ for both datasets. The best-performing version of each model—subsequently evaluated on the test set—was selected based on Cohen's kappa score computed on the respective validation sets. For the training process of RSCRDL_F, we transferred the weights of both encoders and the decoder from the best-performing version of RSCRDL to provide initial knowledge regarding both drugs and targets, thereby enabling an assessment of transfer learning effects on our architecture. Both models were evaluated via a binary classification task on the two aforementioned datasets, partitioned into training (75%), test (20%), and validation (5%) splits. The results of our models were compared against two general-purpose state-of-the-art RS: NeuMF and GLocal-K [362, 387]. Both baseline approaches were deployed with the parameter configurations specified by their original authors.

#### 4.3.3.4. Performance assessment

Starting with the BCE loss values for all training and validation sets, depicted in Figure 4.7 (GLocal-K is excluded from this analysis due to its fundamentally different learning methodology), we observe that RSCRDL performs relatively poorly on the cytokine training and validation sets, exhibiting what appears to be underfitting. We hypothesize that this underfitting stems from one or more of the following factors: (1) the model's high complexity may have created a larger number of local minima, rendering the weight update process more difficult—an issue that could be mitigated by increasing the total number of training epochs; (2) the intentionally high dropout values set to prevent overfitting—a configuration determined through multiple trial-and-error rounds—may be overly aggressive; or (3) and most plausibly, the sparsity observed in the cytokine dataset, where the majority of drugs (3,639 out of 4,011 drugs in the training set) possess

only a single interaction, resulting in feature vectors representing unique drug tokens being updated infrequently—also addressable by extended training. These hypotheses are further supported by the results obtained from the CDK training set, where the dataset exhibits a more balanced distribution (only 503 out of 1,347 drugs in the training set have a single interaction). The combination of this balanced state and the higher number of training epochs allowed RSCRDL to avoid underfitting, with our approach demonstrating significantly improved performance on this dataset, competing with the results of the NeuMF approach. Furthermore, examination of the RSCRDL training and validation plots reveals that they had not stabilized by the end of the learning process, suggesting the potential for further performance improvements with extended training durations. Focusing on the RSCRDL_F approach, which demonstrated slightly superior results compared to its predecessor model, indicating that the transferred weights, combined with the incorporated explicit drug features, positively influenced the learning process by accelerating stabilization and improving convergence. As for the final model presented in Figure 4.7 that is NeuMF, it's evidently the top performing approach in terms of training loss values on both datasets. However, its validation loss values indicate that these results are attributable to overfitting.

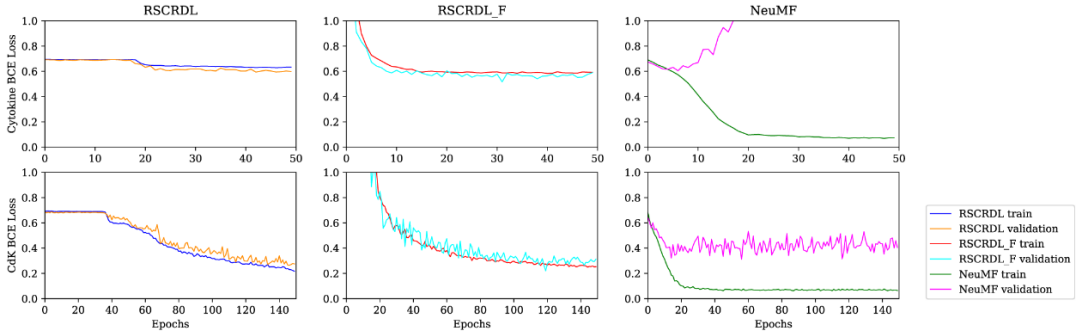



*Figure 4.7 BCE loss values w.r.t number of epochs for the training and validation sets of the cytokine and CDK datasets*

The aforementioned observations are further substantiated by the results presented in Figures 4.8 and 4.9, which now include GLocal-K in the comparison. As observed, NeuMF achieves the highest performance on both training sets. However, its poor validation results clearly indicate that the approach suffered from overfitting—a discrepancy that is particularly pronounced with the cytokine dataset. GLocal-K exhibits a performance that's similar to that of NeuMF, albeit with less impactful results, suggesting that it may also be affected by overfitting. In contrast, our models demonstrate the most stable performance, with training and validation losses remaining closely aligned for each dataset. This stability implies that the latent features learned directly from biological sequences are more effective at representing interactions compared to those extracted from global features, regardless of whether training occurs over short or extended durations. Among all approaches, RSCRDL_F emerges as the top-performing model on both validation sets, achieving the highest F1-score, followed sequentially by RSCRDL, NeuMF, and finally GLocal-K. The final evidence supporting our assumptions is presented in Figure 4.10, which displays each model's Cohen's kappa scores on the validation sets. These results confirm that the overall top-performing model across both datasets is RSCRDL_F, followed by RSCRDL, then NeuMF, and finally GLocal-K. Furthermore, considering that our approaches reached their optimal

performance at late stages of training, we posit that they still possess the potential to achieve even higher results with extended training.

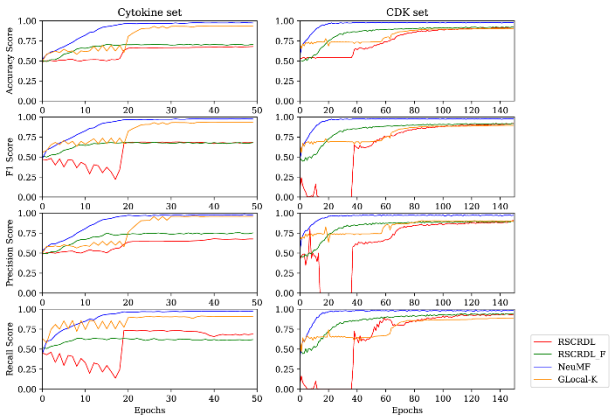



*Figure 4.8 General metrics comparison w.r.t number of epochs for the cytokine and CDK training sets*

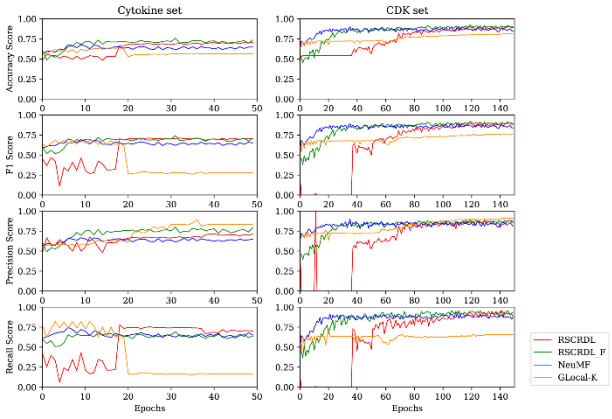



*Figure 4.9 General metrics comparison w.r.t number of epochs for cytokine and CDK validation sets*

Finally, the examination of the evaluation results for all models on both test sets, as presented in Table 4.2, reveals that RSCRDL_F is the top-performing model across both datasets, followed by RSCRDL. This finding indicates that our proposed approaches achieved state-of-the-art performance by effectively exploiting drug-target-related features derived from biological sequences. NeuMF and GLocal-K, on the other hand, demonstrated competitive results despite their general-purpose nature. This outcome is particularly impressive considering that both models operate with little to no domain-specific information, relying instead on global features and interaction patterns without access to the sequential molecular representations that our approaches leverage.

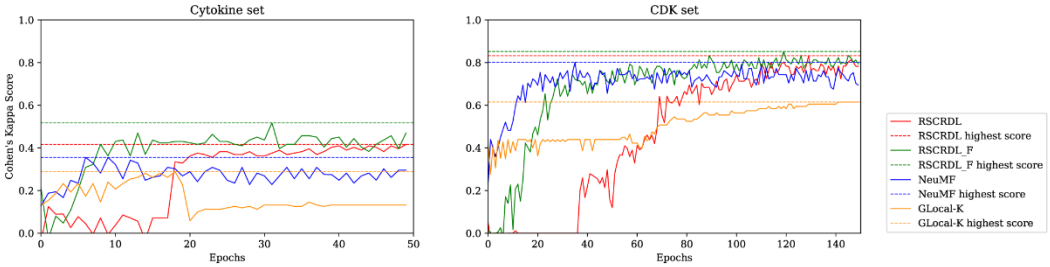



*Figure 4.10 Cohen's kappa score comparison w.r.t number of epochs for cytokine and CDK validation sets*

#### 4.3.3.5. Performance validation

Despite the theoretical sophistication of all trained models, the reliability of their predictions requires investigation through domain-specific tools that extract ground-truth knowledge for explanation, comparison, and validation.

**A. Drugs Analysis**

The first stage of the analysis focuses on drug molecules to help assess their behavioral similarity through structural resemblance. Since the biological activity of drugs is strongly associated with their underlying three-dimensional organization; structure-oriented molecular representations, such as FPs, can be leveraged to estimate similarities between compounds and infer potential functional relationships. In this context, the ECFP6 algorithm was employed due to its demonstrated effectiveness in encoding chemical structures compared with other molecular descriptor approaches [419, 427].

*Table 4.2 Metrics comparison regarding all models for both test sets*

| | Cytokine Test set | | | | CDK Test set | | | |
|---|---|---|---|---|---|---|---|---|
| **Model / Metric** | CRDL | CRDL_F | NCF | GLocal-K | CRDL | CRDL_F | NCF | GLocal-K |
| **Cohen's kappa** | 0.345 | **0.38** | 0.226 | 0.227 | 0.717 | **0.731** | 0.704 | 0.642 |
| **Balanced accuracy** | 0.673 | **0.689** | 0.614 | 0.613 | 0.857 | **0.865** | 0.853 | 0.814 |
| **F1** | **0.664** | 0.646 | 0.623 | 0.569 | 0.843 | **0.853** | 0.84 | 0.781 |
| **Precision** | 0.654 | **0.722** | 0.583 | 0.618 | 0.869 | 0.857 | 0.834 | **0.92** |
| **Recall** | **0.674** | 0.585 | 0.668 | 0.527 | 0.818 | **0.85** | 0.847 | 0.678 |

Following the generation of fingerprint vectors for all compounds, where the ECFP6 bit vector size was fixed at 1024, pairwise structural similarities were computed using Tc. This metric's threshold varies depending on both the used fingerprint strategy and the molecular category under investigation; for drug compounds represented using ECFP6, a threshold value of 0.42 was adopted [428]. Subsequently, insignificant and redundant similarity scores were filtered using the 'RDKit' framework, resulting in a sparse symmetric similarity matrix that can be interpreted as the adjacency matrix of an undirected graph. Within this graph, linked drug nodes are assumed to exhibit structural resemblance and, consequently, may share comparable biological properties or mechanisms of action. The prior structural knowledge inferred from the generated similarity graph was then utilized as an evaluation criterion for the interaction matrices produced by the proposed models. More specifically, each drug $d_i$ was represented as a vector $d_i \in \mathbb{R}^m$, corresponding to its interaction profile with the $m$ protein sequences present in the interaction matrix. These high-dimensional interaction vectors were projected into lower-dimensional spaces using the t-SNE algorithm to facilitate visualization and distributional analysis. To ensure a fair and robust assessment, 225 t-SNE configurations were explored using the Chebyshev distance metric, alongside an additional 225 configurations based on the Correlation distance metric, with each configuration characterized by a distinct set of hyperparameters.

After dimensionality reduction, drug instances were assigned labels according to the connectivity structure of the similarity graph, where compounds belonging to the same connected subgraph shared the same class label, whereas isolated nodes were assigned unique labels. Finally, the quality of the generated projections was assessed through the analysis of intra-cluster

compactness and inter-cluster separability, thereby evaluating how effectively the interaction representations preserved the structural relationships established from the original molecular similarity graph.

- ✓ **Cytokine set:**

The analysis began with the Cytokine dataset, which contains 5054 drugs used to construct the fingerprint-based similarity graph illustrated in Figure 4.11. The generated graph was partitioned into 175 disconnected subgraphs, excluding the $39^{th}$ null subgraph, with component sizes ranging from 2 to 515 nodes. In addition, the null subgraph included 241 isolated drugs that were considered to exhibit unique biological activities. It was also observed that the majority of the generated subgraphs were not fully connected, indicating that structural similarity does not necessarily exist between every pair of drugs belonging to the same component; consequently, only direct neighboring relations were considered during the subsequent deduction stage.

Regarding the low-dimensional drug representations produced through the 450 t-SNE configurations summarized in Tables 4.3 and 4.4, the RSCRDL model demonstrated the most distinguishable clustering behavior. More specifically, it achieved the highest inter-cluster separation score when the Correlation distance metric was employed, while ranking second when the Chebyshev distance metric was used. The RSCRDL_F variant followed closely behind with highly competitive results. Similarly, the GLocal-K approach also exhibited strong performance, as both RSCRDL and GLocal-K alternated between first and second place depending on the selected distance metric. By contrast, NeuMF produced noticeably weaker inter-cluster separability scores under both evaluation settings, which further reflects the overfitting behavior previously observed during the training phase. Concerning intra-cluster compactness, the proposed architectures yielded relatively close and moderately higher values compared with the competing approaches, whereas GLocal-K and NeuMF achieved the lowest intra-cluster distances, indicating tighter grouping among structurally related drugs. Nevertheless, relying solely on low intra-cluster distances is insufficient for assessing the quality of an interaction matrix. Although NeuMF generated compact clusters, its simultaneously low inter-cluster separation resulted in significant overlap between samples belonging to different classes, suggesting that the learned interaction representations were not sufficiently discriminative or biologically informative.

From a visual interpretation perspective, Figures 4.12 and 4.13 illustrate the highest-quality projections produced by each evaluated approach. The representations generated by the GLocal-K model appear to provide the clearest organization of drug compounds, where samples belonging to related categories form relatively compact and distinguishable clusters across the projected space. In contrast, the projections obtained from our proposed models exhibit a different structural behavior, characterized by repetitive trail-like distributions under both distance metrics. More specifically, compounds associated with identical classes tend to form separated sub-clusters distributed along elongated trajectories, which may be attributed to the fact that several compounds grouped within the same similarity subgraph are not necessarily direct structural neighbors, but rather indirectly connected through intermediate relationships. Regarding the NeuMF approach, the projected distributions appear considerably less organized and substantially more difficult to

interpret under both metrics. Drug samples are scattered across the embedding space without forming meaningful or coherent class-oriented structures, suggesting that the learned interaction representations were strongly affected by the sparsity and limited information available during training. Such observations further support the assumption that NeuMF suffered from weaker generalization capabilities compared to the remaining evaluated models, ultimately leading to interaction matrices that were less capable of preserving biologically meaningful structural relationships among compounds.

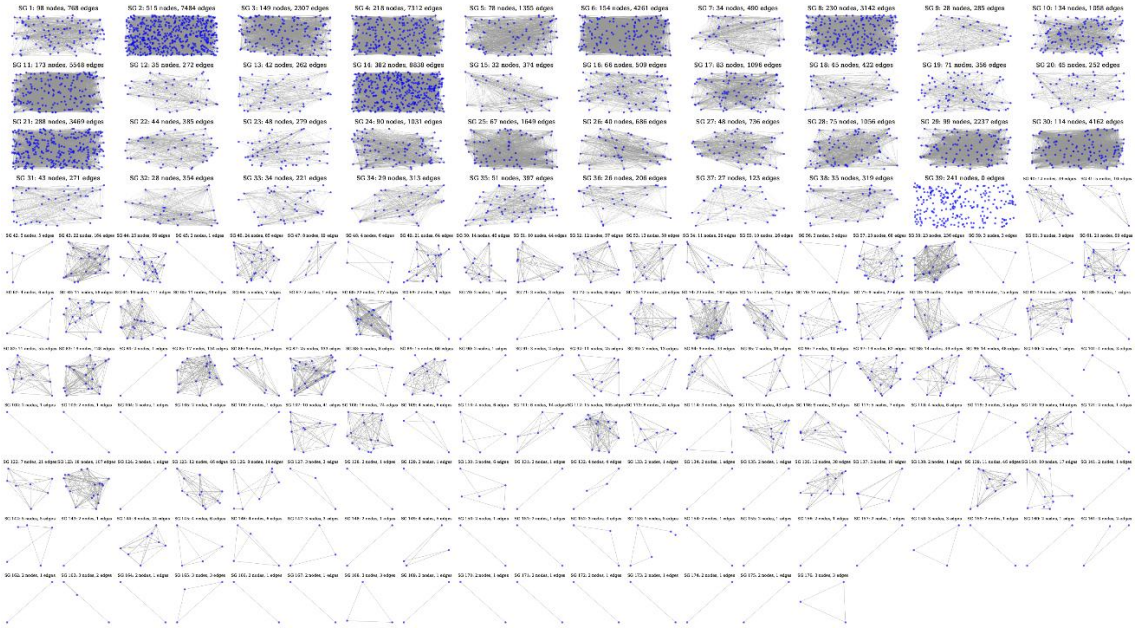


*Figure 4.11 Drugs' fingerprint similarity graph of the cytokine set*

*Table 4.3 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the correlation distance-metric for the cytokine set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| RSCRDL | **31449.9±2984.0** | **35825.5** | 1199.8±105.9 | 421.4 |
| RSCRDL_F | 29351.6±2860.1 | 32470.7 | 1267.2±124.0 | 428.8 |
| NeuMF | 25851.6±6604.7 | 29130.2 | 1265.8±321.2 | **8.8** |
| GLocal-K | 31445.7±4706.5 | 35771.6 | **784.7±123.5** | 153.7 |

*Table 4.4 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the Chebyshev distance-metric for the cytokine set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| RSCRDL | **31077.0±2833.5** | 34804.8 | 1177.5±101.8 | 504.8 |
| RSCRDL_F | 30776.7±3187.3 | 34059.8 | 1135.2±113.8 | 356.4 |
| NeuMF | 26043.9±6194.0 | 28820.1 | 1289.6±307.0 | 27.0 |
| GLocal-K | 14861.1±13142.0 | **34808.6** | **369.3±313.9** | **13.1** |

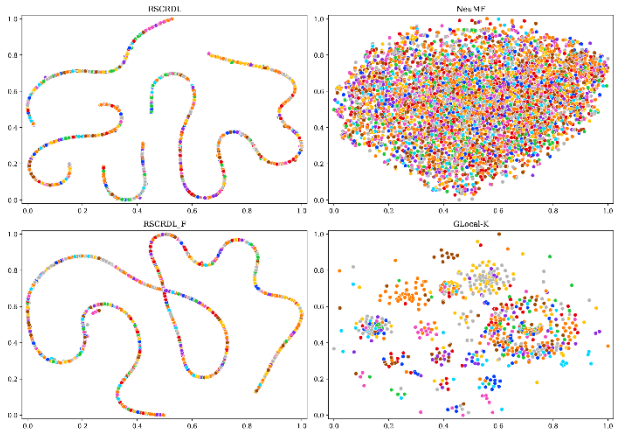



*Figure 4.12 Cytokine set's drugs projections via t-SNE algorithm with correlation distance-metric*

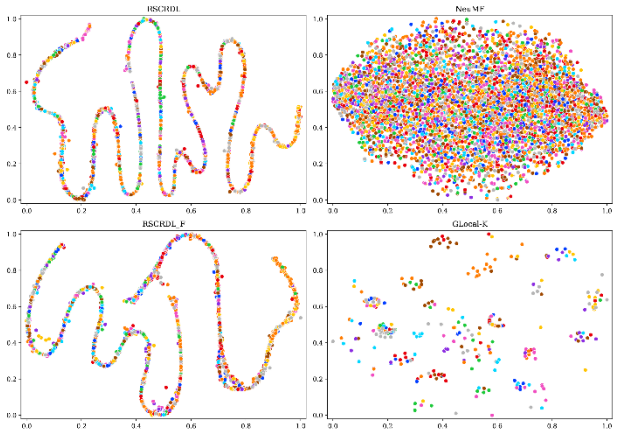



*Figure 4.13 Cytokine set's drugs projections via t-SNE algorithm with Chebyshev distance-metric*

✓ **CDK set:**

Concerning the CDK dataset, the generated fingerprint similarity graph illustrated in Figure 4.14 was constructed from a total of 1476 drug compounds, and exhibited structural characteristics comparable to those observed for the Cytokine set. Nevertheless, the resulting graph was partitioned into a smaller number of connected components, namely 140 subgraphs (excluding the null component), with cluster sizes ranging from 2 to 96 compounds. Most subgraphs are incomplete, indicating the existence of indirect structural relationships between several compounds rather than strict pairwise similarities. Additionally, 225 isolated compounds were grouped within the graph's $11^{th}$ null subgraph, suggesting potentially unique biological or structural behaviors. The quantitative results obtained from the deployed t-SNE configurations, summarized in Tables 4.5 and 4.6, reveal that the four evaluated models achieved relatively close performance levels compared to those observed for the Cytokine dataset. Such behavior suggests that all approaches were able to capture more informative latent representations from the interaction matrices. Nevertheless, our proposed models achieved the highest overall and peak inter-cluster separation values under both the Chebyshev and Correlation distance metrics, highlighting their superior ability to discriminate between structurally distinct drug classes. Although NeuMF produced lower intra-cluster distances, indicating tighter local grouping of similar compounds, its overall projection quality remained less significant due to its weaker inter-class separability.

The visual representations displayed in Figures 4.15 and 4.16 further support these observations. All evaluated models generated projection patterns resembling the periodic trail-like structures previously observed with the Cytokine set, with RSCRDL_F and GLocal-K producing the most coherent and aligned distributions, followed by RSCRDL. In contrast, the NeuMF projections appeared comparatively underdeveloped and less organized, resembling an incomplete version of the patterns generated by the remaining models, which may indicate insufficient learning of interaction context. Furthermore, the clearer emergence of periodic projection structures across all models for the CDK dataset suggests that this dataset provided more exploitable interaction information than the Cytokine dataset. Importantly, our proposed approaches were the only models capable of consistently reproducing these structured projection

behaviors across both datasets, supporting the hypothesis that sequence-driven representations enabled faster and more effective learning of latent drug characteristics and biological behaviors.

*Table 4.5 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the correlation distance-metric for the CDK set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| RSCRDL | 29147.5±2931.8 | 32047.5 | **294.5±29.6** | 111.8 |
| RSCRDL_F | **30510.0±3063.0** | **35393.6** | 322.7±31.7 | 123.2 |
| NeuMF | 29359.5±4731.9 | 31942.2 | 366.1±59.7 | **41.4** |
| GLocal-K | 28828.6±3007.9 | 31857.8 | 377.0±39.6 | 156.6 |

*Table 4.6 Inter and Intra-cluster criteria results of all 225 t-SNE models based on the Chebyshev distance-metric for the CDK set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| RSCRDL | **29363.1±3150.8** | 32727.4 | 299.8±30.5 | 117.0 |
| RSCRDL_F | 28440.6±3095.8 | **32888.6** | **281.0±26.3** | 108.0 |
| NeuMF | 29342.6±4586.8 | 32359.6 | 375.1±58.8 | **83.6** |
| GLocal-K | 28505.0±2761.2 | 31040.8 | 372.3±38.5 | 155.9 |

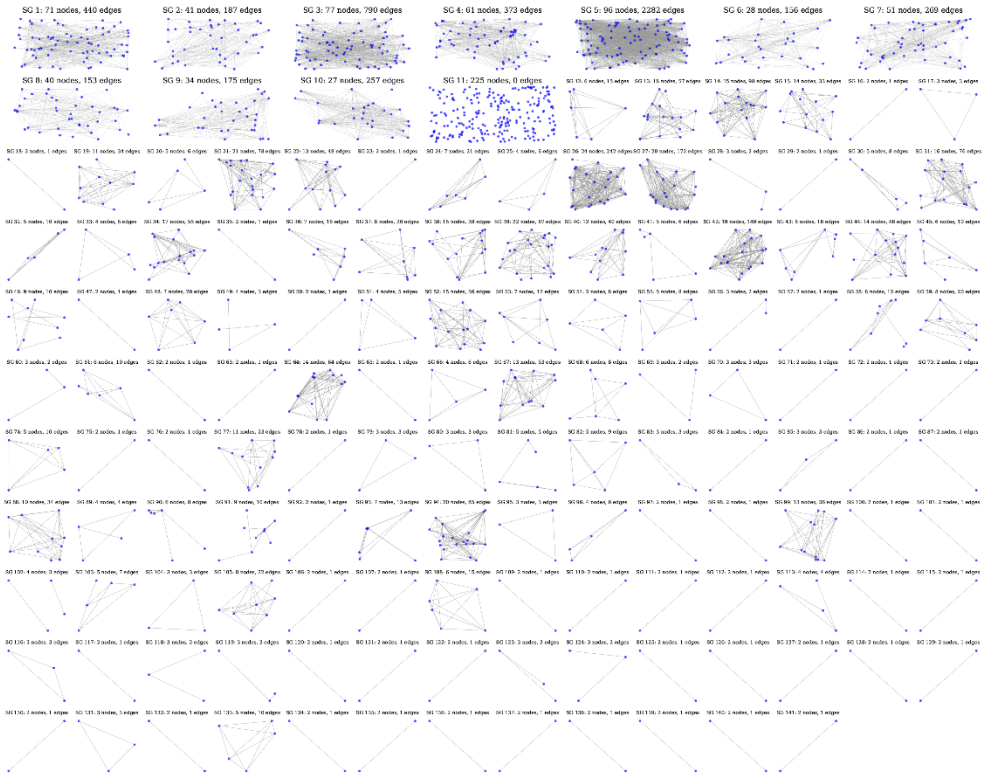


*Figure 4.14 Drugs' fingerprint similarity graph of the CDK set*

## B. Protein sequences analysis

Shifting the analysis toward the more challenging case of protein sequences, where we conducted a composite similarity assessment that jointly integrates both one-dimensional and three-dimensional similarity measurements, corresponding respectively to sequence homology and structural resemblance. The first analysis stage focused on evaluating similarities between AA sequences forming the primary structures of proteins. To accomplish this task, the BLAST algorithm was employed to compute alignment-based homologies between protein sequences through the exploitation of the E-value significance metric, where similarities associated with values below $10^{-6}$ were considered biologically meaningful [429]. The second analysis stage targeted structural similarity assessment and followed a procedure analogous to the one previously adopted for molecular compounds. More specifically, protein structures were represented through $ECFP_6$ fingerprints using a bit-string length fixed at 2048. Since no universally established Tc threshold exists for protein-sequence-derived fingerprints, an empirical threshold selection strategy was adopted based on observations extracted from the drug-oriented experiments. During the previous analyses, only a very small fraction of the computed Tc values—less than 1% of the total similarity scores generated for both the Cytokine and CDK datasets—were found to be sufficiently significant. Guided by these observations, we inferred that an equivalent significance level for protein structural similarities would require a substantially higher threshold value i.e., 0.8, leading us to select ($Tc \geq 0.9$) in order to retain only highly confident structural relationships.

Following the computation of both sequence-based and structure-based similarities, the final protein similarity graph was constructed by intersecting the outputs of the two analysis procedures. Consequently, two protein sequences were considered related only when they simultaneously satisfied both the homology-based and structural similarity criteria, allowing the resulting graph to reflect stronger and biologically more reliable associations between proteins.

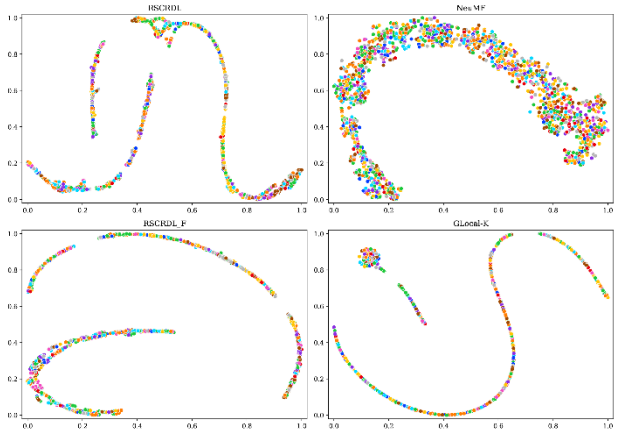



*Figure 4.15 CDK set's drugs projections via t-SNE algorithm with correlation distance-metric*

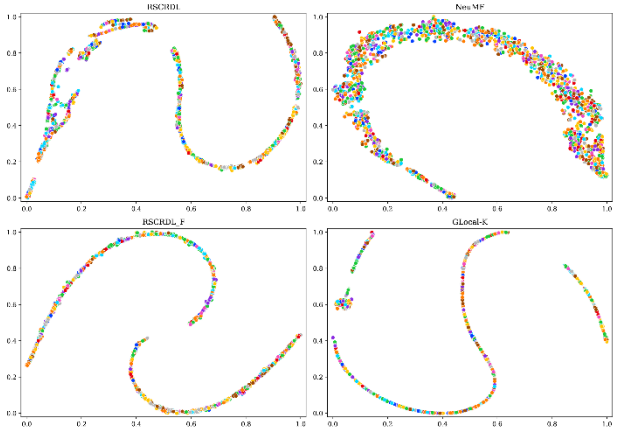



*Figure 4.16 CDK set's drugs projections via t-SNE algorithm with Chebyshev distance-metrics*

✓ **Cytokine set:**

Starting with the Cytokine set and its 60 protein sequences, the generated similarity graph illustrated in Figure 4.17 reveals the formation of several compact groups composed of 2 to 4 proteins independently of their originating organisms. Overall, the graph was partitioned into 11 complete subgraphs. Since the retained edges satisfy the two similarity criteria jointly, connected proteins are expected to exhibit comparable biological behaviors and potentially fulfill analogous

targeting roles during intermolecular interactions. Moreover, 19 out of the graph's 22 edges were associated with E-values equal to zero, indicating extremely strong sequence homology among the connected proteins. In contrast, the null component represented by the $12^{th}$ subgraph contains more than half of the proteins, suggesting that these isolated sequences may possess distinctive biological functionalities and interaction patterns.

The resulting graph structure was subsequently exploited to define class labels during the projection phase of protein interaction vectors through the deployed t-SNE configurations, consisting of 400 models for each previously adopted distance metric. The obtained quantitative results, summarized in Tables 4.7 and 4.8, alongside the most representative visual projections shown in Figures 4.18 and 4.19, clearly demonstrate the superiority of our proposed approaches over the competing methods across all evaluation scenarios. Among all evaluated models, RSCRDL_F achieved the best overall performance, combining the highest global projection quality with the highest and second highest inter-cluster values under the Correlation and Chebyshev metrics respectively, while simultaneously maintaining the lowest intra-cluster distances for both metrics. These findings strongly support the effectiveness of incorporating the pretrained encoder-decoder representations inherited from RSCRDL together with explicit molecular descriptors, which collectively enhanced the model's capability to construct biologically meaningful interaction representations. Although RSCRDL ranked slightly below its enhanced variant, it still achieved highly competitive results that substantially surpassed those obtained by the two general-purpose state-of-the-art RS.

In contrast, the competing approaches produced relatively similar results between themselves, yet remained significantly behind the performance levels reached by our models, further emphasizing the limitations of conventional RS when applied to specialized bioinformatics-oriented interaction prediction tasks. Visual inspection of the generated projections further reinforces these observations. Excluding isolated proteins, RSCRDL successfully grouped proteins belonging to identical classes into coherent and distinguishable clusters under both distance metrics, highlighting the robustness of its learned interaction matrix. RSCRDL_F produced even more organized distributions, where the periodic trail-like projection patterns previously observed during drug analysis reappeared among proteins sharing the same classes. Conversely, the projections generated by the state-of-the-art baselines appeared comparatively scattered and partially random, suggesting that their learned interaction matrices were less effective at capturing latent protein-related features and were likely biased toward exploiting global drug-related characteristics instead.

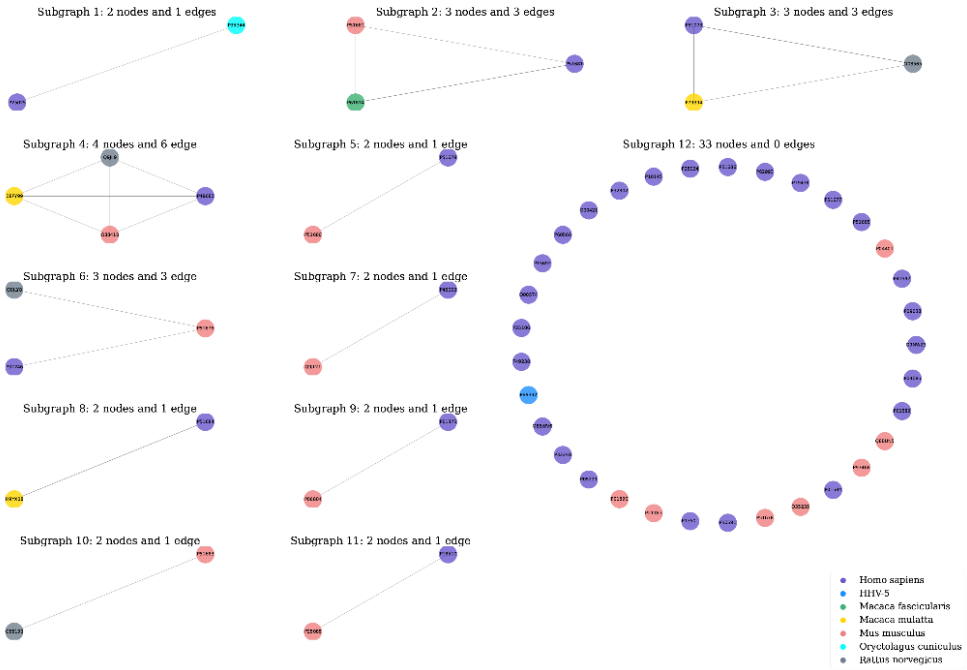



*Figure 4.17 Cytokine set's protein sequences similarity graph*

*Table 4.7 Protein inter/intra-cluster results of all 400 t-SNE models based on the correlation distance-metric for the cytokine set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| RSCRDL | 327.192±113.53 | 492.525 | 3.772±1.09 | 5.418 |
| RSCRDL_F | **331.030±114.30** | **497.458** | **3.164±1.24** | **1.115** |
| NeuMF | 316.309±98.34 | 395.102 | 5.314±1.67 | 6.372 |
| GLocal-K | 311.216±88.47 | 442.215 | 6.091±1.80 | 8.058 |

*Table 4.8 Protein inter/intra-cluster results of all 400 t-SNE models based on the Chebyshev distance-metric for the cytokine set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| RSCRDL | 338.005±110.28 | **544.346** | 3.688±1.06 | 2.317 |
| RSCRDL_F | **358.176±115.02** | 524.739 | **3.265±1.18** | **0.576** |
| NeuMF | 321.250±105.64 | 440.284 | 4.987±1.56 | 6.226 |
| GLocal-K | 309.622±100.33 | 405.736 | 5.358±1.53 | 6.917 |

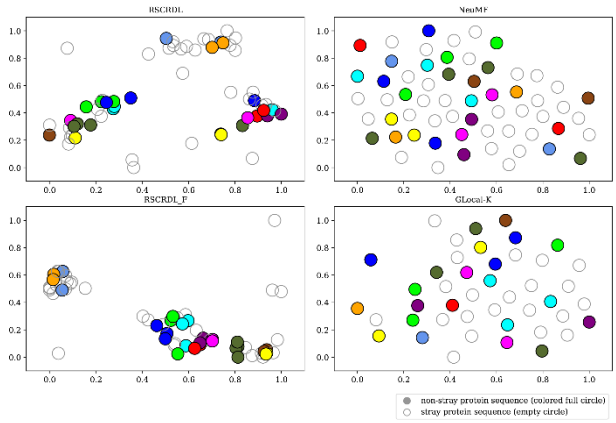



*Figure 4.18 Cytokine set's protein sequences projections via t-SNE algorithm with correlation distance-metric*

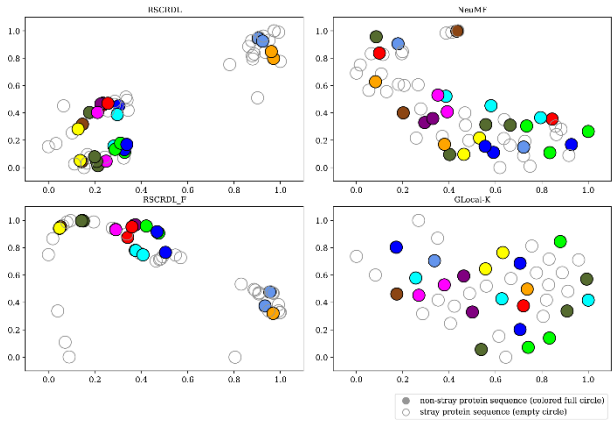



*Figure 4.19 Cytokine set's protein sequences projections via t-SNE algorithm with Chebyshev distance-metric*

✓ **CDK set:**

Regarding the CDK dataset, the similarity graph generated from its 35 protein sequences and illustrated in Figure 4.20 exhibits a markedly sparse connectivity structure compared to the previously analyzed set. Only three edges were retained as biologically significant relationships, with all corresponding alignments being characterized by E-values equal to zero, indicating extremely strong homology between the connected protein pairs.

The quantitative results obtained from the 800 deployed t-SNE models, summarized in Tables 4.9 and 4.10, reveal that the four evaluated approaches achieved relatively comparable performances. Nevertheless, each model demonstrated superiority under specific evaluation settings depending on the adopted distance metric. GLocal-K achieved the best global performance under the Correlation metric, combining the highest overall inter-cluster separation with the lowest intra-cluster distances. In parallel, NeuMF demonstrated more competitive behavior this time by reaching the minimum intra-cluster value and the highest overall inter-cluster value under the Correlation and Chebyshev metrics respectively, indicating improved local grouping and class discrimination capabilities for this dataset. Concerning our proposed approaches, RSCRDL achieved the most favorable results under the Chebyshev distance metric by simultaneously producing the best inter and intra-cluster values, whereas RSCRDL_F obtained the highest inter-cluster score under the Correlation metric. Furthermore, the visual projections displayed in Figures 4.21 and 4.22 indicate that all evaluated models produced relatively similar distribution patterns, with no highly distinguishable structural differences between the generated projections, which may be attributed to the limited number of meaningful protein associations available within this dataset.

*Table 4.9 Protein inter/intra-cluster results of all 400 t-SNE models based on the correlation distance-metric for the CDK set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| RSCRDL | 188.959±46.19 | 242.753 | 1.281±0.46 | 1.517 |
| RSCRDL_F | 189.830±40.28 | **251.270** | 1.277±0.47 | 1.234 |

| | | | | |
|---|---|---|---|---|
| **NeuMF** | 185.368±52.00 | 247.217 | **1.234±0.45** | 1.262 |
| **GLocal-K** | **199.733±38.31** | 219.984 | 1.377±0.42 | **1.219** |

*Table 4.10 Protein inter/intra-cluster results of all 400 t-SNE models based on the Chebyshev distance-metric for the CDK set*

| Criterion | Inter-cluster distance | | Intra-cluster distance | |
|---|---|---|---|---|
| Feature Model | Mean ± std | Best value (max) | Mean ± std | Best value (min) |
| **RSCRDL** | 192.391±41.51 | **255.296** | **1.279±0.50** | **1.170** |
| **RSCRDL_F** | 189.607±41.03 | 245.633 | 1.284±0.51 | 1.269 |
| **NeuMF** | **205.950±43.40** | 255.086 | 1.467±0.44 | 1.67 |
| **GLocal-K** | 195.687±48.26 | 253.810 | 1.350±0.48 | 1.597 |

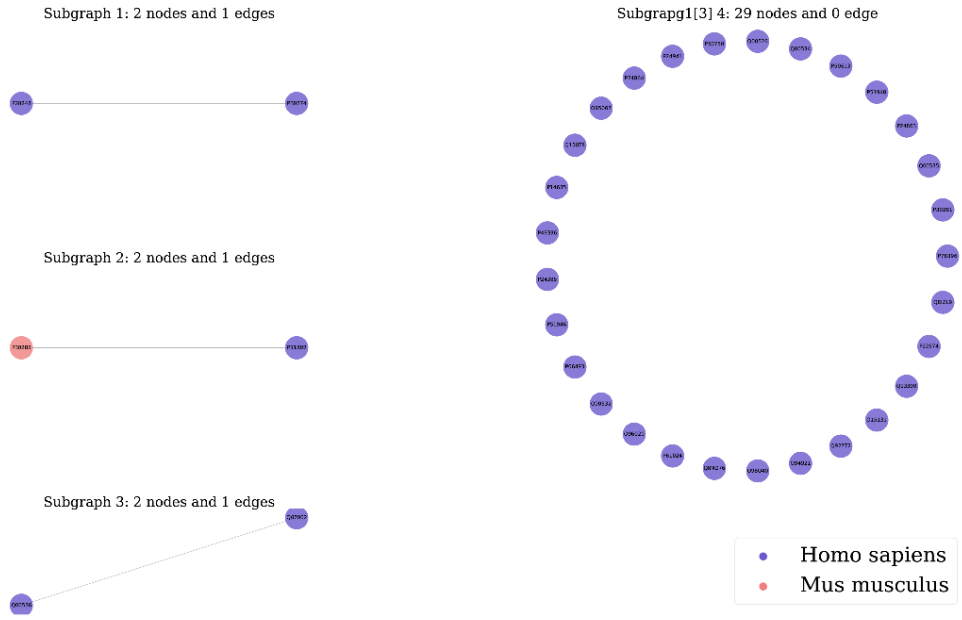



*Figure 4.20 CDK set's protein sequences similarity graph*

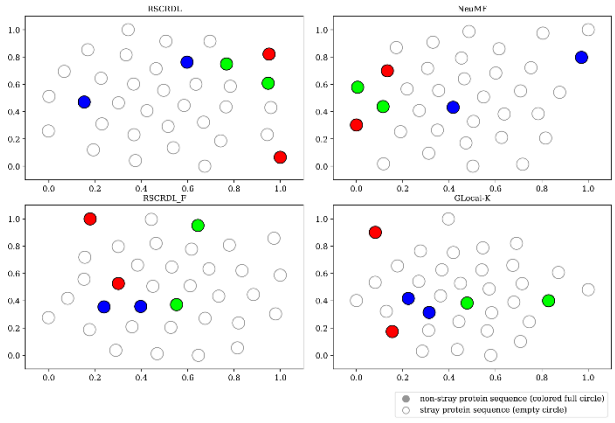



*Figure 4.21 CDK set's protein sequences projections via t-SNE algorithm with correlation distance-metric*

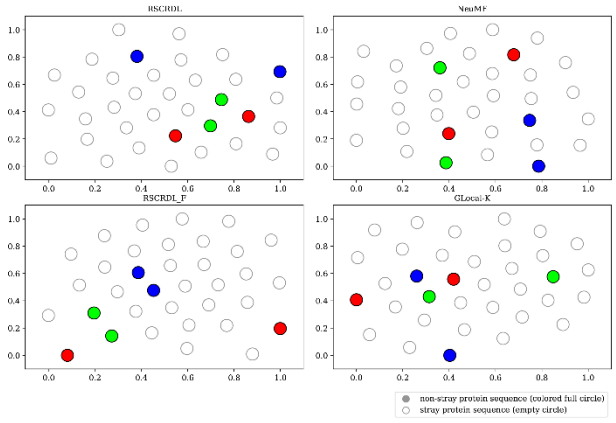



*Figure 4.22 CDK set's protein sequences projections via t-SNE algorithm with Chebyshev distance-metric*

#### 4.3.3.6. The application of RSCRDL and RSCRDL_F in drug repurposing strategies

By exploiting the ability of our proposed models to infer missing associations and generate dense interaction matrices, novel candidate interactions can be deduced through the combination of: the ability to render sparse interaction matrices full, the similarity graphs generated for both proteins and drugs, and the experimentally validated positive interactions available within each dataset. The adopted deduction strategy, illustrated in Figure 4.23, propagates known interactions across biologically or structurally similar entities while ensuring consistency with the predictions generated by our approaches. More specifically, a protein sequence is assumed to potentially interact with neighboring drugs that are structurally similar to compounds with which it already possesses validated positive interactions, provided that such deductions do not contradict the predictions of our models—Eq. (4.23). Similarly, the reverse deduction process assumes that a drug may interact with proteins that are biologically related to its known targets, thereby enabling both drug-based and protein-based interaction inference.

$$\begin{gathered}\forall\, p_i \in PSG_k, d_j \in DSG_l: \\ if\ \exists\ Valid_Positive_Interaction(p_i, d_j) \\ and\ RSCRDL(p_i, N(d_j)) \geq th \\ and\ RSCRDL_F(p_i, N(d_j)) \geq th \\ then\ \exists Candidate_Positive_Interaction(p_i, N(d_j)) \\ and\ if\ RSCRDL(N(p_i), d_j) \geq th \\ and\ RSCRDL_F(N(p_i), d_j) \geq th \\ then\ \exists Candidate_Positive_Interaction(N(p_i), d_j)\ \}\end{gathered} \qquad \text{Eq (4.31)}$$

Within this formation, $p_i$ denotes a protein sequence belonging to the protein subgraph $PSG_k$, which exhibits a $Valid_Positive_Interation$ with a drug $d_i$ that is part of the drug subgraph $DSG_l$. $RSCRDL$ and $RSCRDL_F$ represent the predicted interaction scores produced by our proposed models for a given protein–drug pair, while $th$ denotes a predefined confidence threshold used to filter the resulting candidates, with higher values indicating more reliable predictions. Finally, $N()$ refers to the set of neighboring molecules associated with a given entity within its corresponding similarity subgraph.

Starting with the Cytokine set, where Eq. (4.31) was applied using a confidence threshold fixed at $0.65$, which resulted in the inference of one protein-oriented candidate interaction and $140$ drug-oriented candidate interactions, while excluding deductions involving non-human protein sequences. The inferred protein-based interaction concerns the protein "Interleukin-1 receptor antagonist" (P18510), which is involved in the regulation of several interleukin-1-mediated inflammatory and immune mechanisms [430]. Additionally, this protein is associated with multiple pathological conditions, including inflammatory disorders, diabetes-related complications, gastric cancer, and several cancer prognostic processes [430-433]. Furthermore, P18510 exhibited strong similarity with its neighboring protein P25085 (mouse), reflected by an E-value of $10^{-106}$ and a fingerprint similarity score of $0.915$, and since the latter possesses a validated positive interaction with CHEMBL3416692 that's characterized by an $IC50$ value of $52nM$; both RSCRDL and RSCRDL_F estimations that reached the approximately 68.6% are further strengthened.

Concerning the 140 drug-based inferred interactions, they involve six distinct human protein sequences and 135 drug compounds. A substantial portion of these candidates targeted the protein "C-C chemokine receptor type 5" (P51681), which plays an important role in inflammatory chemokine signaling, HIV-1 co-receptor activity, and T-lymphocyte migration toward infected tissues, while also being associated with type 1 diabetes mellitus-related disorders [434]. Among the inferred candidates, nine compounds satisfied all rule-of-five constraints without violations. One notable example is CHEMBL1782578, for which two neighboring molecules, namely CHEMBL1683190 and CHEMBL1683074, possess experimentally validated positive interactions with P51681, characterized by $IC50$ values of $5.3nM$ and $11.1nM$, respectively. Additionally, CHEMBL1782578 demonstrated fingerprint similarity scores of 0.538 and 0.74 with these neighboring compounds, further reinforcing the plausibility of the inferred interaction being positive as hypothesized by RSCRDL and RSCRDL_F, with probability scores reaching the values of 68.801% and 72.274%, respectively. The five most confident drug-based candidate interactions involving compounds with no rule-of-five violations are summarized in Table 4.11 according to the confidence estimates generated by RSCRDL_F.

As for the 140 drug-based candidates (which involve 6 different human protein sequences and 135 drugs), 43 of them focused on 'C-C chemokine receptor type 5' (P51681), which acts as an inflammatory CC-chemokines receptor, a co-receptor for human immunodeficiency virus-1, and a contributor during the migration of T-lymphocytes to infected sites, while being associated with 'Type 1 diabetes mellitus 22' [434]. Nine of P51681's candidate interactions revolve around drugs with no rule-of-5 violations, such as CHEMBL1782578 that 2 out of its 35 neighbors (CHEMBL1683190 and CHEMBL1683074) being confirmed to have valid positive interactions with P51681 with IC50 values of 5.3 and 11.1 $nM$. And with CHEMBL1782578 having the FP similarity values of 0.538 and 0.74 with its 2 aforementioned neighbors; our models' assumptions regarding its candidate positive interaction with P51681 are further strengthened, with probabilities reaching 68.801% and 72.274% as anticipated by RSCRDL and RSCRDL_F respectively. With top-5 most confident drug-based candidates concerning drugs with no rule-of-5 violations being depicted in table 4.11 (sorted according to the estimations of $RSCRDL_F$).

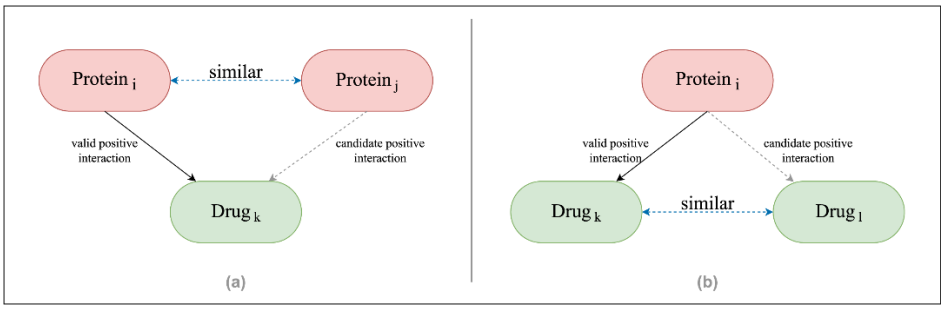



*Figure 4.23 Candidate interactions deduction process*

*Table 4.11 Top-5 drug-based candidate positive interactions for the protein sequences of the cytokine set*

| Protein | Candidate drug | Neighbors(candidate) | ECFP6(candidate, neighbors(candidate)) | RSCRDL | RSCRDL_F |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| P51681 | CHEMBL1782578 | CHEMBL1683190 CHEMBL1683074 | 0.538, 0.74 | 68.81% | 72.27% |
| | CHEMBL1782567 | CHEMBL1683190 CHEMBL1683074 | 0.472, 0.689 | 68.93% | 71.98% |
| | CHEMBL1683077 | CHEMBL1683190 CHEMBL1683074 | 0.83, 0.708 | 68.62% | 71.58% |
| | CHEMBL1782570 | CHEMBL1683190 CHEMBL1683074 | 0.513, 0.745 | 68.92% | 71.4% |
| | CHEMBL2236475 | CHEMBL1683074 | 0.574 | 68.83% | 71.33% |

Proceeding to the CDK set, where the confidence threshold ($th$) of Eq. (4.31) was increased to 0.8, resulting in the inference of one protein-oriented candidate interaction. This candidate involved the protein "Cyclin-A2" (P20248), a key regulator of cell-cycle progression [430] that has been associated with several cancer-related pathologies, including retinoblastoma and adenocarcinoma [430], while also being identified as an unfavorable prognostic biomarker in multiple cancer types such as renal, pancreatic, liver, lung, and endometrial cancers [433]. Based on the predictions generated by RSCRDL and RSCRDL_F, P20248 was hypothesized to positively interact with CHEMBL4083619 with confidence scores reaching 84.32% and 80.01%, respectively. This deduction is supported by the fact that CHEMBL4083619 exhibits a validated positive interaction with the bovine protein P30274 that's characterized by an $IC50$ value of $52nM$, while P20248 shares extremely strong similarity with P30274, reflected by an E-value equal to zero and a fingerprint similarity score of 0.945.

Regarding the drug-oriented deductions, a total of 426 candidate interactions were inferred, involving 22 human protein sequences and 79 molecular compounds. One representative example concerns the protein "G2/mitotic-specific cyclin-B1" (P14635), which plays a major role during cell-cycle regulation [434] and has been linked to several cancer-related conditions, including retinoblastoma and laryngeal squamous cell carcinoma, while also serving as an unfavorable prognostic marker in lever, renal, and pulmonary cancers [433]. For this protein, 24 candidate interactions involving compounds with zero to two rule-of-five violations were identified, including the interaction with CHEMBL4443261, which satisfies all rule-of-five constraints. Furthermore, four neighboring compounds among the 62 neighbors of CHEMBL4443261 possess experimentally validated positive interactions with P14635, characterized by $IC50$ values ranging from 55 to $90nM$. These neighboring molecules (CHEMBL4078060, CHEMBL4094561, CHEMBL4548580, and CHEMBL4063500) also exhibit fingerprint similarity scores of 0.458, 0.462, 0.576, and 0.475 with the candidate drug, respectively, thereby reinforcing the plausibility of the inferred interaction being positive—RSCRDL and RSCRDL_F probabilities reached the values of 94.767% and 81.737%, respectively. The five most confident drug-oriented candidates involving compounds without rule-of-five violations are presented in Table 4.12 according to the confidence scores generated by RSCRDL_F.

The final evaluation stage focused on orphan human protein sequences that were identified as biologically related neighbors to proteins already present within the studied datasets. In this context, Eq. (4.32) was applied in order to infer plausible interactions for these previously unobserved proteins and further evaluate the generalization capabilities of the proposed models.

$$\exists\ Candidate_Positive_Interaction(orphan_p, drug_i)\ if:\ RSCRDL(orphan_p, drug_i) \geq th\ and\ RSCRDL_F(orphan_p, drug_i) \geq th\ and\ \exists\ p_j \in Neighbors(orphan_p):\ RSCRDL(p_j, drug_i) \geq th\ and\ RSCRDL_F(p_j, drug_i) \geq th\ and\ \exists\ Valid_positive_interaction(p_j, drug_i) \quad \text{Eq (4.32)}$$

*Table 4.12 Top-5 drug-based candidate positive interactions for the protein sequences of the CDK set*

| Protein | Candidate drug | Neighbors(candidate) | ECFP6(candidate, neighbors(candidate)) | RSCRDL | RSCRDL_F |
|---|---|---|---|---|---|
| P14635 | CHEMBL4443261 | CHEMBL4078060 CHEMBL4094561 CHEMBL4548580 CHEMBL4063500 | 0.458, 0.462, 0.576, 0.475 | 94.77% | 81.74% |
| P24941 | | | | 94.78% | 81.74% |
| P51948 | | CHEMBL4548580 | 0.576 | 94.78% | 81.74% |
| P06493 | | CHEMBL4078060 CHEMBL4094561 CHEMBL4548580 CHEMBL4063500 | 0.458, 0.462, 0.576, 0.475 | 94.78% | 81.74% |
| Q00535 | | | | 94.78% | 81.74% |

The first orphan protein sequence investigated in this study corresponds to "Atypical Chemokine Receptor 4" (Q9NPB9), which demonstrated strong biological similarity with 21 protein sequences belonging to the Cytokine set. The observed similarities were characterized by E-values ranging from $2.10^{-31}$ to $1.10^{-73}$, alongside fingerprint similarity scores varying between 0.8 and 0.854. Biologically, this protein is known to participate in the migration processes of both immune and cancer cells [434], associated with disorders such as Sézary syndrome and immune deficiency diseases [430], and has been identified as an overexpressed biomarker in stomach cancer [433]. By processing Q9NPB9 against the 5054 compounds of the Cytokine set through our proposed frameworks, a total of 234 candidate interactions were inferred, including 68 interactions involving compounds that satisfy all rule-of-five constraints. One representative candidate interaction concerns the compound CHEMBL1683074, which exhibits experimentally validated positive interactions with both O55193 (rat) with an $IC50$ value of $2.9nM$, and P41597 (human) with an $IC50$ range of $3.0{\sim}3.9nM$. Moreover, both O55193 and P41597 share strong similarity relationships with Q9NPB9, reflected by the E-values of $2.10^{-54}$ and $3.10^{-55}$, as well as fingerprint similarity scores of 0.817 and 0.815, respectively. These observations further support the interaction probabilities predicted by RSCRDL and RSCRDL_F for the Q9NPB9-CHEMBL1683074 pair, which reached 68.92% and 71.97%, respectively. The five most confident inferred candidates involving compounds without rule-of-five violations are summarized in Table 4.13 according to the confidence estimates generated by RSCRDL_F.

*Table 4.13 Top-5 candidate positive interactions regarding Q9NPB9 and drugs with no rule-of-5 violations*

| Orphan | Neighbors (orphan) | E-value(orphan, Neighbors(orphan)) | fingerprint(orphan, Neighbors(orphan)) | drug | RSCRDL | RSCRDL_F |
|---|---|---|---|---|---|---|
| Q9NPB9 | P41597 | $3.10^{-55}$ | 0.815 | CHEMBL1683063 | 68.91% | 72.68% |
| | | | | CHEMBL1683064 | 68.88% | 72.68% |
| | | | | CHEMBL1683059 | 68.64% | 72.38% |
| | | | | CHEMBL1782578 | 68.81% | 72.27% |
| | | | | CHEMBL1782567 | 68.93% | 71.98% |

Regarding the orphan protein sequences associated with the CDK dataset, our analysis enabled the inference of candidate interactions for three previously uncharacterized targets. Among them, particular attention was given to "Cyclin-dependent kinase 17" (F5H6Z0), which has been linked to both "Spastic Paraplegia 16, X-Linked" and "Spastic Paraplegia 12, Autosomal Dominant" disorders [430], while also being identified as an unfavorable prognostic biomarker in renal cancer [433]. Based on its strong resemblance to 18 neighboring protein sequences, characterized by E-values ranging from 0 to $4.10^{-49}$ and fingerprint similarity scores between 0.811 and 0.911; a total of 203 potential interactions were inferred for F5H6Z0. Among these inferred interactions, 178 involved compounds satisfying the rule-of-five criteria, including CHEMBL4540910. This compound already demonstrated experimentally validated positive interactions with four neighboring proteins of F5H6Z0, namely Q00535, Q00526, P24941, and P06493, with inhibition scores of 88%, 73%, 71%, and 51%, respectively. Moreover, the orphan target exhibited highly significant sequence and structural similarities with these neighboring proteins, with E-values of $3.10^{-120}$, $10^{-113}$, $6.10^{-111}$, and $3.10^{-117}$, alongside fingerprint similarity scores of 0.817, 0.822, 0.815, and 0.813. These observations further reinforce the predictions generated by both RSCRDL and RSCRDL_F, which estimated the probability of CHEMBL4540910 inhibiting more than 50% of F5H6Z0 activity at 96.39% and 80.66%, respectively. The five most confident candidate interactions involving compounds without rule-of-five violations are summarized in Table 4.14 according to the confidence scores produced by RSCRDL_F.

*Table 4.14 Top-5 candidate positive interactions regarding F5H6Z0 and drugs with no rule-of-5 violations*

| Orphan | Neighbors (orphan) | E-value(orphan, Neighbors(orphan)) | fingerprint(orphan, Neighbors(orphan)) | Candidate drug | RSCRDL | RSCRDL_F |
|---|---|---|---|---|---|---|
| F5H6Z0 | P11802 | $2.10^{-67}$ | 0.835 | CHEMBL545295 | 95.65% | 82.37% |
| | Q00535 Q00534 Q00526 P24941 | $3.10^{-120}$ $7.10^{-69}$ $1.10^{-113}$ $6.10^{-111}$ $2.10^{-67}$ $3.10^{-117}$ | 0.817 0.837 0.822 0.815 0.835 0.813 | CHEMBL1909343 | 90.7% | 82.27% |

| | | | | | | |
|---|---|---|---|---|---|---|
| | P11802 P06493 | | | | | |
| | Q00535 P50613 Q00526 P24941 P06493 | $3.10^{-120}$ $2.10^{-72}$ $1.10^{-113}$ $6.10^{-111}$ $3.10^{-117}$ | 0.817 0.811 0.822 0.815 0.813 | CHEMBL 23254 | 92.59% | 81.95% |
| | P24941 | $6.10^{-111}$ | 0.815 | CHEMBL 4160587 | 93.2% | 81.92% |
| | Q00535 P50613 P50750 P24941 P06493 | $3.10^{-120}$ $2.10^{-72}$ $3.10^{-68}$ $6.10^{-111}$ $3.10^{-117}$ | 0.817 0.811 0.819 0.815 0.813 | CHEMBL 14762 | 92.24% | 81.9% |

Another orphan protein sequence investigated in this study was the "Cyclin P" (Q9H8S5), which is predominantly localized within the cellular nucleus, where it participates in cell-cycle regulation, cellular proliferation, migration processes, mitotic phase transitions, and the regulation of cyclin-dependent serine/threonine kinase activity [430]. By exploiting the similarity relationships established with five neighboring proteins identified within the CDK set, four of these homologous sequences contributed to the inference of 81 potential candidate interactions. Notably, 80 of the inferred candidates involved compounds that satisfied the rule-of-five constraints without violations. These interaction hypotheses were derived using evidence gathered from one to three neighboring proteins simultaneously, as summarized in Table 4.15, where the five most confident candidate interactions are ranked according to the confidence scores produced by the RSCRDL_F framework for compounds exhibiting no rule-of-five violations.

*Table 4.15 Top-5 candidate positive interactions regarding Q9H8S5 and drugs with no rule-of-5 violations*

| Orphan | Neighbors (orphan) | E-value(orphan, Neighbors(orphan)) | fingerprint(orphan, Neighbors(orphan)) | drug | RSCRDL | RSCRDL_F |
|---|---|---|---|---|---|---|
| Q9H8S5 | P78396 P20248 P14635 | $2.10^{-24}$ $4.10^{-23}$ $3.10^{-29}$ | 0.821 0.807 0.809 | CHEMBL23254 | 92.629% | 81.952% |
| | P20248 | $4.10^{-23}$ | 0.807 | CHEMBL4160587 | 93.229% | 81.925% |
| | P78396 P20248 P14635 | $2.10^{-24}$ $4.10^{-23}$ $3.10^{-29}$ | 0.821 0.807 0.809 | CHEMBL14762 | 92.276% | 81.902% |
| | P20248 | $4.10^{-23}$ | 0.807 | CHEMBL2397859 | 95.43% | 81.78% |
| | P78396 P20248 P14635 | $2.10^{-24}$ $4.10^{-23}$ $3.10^{-29}$ | 0.821 0.807 0.809 | CHEMBL1738742 | 89.617% | 81.759% |

The last orphan protein sequence analyzed in this study is "Cyclin-I" (Q14094), which is known for its characteristic oscillatory expression patterns across the different phases of the cell cycle [430]. It is also reported as a favorable prognostic biomarker in renal cancer [433],

highlighting its potential relevance in disease progression assessment. Leveraging similarity links with three out of its four corresponding neighbors in the CDK set, a total of 68 candidate interactions were inferred, among which, 67 satisfy the rule-of-five constraints without violations. Q14094 exhibits interaction patterns comparable to those observed for the previously discussed orphan sequence Q9H8S5, with which it shares a notable similarity characterized by an E-value of $10^{-16}$ and a fingerprint similarity score of 0.795, which is close to the threshold established in our experimental setting. The five most confident candidate interactions, restricted to rule-of-five compliant compounds, are presented in Table 4.16 and ranked according to the prediction scores produced by the RSCRDL_F model.

*Table 4.16 Top-5 candidate positive interactions regarding Q14094 and drugs with no rule-of-5 violations*

| Orphan | Neighbors (orphan) | E-value(orphan, Neighbors(orphan)) | fingerprint(orphan, Neighbors(orphan)) | Candidate drug | RSCRDL | RSCRDL_F |
|---|---|---|---|---|---|---|
| Q14094 | P78396 P20248 | 10-08 2.10-09 | 0.828 0.827 | CHEMBL405039 | 95.823 % | 81.733% |
| | P20248 | 2.10-09 | 0.827 | CHEMBL4060451 | 91.422 % | 81.725% |
| | P20248 | 2.10-09 | 0.827 | CHEMBL4060610 | 88.972 % | 81.682% |
| | P20248 | 2.10-09 | 0.827 | CHEMBL4207368 | 96.055 % | 81.525% |
| | P20248 | 2.10-09 | 0.827 | CHEMBL4212850 | 96.234 % | 81.461% |

## 4.4. Second proposition: Transformer-based recommender system and translator approaches for intermolecular interaction prediction and drug generation as matrix completion and matrix expansion tasks

In this section, we present our second pair of proposed approaches, which leverage transformer architectures to address intermolecular interaction prediction and *de novo* drug generation as complementary matrix completion and matrix expansion tasks. Motivated by the limitations of conventional RSs in modeling the high complexity and low-profile nature of molecular entities, these approaches extend the RS paradigm beyond retrospective interaction retrieval toward proactive molecular discovery. As illustrated in Figure 4.25, we introduce a transformer-based recommender system (RSSSTr) that estimates unobserved intermolecular interactions by jointly exploiting sequential and structural molecular representations together with historical interaction data, for more prominent matrix completion task. In parallel, we propose an enhanced transformer-based translator (TSSTr) designed to generate novel drug candidates conditioned on protein sequences and their mutated variants (Figure 4.26), effectively enabling matrix expansion through the introduction of new items into the interaction space. Central to both models is an architectural strategy that efficiently encodes structural molecular information via our novel proposed fingerprint indexation mechanism (Figure 4.24), allowing both frameworks to exploit molecular fingerprints in a scalable and expressive manner. By coupling generative and recommendation capabilities within a single coherent framework, RSSSTr and TSSTr operate synergistically: the

translator expands the chemical search space with biologically plausible candidates, while the recommender evaluates and prioritizes their relevance to target proteins. Together, these models demonstrate how transformer-based systems can bridge interaction prediction and molecular design, offering a scalable, data-driven solution for advanced drug discovery and repurposing in data-sparse and rapidly evolving biomedical contexts.

### 4.4.1. FP_indexation method

Conventional tokenization method of sequential data involves mapping the discrete tokens forming raw sequences to numerical values (Figure 4.24 (a)), which are then projected into a latent space. This approach is limited to sequences that are composed of value-dependent tokens, where the context of a token depends solely on its identity while completely ignoring its position (e.g., the word 'cat' retains constant meaning regardless of its position in a sentence, thus will always be identified by the same token). In contrast, FP vectors represent value-and-position-dependent sequences, where both identity and position of a token determine its context (all '0's indicate structural feature absence, but the $0^i$ and $0^j$ correspond to distinct missing structural features, hence mustn't be characterized by the same token, same goes for all the '1's), meaning that standard tokenization method would falsely map all tokens that resemble structural feature absence ('0's) using only one token and so on (Figure 4.24 (b)), resulting in feeding erroneous structural feature embeddings to the model, and causing it to propagate incorrect information.

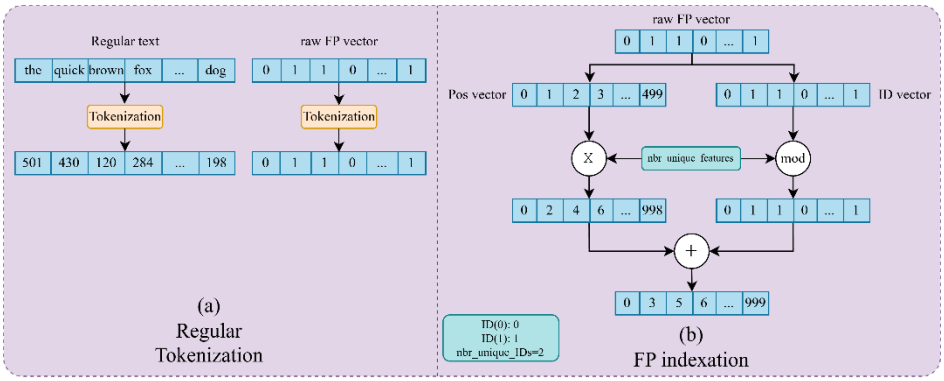



*Figure 4.24 Difference between tokenizing a binary FP vector via (a) a regular tokenization method, and (b) via our FP indexation method*

To resolve such issue; we developed a novel tokenization method that incorporates both identities and positions of tokens in sequences for more faithful representations of FP vectors without any context loss, as depicted in Eq (4.33).

$$FP_indexation(V_{FP}) = Concat\begin{pmatrix} Position(feature_i) \times nbr_unique_IDs \\ +ID(feature_i) mod(nbr_unique_IDs) \end{pmatrix} \quad for\ feature_i \in V_{FP} \qquad \text{Eq (4.33)}$$

Where $Position$ is the position of $feature_i$ in the FP vector $V_{FP}$ (with positions starting from 0), $ID$ is a token's context value (the value usually leveraged in conventional tokenization method), with features sharing the same values having the same $ID$ which must be an integer value that starts from 0 (note that we included the $Mod$ operation solely to ensure that $ID$ values do actually

start from 0, hence it can be ignored), with $nbr_unique_IDs$ being the total number of unique $ID$s (in the case of FP vectors its 2). The application of this method on a binary FP vector would generate a total of $nbr_unique_IDs \times length(V_{FP})$ unique tokens with only $length(V_{FP})$ tokens being present in the resulting tokenized FP vector, meaning that the application of Eq (4.33) on a 500-character-long binary FP vector would generate a total of 1000 unique representations regarding both tokens that have the $ID$ values of 0 or 1 (with the former being characterized by even indices, while the latter being characterized by odd indices) as depicted in Figure 4.24 (b). Additional example that further demonstrates the application of our $FP_indexation$ approach can be witnessed in Table 4.17, where we compute all possible combinations of its application on a 3 character-long vector that consists of 2 unique values 'A', and 'B', from which we can deduce the tokenized vector of $[B, A, B]$ which would be the combination of token 'B' in the $0^{th}$ position, token 'A' in the $1^{st}$ position, and token 'B' in the $2^{nd}$ position, that is $\{1, 2, 5\}$.

*Table 4.17 Demonstration of the tokenization process of a 3-character-long binary vector via our FP_indexation method*

| Feature | ID(Feature) | Num_unique_IDs | Possible Position | Position*Num_unique_IDs | ID + (Position*Num_unique_IDs) |
|---|---|---|---|---|---|
| A | 0 | 2 | 0 | 0 | 0 |
| | | | 1 | 2 | 2 |
| | | | 2 | 4 | 4 |
| B | 1 | | 0 | 0 | 1 |
| | | | 1 | 2 | 3 |
| | | | 2 | 4 | 5 |

### 4.4.2. The architecture of RSSSTr

Knowing that the behaviors of molecules are determined by their structural features that can be represented both sequentially and structurally; we decided to approach a pair of molecules' interaction $V_{interaction} \in \mathbb{R}^{n \times d_model}$ as the summation of their encoded sequential ($V_{seq_1} \in \mathbb{R}^n$, $V_{seq_2} \in \mathbb{R}^n$) and structural ($V_{FP_1} \in \mathbb{R}^n$, $V_{FP_2} \in \mathbb{R}^n$) feature vectors that are processed parallelly via RSSSTr's four encoders $SeqEnc_1()$, $SeqEnc_2()$, $StrEnc_1()$, and $StrEnc_2()$, which are employed in order to extract as much latent features as possible as depicted in Eq (4.34), with $Emb()$ being an embedding layer, $Tokenize()$ a regular tokenization method, $FP_indexation()$ our FP indexation method that's depicted in Eq (4.34), $Pos_info \in \mathbb{R}^{n \times d_model}$ a positional information vector, $n$ the unified sequence length, and $d$ model the model's latent space size.

$$V_{interaction}(V_{seq_1}, V_{seq_2}, V_{FP_1}, V_{FP_2}) = \sum_{i=1}^{2} SeqEnc_i\left(Emb\left(Tokenize\left(V_{seq_i}\right)\right) + Pos_info\right) + \sum_{i=1}^{2} StrEnc_i(Emb\left(FP_indexation\left(V_{FP_i}\right)\right) + Pos_info) \quad \text{Eq (4.34)}$$

The resulting encoded feature vector of the designated interaction is then decoded via RSSSTr decoder $Dec()$ for more feature extraction, before being flattened and passed to a $Dense$ layer

with a $Sigmoid$ activation function in order to estimate the overall class of the interaction by applying Eq (4.35), with RSSSTr overall architecture being depicted in Figure 4.25.

$$\hat{Y}_{V_interaction} = Sigmoid(Dense(Flatten(Dec(V_{interaction})))) \quad \text{Eq (4.35)}$$

#### 4.4.2.1. Preprocessing phase

Before being processed by RSSSTr; a molecule's representation $Seq = (t_0, \dots, t_m)$ is padded, then tokenized via either a regular tokenization method (in the case of sequential representations) or via our $FP_indexation$ method (in the case of FP vectors). This newly generated vector representation $Seq' = (i_0, \dots, i_m, Pad_{m+1}, \dots, Pad_n) \in \mathbb{R}^n$ is then mapped into a latent space via an embedding layer as depicted in Eq (4.36), which generates the sequence's latent vector representation $V_{seq} = (v_0, \dots, v_n) \in \mathbb{R}^{n \times h_emb}$ with $V_i \in \mathbb{R}^{h\text{-}emb}$ being the $h$-dimensional vector representation of the $i^{th}$ token in the sequence, which is then injected with positional information as depicted in Eq (4.34) to help link each token's latent representation with its position in the sequence to grant self-attention mechanisms the ability to perceive the alignment of tokens more efficiently.

$$\begin{gathered} V_{seq} = Emb(Seq') \\ Emb(Seq') = Concat(Emb(i_0), \dots, Emb(i_m), Emb(Pad_{m+1}), \dots, Emb(Pad_n)) \end{gathered} \quad \text{Eq (4.36)}$$

#### 4.4.2.2. Encoding phase

After all four input vector representations of the molecules involved in an interaction finish being preprocessed; each of their vectors is encoded by one of RSSSTr four identical encoders, which are composed of several stacked identical encoder layers, followed by a normalization layer, as depicted in Eq (4.37).

$$Enc(V_{seq}) = Norm(Enc_Layer_3(Enc_Layer_2(Enc_Layer_1(V_{seq}) + V_{seq}) + V_{seq})) \quad \text{Eq (4.37)}$$

With $Enc()$ being one of the four encoders $SeqEnc1()$, $SeqEnc2()$, $StrEnc1()$, or $StrEnc2()$, $Norm()$ being a normalization layer, and $Enc_Layer_i()$ being an encoding layer. Our encoder's architecture is distinguishable from the original encoder's architecture by the additional residual connections (depicted as dashed red arrows in Figure. 4.25) implemented for the purpose of feeding the original input passed to the encoder to the input of each of its encoder layers (excluding the first encoder layer), with the intention of further enhancing the flow of gradient during back-propagation [435], while granting later encoder layers more insight regarding the original sequence processed by the encoder through enabling them to witness features lost during the encoding processes performed by previous layers. Note that the encoder layers implemented in our model adopt the Pre-LN approach [436, 437] where the normalization layer is positioned prior to the sub-layer, rather than the conventional Post-LN one [239] (normalization layer is positioned after the sub-layer), since the latter one poses higher risks of vanishing/exploding gradients (especially when dealing with large loss values) by preventing the model from passing the gradient back without skipping normalization layers [437], in contrast the former one that can skip normalization layers during back propagation.

#### 4.4.2.3. Decoding phase

The four encoded vector representations of the two involved molecules are then passed on to RSSSTr decoder that processes their summation as a single feature vector, which can be considered at this point as the interaction's encoded feature vector that incorporates both sequential and structural features of the molecules involved in it. The decoder's architecture is set to be identical to that of the encoders that is depicted in Eq (4.37), and knowing that RSSSTr is not performing a translation task; only padding masks are deployed within its decoder, since recommendation tasks require full access to the processed feature vectors, hence eliminating the need for causal masks. Note that the choice of such architecture for the decoder came after several experiments in which we tried to implement architectures that are similar to those of bert [438] and biobert [103], as well as our attempts to integrate two parallel cross-attention mechanisms where each one processes a molecule's feature vector (that was the sum of its encoded sequential and structural feature vectors) as Query vector, and the other molecule's feature vector as Key/Value vectors, which resulted in a way too complex model that failed to pass the training phase.

#### 4.4.2.4. Projection layer

Finally, an interaction's decoded feature vector $V'_{interaction} \in R^{n \times d_model}$ is flattened, then passed to a $Dense$ layer with a $Sigmoid$ activation function to estimate the interaction's overall class as depicted in Eq (4.35).

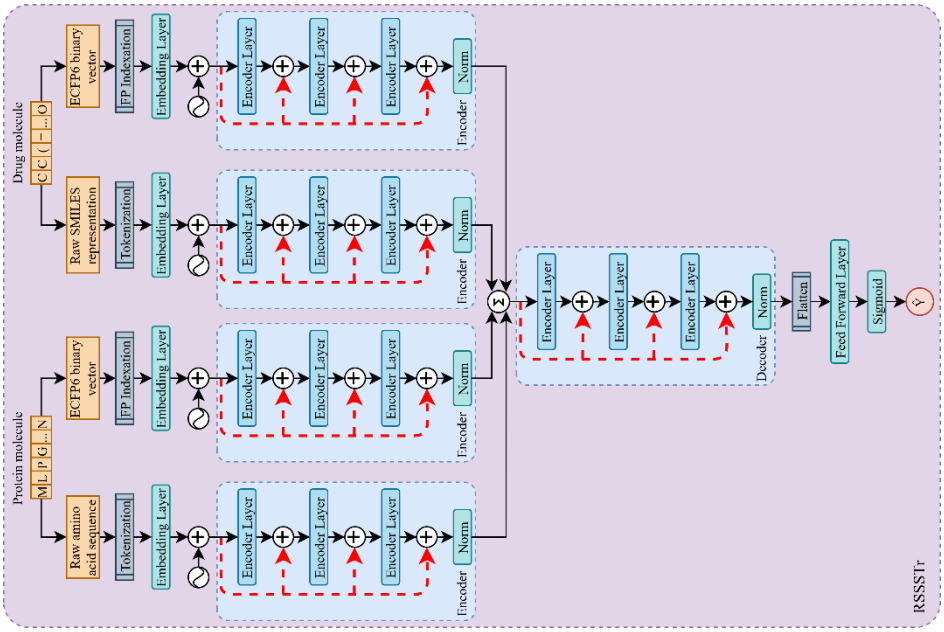



*Figure 4.25 The architecture of RSSSTr*

### 4.4.3. The architecture of TSSTr

Our translator model on the other hand is characterized by the same conventional methods and architecture implemented in transformer-based translator models, with the only difference being (1) TSSTr incorporation of the aforementioned residual connections that feed the original input passed to an encoder to each of its encoder layers, and (2) its adoption of an additional encoder

sub-architecture that grants it the ability to process the structural feature vector of the source molecule alongside its sequential one in order to grant it access to more latent features for more prominent translation outcomes as depicted in Figure 4.26.

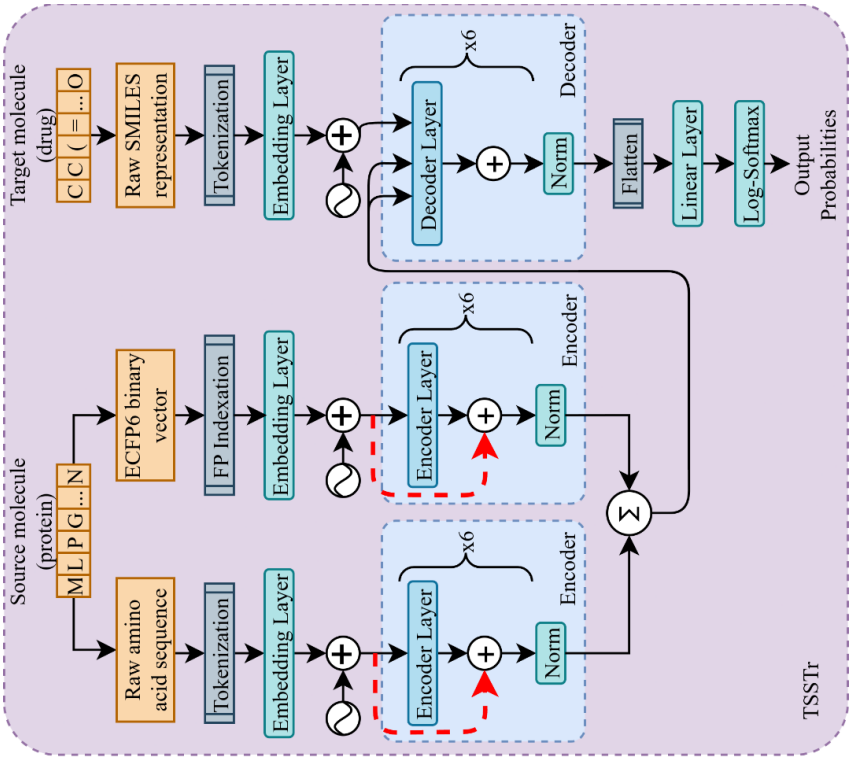



*Figure 4.26 The architecture of TSSTr*

### 4.4.3.1. Encoding phase

With architectures that are similar to those implemented in RSSSTr; TSSTr two encoders process both sequential and structural vector representations of the source molecule separately. The resulting encoded two vectors are then summed in order to form the source molecule's overall encoded feature vector that incorporates both its sequential and structural features, which is then passed to TSSTr decoder to be processed as Key/Value vectors in its multi-head cross attention mechanisms.

### 4.4.3.2. Decoding phase

TSSTr decoder on the other hand is identical to that of the original transformer-based translator, where it stacks several decoder layers followed by a normalization layer, with each decoder layer being composed of three stacked residual connections (adopt the Pre-LN approach) that engulf a masked multi-head self-attention (applies a padding mask combined with a causal mask), a multi-head cross-attention (processes the previous layer's output as its Query vector, and the sum of the two encoders' outputs as its Key/Value vectors), and a position-wise feed-forward layer as their sub-layers respectively [239]. Note that the residual connections that we implemented in the encoders of both RSSSTr and TSSTr and the decoder of RSSSTr are not implemented in TSSTr decoder, knowing that applying them would cause the target molecule's feature vector to

bypass the causal masks (applied in each decoder layer's masked-multi-head self-attention) set to prevent the decoder from processing future tokens and preserve the model's autoregressive nature. Furthermore, TSSTr does not process the target molecule's FP vector for several reasons, (1) FP vectors represent the overall structure of a molecule, hence cannot be generated token-wise like sequential vectors, (2) the structural features present in FP vectors are not aligned according to their presence in the respective molecule sequence, hence, causal masks cannot be effectively applied on them, and (3) we thought of generating local FP vectors for each of the sub-sequences that are generated token-wise during inference, but the majority of them would possess invalid SMILES sequences (in the case of drugs) that FP vectors cannot be generated for (AA sub-sequences can bypass such issue, but this paper focuses more drugs rather than protein sequences).

#### 4.4.3.3. Projection layer

Finally, the decoded feature vector is then flattened before passing through a Dense layer with a $Log\text{-}Softmax$ activation function that maps it to the same size as the vocabulary size of target molecules, indicating the probability of each token being next in the sequence.

### 4.4.4. Evaluation and analysis

#### 4.4.4.1. Datasets

For the purpose of evaluating RSSSTr; experiments were conducted on four datasets that incorporate the interactions of human protein sequences with other molecules, with two sets revolving around the DTIs of the human GPCR and CDK single-protein sequences and their interactions with drugs, which are extracted from the ChEMBL database [41] based on $IC50$ score values (same threshold values deployed earlier). The other two sets involved the direct and physical PPIs of human protein sequences (regardless of their family) and protein sequences belonging to either HIV or SARS-CoV-2 (COV) organisms that are extracted from the IntAct database [439]. these interactions are based on recorded MI-score (confidence) values, with interactions that are characterized by MI-score values that are greater than or equal to $0.5$ being considered as positive interactions (assigned positive classes), and vice versa. As for TSSTr evaluation process, and since it's obvious that we would only be interested in generating relevant molecules; both its training and inference tasks were limited to the positive samples cross all four aforementioned sets (negative samples are completely ignored), while being performed considering both translation directions (translate from human protein sequences to relevant virus protein sequences or drugs, and vice versa). Note that despite the fact that this paper primarily focuses on drug discovery; we purposely included the two target-target sets in the process of evaluating both RSSSTr and TSSTr so that we can demonstrate the robustness of our methods in extracting the structural features residing in binary FP vectors and the implications of doing so. All molecules present in all four datasets are $500$ tokens long or less, with general information regarding all four datasets being depicted in Table 4.18.

#### 4.4.4.2. Experimental setup

Focusing on RSSSTr, which is characterized by a $256$-sized latent-space that is set for all of its four embedding layers, with its four encoders alongside its decoder being composed of 3-

stacked encoder layers that are all preserving dimensionality at 256 via their self-attention mechanisms, while possessing position-wise feed-forward sub-layers that are all characterized by a hidden space size of 1024. TSSTr on the other hand has a higher complexity, with a 512 latent-space size set for all of its embedding layers and self-attention mechanisms, a total of 6-stacked encoder layers in each encoder, with a 2048 hidden size set for each encoder layer's position-wise feed-forward sub-layer, and a total of 6-stacked decoder layers that also have a latent-space size of 512 and a hidden size of 2048 set for each position-wise feed-forward sublayer. Both models possess a unified max sequence length of 500 tokens, while having the same number of attention heads that's 8, with all their dropout layers set to the value of 0.3, while their weights are initialized via the Xavier initialization method [225]. Note that the sequential vectors chosen to represent the studied individuals are AA sequences regarding protein individuals, and SMILES representations [440] regarding drug molecules, whereas the structural feature vectors representing both types of molecules are generated via $ECFP_6$ with a bit-string length of 500 (its dynamic length renders it able to fit to the max sequence length, thus providing more structural features to process while eliminating the need for padding masks). The choice of such algorithm was made after several experiments that involved several FP-generating algorithms, with dictionary-based ones (e.g., MACCS) failing to efficiently generate FP vectors for protein sequences (the limited size of their dictionaries renders most protein sequences with identical FP vectors, since all of them are composed of the same set of AA that are just aligned differently) and forcing models deploying them to depend solely on the sequential features of protein sequences, hence, the need to deploy a hash-method based FP algorithm.

*Table 4.18 Characteristics of the deployed datasets*

| Dataset Properties | GPCR | CDK | HIV | COV |
|---|---|---|---|---|
| Number of interactions | 1413 | 4641 | 361 | 2852 |
| Number of human protein sequences | 6 | 13 | 158 | 1205 |
| Number of drugs (or virus protein sequences) | 1014 | 2970 | 21 | 68 |
| Number of positive samples | 893 | 3092 | 150 | 867 |
| Number of negative samples | 520 | 1549 | 211 | 1985 |
| Interaction matrix sparsity | 76.78 % | 87.98 % | 89.12 % | 96.52 % |

#### 4.4.4.3. Training process

The training process of RSSSTr was performed for a total of 500 epochs on batches of 8 on all four datasets that were split to 70%, 10%, and 20% as a training, validation, and test sets respectively, with its performance being monitored via BCE loss function, and optimized via the Adam optimizer [408] that has a learning rate of $10^{-4}$, with the Cohen kappa score being chosen as the metric used to evaluate the model's performance on the validation sets in order to retrieve the top performing version of the model, which would be used during inference on the test sets and later in Section 4.4.4.5. Additionally, to better visualize the results of RSSSTr, we had it compared with four state-of-the-art RS, with one being a problem-specific approach that is RSCRDL [441], while the other three being general-purpose ones, encompassing NeuMF [362],

ConvNCF [363], and GLocal-K [387], with all four models being selected based on their remarkable results across Cytokine and CDK, Pinterest, Gowalla, and MovieLens 1M datasets respectively. The training process of TSSTr on the other hand was also performed for a total of 500 epochs on batches of 6 on all four datasets that were split to 90%, and 10% as a training and validation sets respectively (test set is ignored), with its performance being monitored via the CE loss function that's characterized by a label smoothing value of 0.1, and optimized via the Adam optimizer that has a learning rate of $10^{-4}$ and an epsilon value of $10^{-9}$, with the METEOR score [417] being chosen as the metric used to evaluate the model's performance on the validation sets in order to retrieve the top performing version of the model, which would be exploited later in Section 4.4.4.5. To better visualize the results of TSSTr; we had it compared with a state-of-the-art Pre-LN transformer model [437, 239] (since it was proven to be more efficient than the Post-LN one). Note that all five state-of-the-art approaches are implemented with the set of parameters stated by their authors. Furthermore, while testing both our approaches; we intentionally did not incorporate any form of performance-enhancing methods (e.g., self-supervision, pretraining, hyperparameter tuning, etc.) since our main intention in this paper is displaying the raw potentials of our proposed approaches without any interference from such methods.

#### 4.4.4.4. Performance assessment

### A. Recommender systems results

Focusing on the five RSs, and starting with their training and validation BCE loss values across all four datasets that are depicted in Figures 4.27 and 4.28 (GLocal-K is excluded due to its distinct training paradigm), where it's noticed that RSSSTr, NeuMF, and ConvNCF are exhibiting signs either overfitting or overconfidence, as their training loss values are rapidly converging and stabilizing at optimal values (with NeuMF being the fastest model to converge, followed by ConvNCF, then RSSSTr that converges later at slightly higher loss values) despite outputting exploding validation loss values (both NeuMF and ConvNCF validation loss values kept increasing over the course of training regarding all sets but HIV, whereas RSSSTr showed similar behaviors but only regarding CDK and COV sets).

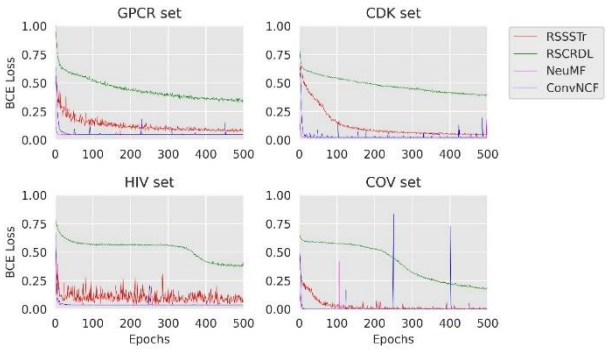



*Figure 4.27 The BCE loss values of RS w.r.t the number of epochs across all four training sets (lower is better)*

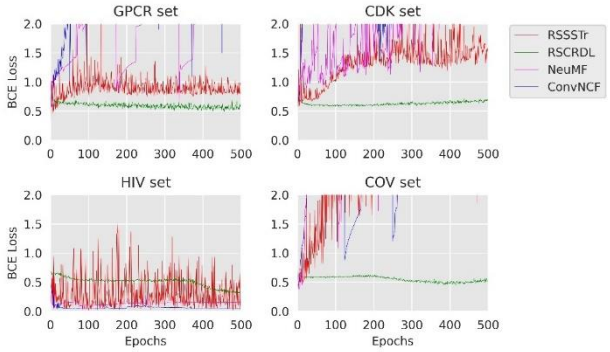



*Figure 4.28 The BCE loss values of RS w.r.t the number of epochs across all four validation sets (lower is better)*

The aforementioned assumption regarding three models being overconfident is further supported by the deployed metrics' results (balanced accuracy, precision, recall, and F1 score)

regarding all training and validation sets that are depicted in Figures 4.29 and 4.30 (including GLocal-K), with RSSSTr, NeuMF, and ConvNCF results concerning the deployed metrics being both robust and significant across all training and validation sets (showing no signs of poor generalization), thereby eliminating the possibility of them suffering from overfitting, and further explaining their results being more significant than those of RSCRDL, which showed no signs of overconfidence in said metrics, with RSSSTr being able to outperform RSCRDL across all four validation sets, while both NeuMF and ConvNCF are able to do so but only regarding the two target-target sets. RSCRDL on the other hand displayed the most consistent loss values across all four sets, despite its significantly higher training loss values that imply to the possibility of it suffering from underfitting, which is reinforced by RSCRDL being temporarily trapped in local minima while processing the two target-target sets, highlighting the limitations of RNNs at extracting latent features across long sequences (hence the higher loss values), and the inefficiency of AA-based representations at incorporating structural features of molecules (thus explaining the gap between its target-drug and target-target sets' metric results). Concluding with GLocal-K which showed the most controversial results (with its results concerning both GPCR and HIV sets being both competitive and robust, in contrast to its results regarding both CDK and COV sets that are less significant) that strongly suggest its vulnerability to large-sized sparse matrices.

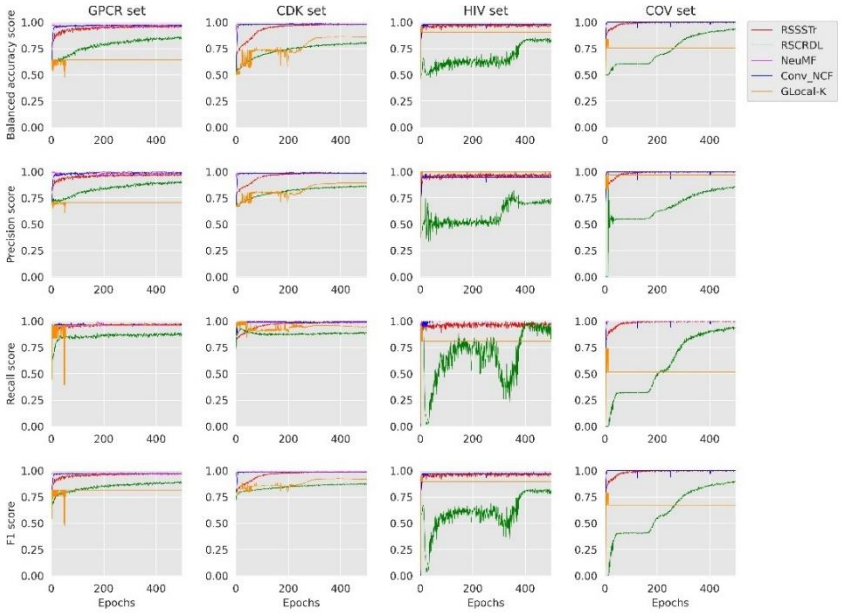



*Figure 4.29 Metrics' results w.r.t the number of epochs for all five RS across all four training sets (higher is better)*

Our speculations regarding all five RS are further strengthened by the Cohen kappa score results (Figure 4.31) concerning all four validation sets, with RSSSTr emerging as the top-performing model across all four validation sets, demonstrating the efficiency of its transformer-based encoder-decoder architecture and the residual connections it implements, alongside the robustness of the structural feature vectors it leverages that are generated via our $FP_indexation$ method (since RSSSTr was able to outperform RSCRDL, which depends solely on sequential representations, while avoided being trapped in local minima in both target-target sets). Following RSSSTr in both target-drug sets is RSCRDL, despite its underperformance in the two target-target

sets, further reinforcing our assumptions regarding the inadequacy of AA-based representations at capturing the structural features of molecules. In contrast, NeuMF and ConvNCF displayed less significant results regarding both target-drug sets, implying to the limitations of their general-purpose techniques at handling poor-profiled individuals (69.6% and 68.2% of the drugs present in GPCR and CDK sets respectively had only one recorded interaction). Such assumption is further reinforced by their improved performances regarding both target-target sets, where most individuals possess multiple interactions. GLocal-K on the other hand performed poorly all but HIV set, highlighting the inefficiency of kernel-based methods at processing large and sparse matrices. Lastly, by analyzing the inference results of all five models' across all four test sets that are depicted in Tables 4.19 and 4.20; we can conclude that RSSSTr is the most efficient model, with a significant gap noticeable between its performance and those of the other RS setting as a solid proof regarding the potentials residing in the techniques it adopts (the combination of sequential and structural molecular representations leveraged via transformer architectures that are supported by additional residual connections, paired with our FP indexation method that helps extract the structural features of molecules from binary FP vectors).

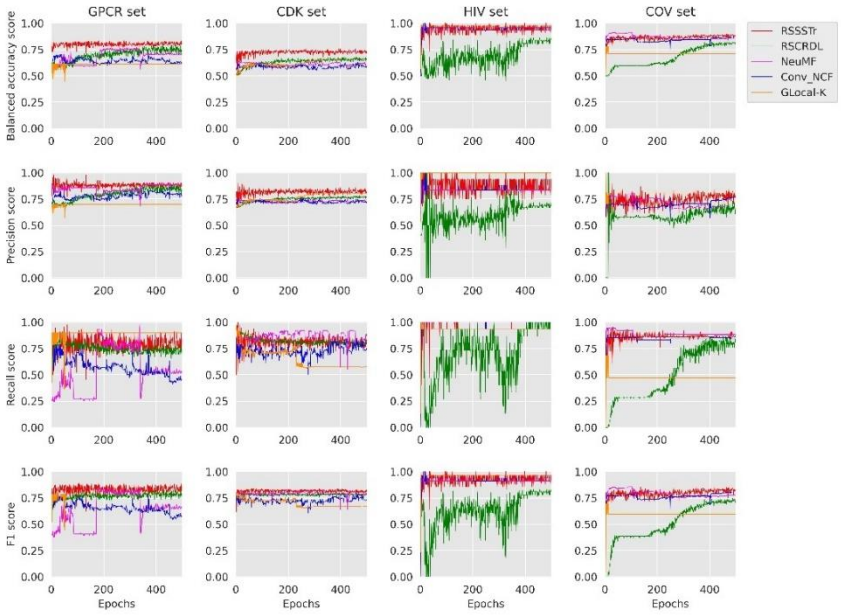



*Figure 4.30 Metrics' results w.r.t the number of epochs for all five RS across all four validation sets (higher is better)*

*Table 4.19 Metrics results' regarding all five RS concerning the two target-drug test sets*

| | GPCR test set | | | | | CDK test set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RSSSTr | RSCRDL | NeuMF | ConvNCF | GLocal-K | RSSSTr | RSCRDL | NeuMF | ConvNCF | GLocal-K |
| **Cohen kappa** | **0.675** | 0.544 | 0.587 | 0.567 | 0.322 | **0.504** | 0.317 | 0.237 | 0.182 | 0.227 |
| **Balanced accuracy** | **0.825** | 0.771 | 0.787 | 0.784 | 0.641 | **0.756** | 0.663 | 0.608 | 0.587 | 0.629 |
| **F1** | **0.891** | 0.834 | 0.856 | 0.839 | 0.813 | **0.829** | 0.760 | 0.786 | 0.753 | 0.657 |
| **Precision** | **0.848** | 0.830 | 0.831 | 0.842 | 0.708 | **0.844** | 0.781 | 0.728 | 0.719 | 0.787 |
| **Recall** | 0.938 | 0.839 | 0.882 | 0.837 | **0.955** | 0.814 | 0.739 | **0.854** | 0.791 | 0.563 |

*Table 4.20 Metrics results' regarding all five RS concerning the two target-target test sets*

| | HIV test set | | | | | COV test set | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RSSSTr | RSCRDL | NeuMF | ConvNCF | GLocal-K | RSSSTr | RSCRDL | NeuMF | ConvNCF | GLocal-K |
| **Cohen kappa** | 0.829 | 0.405 | 0.805 | 0.774 | **0.887** | **0.819** | 0.566 | 0.796 | 0.744 | 0.524 |
| **Balanced accuracy** | 0.914 | 0.714 | 0.912 | 0.890 | **0.948** | 0.913 | 0.793 | **0.914** | 0.888 | 0.756 |
| **F1** | 0.900 | 0.694 | 0.892 | 0.871 | **0.935** | **0.874** | 0.705 | 0.862 | 0.828 | 0.663 |
| **Precision** | 0.900 | 0.595 | 0.829 | 0.844 | **0.906** | **0.860** | 0.665 | 0.807 | 0.772 | 0.694 |
| **Recall** | 0.900 | 0.833 | **0.967** | 0.900 | **0.967** | 0.890 | 0.750 | **0.924** | 0.891 | 0.634 |

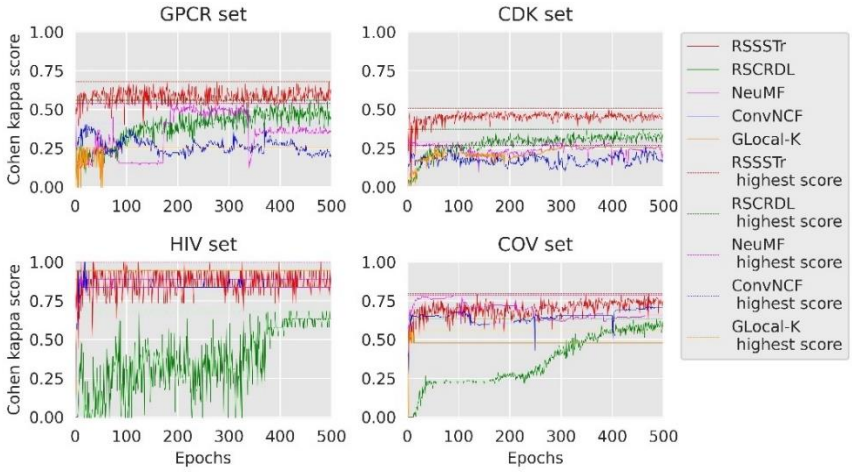



*Figure 4.31 Cohen kappa score results w.r.t the number of epochs for all five RS across all validation sets (higher is better)*

## B. Translator models results

Moving on the translator models' results, and starting with their training and validation loss values regarding all eight translation tasks that are depicted in Figure 4.32, where it's noticed that TSSTr is significantly outperforming the Pre-LN transformer in all four target-target translation tasks by converging rapidly while stabilizing at more optimal loss values, indicating the robustness of the residual connections implemented in its encoders that are granting it the ability to avoid vanishing/exploding gradients more efficiently, as well as its ability to overcome the lack of structural information that proteins' AA sequences fail to provide by exploiting proteins' structural features generated via the incorporated FP indexation method (hence explaining the Pre-LN transformer's significantly higher training and validation loss values regarding both target-target sets' translation tasks). The aforementioned statements are further strengthened by the loss values acquired from the four target-drug related translation tasks, where the availability of sequential representations that are more efficient at incorporating the structural features of molecules (i.e., SMILES) do significantly enhance the overall performance of the Pre-LN transformer to the point of rendering it able to perform on the same level as TSSTr (despite converging at slower rates compared to TSSTr), with both models showing less significant loss values regarding the two P2D translation tasks, which we assume to be caused by the large number of less-frequent drugs that

are hindering the two models' learning paradigms, with TSSTr being more affected than the Pre-LN transformer due to the elevated complexity of its architecture.

Concluding with the two models' METEOR score values regarding all eight validation sets' translation tasks that are depicted in Figure 4.33, from which it's evident that TSSTr is more efficient at intermolecular translation tasks (converges faster while achieving more robust results in all eight translation tasks) through exploiting both sequential and structural features of molecules, which can be crucial especially in tasks that involve protein sequences, since their sequential representations cannot efficiently incorporate their structural features (as shown in all four target-target based translation tasks).

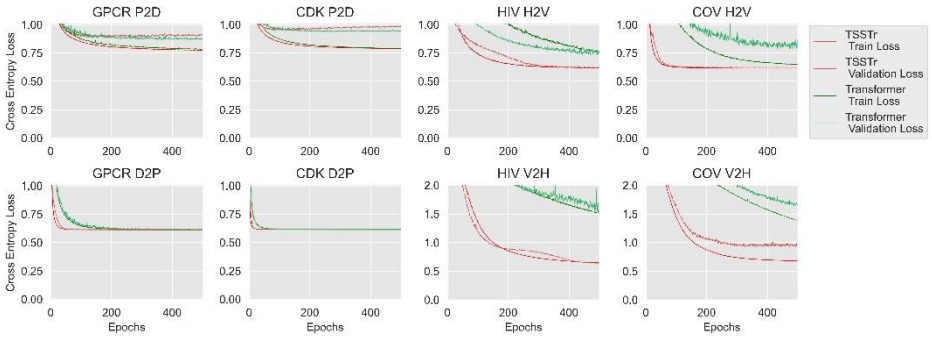



*Figure 4.32 CE loss values of translator models w.r.t the number of epochs regarding all eight translation tasks across all training and validation sets (lower is better), with P2D, D2P, H2V, V2H being the protein-to-drug, drug-to-protein, human-to-virus protein, and virus-to-human protein translation tasks respectively*

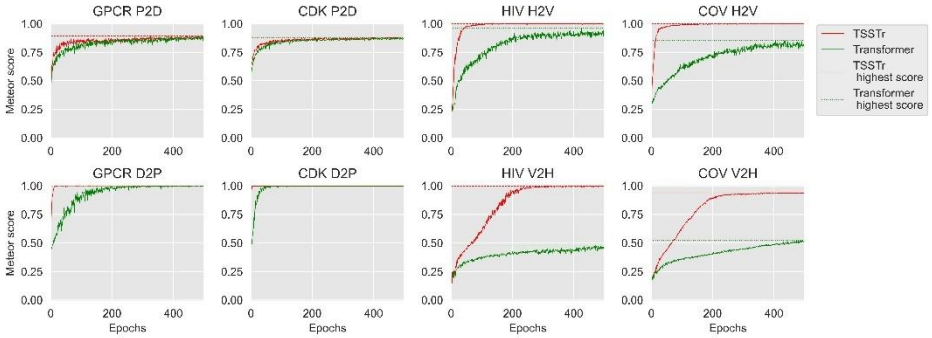



*Figure 4.33 Meteor score values of translator models w.r.t the number of epochs regarding all eight translation tasks across all four validation sets (higher is better), with P2D, D2P, H2V, V2H being the protein-to-drug, drug-to-protein, human-to-virus protein, and virusto- human protein translation tasks respectively*

#### 4.4.4.5. The application of RSSSTr and TSSTr in drug repurposing and *de novo* drug design strategies involving both regular and orphan protein sequences

Now that both RSSSTr and TSSTr are proven to possess the knowledge needed for them to perceive the behaviors of molecules, and knowing that both models are able to process even unobserved molecules; both approaches are exploited in order to (1) perform matrix completion task through the deduction of unobserved interactions via RSSSTr, and (2) perform matrix expansion task by generating novel relevant drug molecules across the two target-drug sets.

**A. Matrix completion task**

Starting with the matrix completion task that is accomplished in a manner similar to that of RSCRDL, where unobserved candidate positive interactions are estimated via RSSSTr by processing sparse interaction matrices, with said candidates being filtered based on the intermolecular similarities measured between residents of the interaction matrices, which are performed via biological similarity measuring methods as depicted in Eq (4.38) (note that this equation is specifically meant to extract candidate positive interactions from target-drug sets, and in order to apply it on target-target sets; the $3^{rd}$ clause of the equation is replaced with a clause that is identical to the $2^{nd}$ one with the exception of referring to the other set of proteins present in the set), where an unobserved interaction between protein $p_i$ and drug $d_j$ is considered as a candidate positive one only if (1) RSSSTr estimated it as a positive interaction (the probability of it being positive is greater than or equal to the threshold value $th_{RS}$ that is set to $50\%$), (2) RSSSTr has correctly estimated an already existing valid positive interaction between a neighbor (or more) of either molecules involved in the candidate positive interaction and the other molecule (a neighboring protein $n_k$ of $p_i$ has a valid positive interaction with $d_j$ that RSSSTr has correctly estimated, or a neighboring drug $n_l$ of $d_j$ has a valid positive interaction with $p_i$ that RSSSTr has correctly estimated), with protein sequences being considered as neighbors only if the E-value score between their AA sequences is less than or equal to the threshold value $th_{evalue}$ that is set to $10^{-6}$ and the Tc value $th_{Tc_P}$ between their ECFP$_6$ vectors (that are characterized by bit-string lengths of $1024$) is greater than or equal to $0.8$, while drugs being considered as neighbors only if the Tc value $th_{Tc_D}$ between their respective ECFP$_6$ vectors is greater than or equal to $0.42$ (note that in order to retrieve more confident results, all that's needed is just to increase the threshold values $th_{RS}$, $th_{Tc_P}$, $th_{Tc_D}$ and decrease the value of $th_{evalue}$).

$$\begin{gathered}\exists Candidate_positive_interaction(p_i, d_j) \;\; if: \;\; RSSSTr(p_i, d_j) \geq th_{RS} \\ And \\ \exists n_k \in P: \;\; Evalue(p_i, n_k) \leq th_{evalue} \;\; And \;\; Tc\big(ECFP_6(p_i), ECFP_6(n_k)\big) \geq th_{Tc_P} \\ \text{And } \exists valid_positive_interaction(n_k, d_j): \;\; RSSSTr(n_k, d_j) \geq th_{RS} \\ Or \\ \exists n_l \in D: \;\; Tc\left(ECFP_6(d_j), ECFP_6(n_l)\right) \geq th_{Tc_D} \\ And \; \exists valid_positive_interaction(p_i, n_l): \;\; RSSSTr(p_i, n_l) \geq th_{RS}\end{gathered} \qquad \text{Eq (4.38)}$$

The application Eq (4.38) on the four datasets resulted in a total of $703$ and $12779$ protein-based candidates regarding GPCR and CDK sets respectively, $1164$ drug-based candidates regarding the CDK set, $2$ human protein-based candidates regarding the COV set, and $24$ and $12$ virus protein-based candidates regarding COV and HIV sets respectively (note that we were not able to deduce any drug-based or human protein-based candidate positive interactions for GPCR and HIV sets respectively). Showcasing some of the candidate positive interactions that were deduced for the two target-drug sets, and starting with the one that involves 'P32249' (G-protein coupled receptor 183); a member of the GPCR protein family located in a cell's membrane where it contributes in cells migration and communications [442, 443], involved in several GPCR-related signaling pathways [430], associated with 'DFNB24' and 'GEFS+ Type 6' diseases [430], while being considered as a favorable prognostic marker in cervical cancer [433]. Based on the $2$ neighbors that this protein has in the GPCR set; we were able to deduce a total of $328$ candidate

positive interactions for it, with 272 of them involving drugs with no violations to the rule-of-5 such as 'CHEMBL3718309', with RSSSTr estimating the likelihood of their interaction to be 100% positive, which is based on the candidate protein's neighbor 'Q9NQS5' (with which 'P32249' share an E-value score of $9.72 \times 10^{-9}$ alongside a Tc similarity value of 0.802) that is characterized by an $IC50$ value of $0.01nM$ for its interaction with the candidate drug. Additional candidate interactions are displayed in Table 4.21, where the top 5 most confident protein-based candidate positive interactions that involve drugs with no rule-of-5 violations across both CDK and GPCR sets.

*Table 4.21 Top 5 most confident protein-based candidate positive interactions involve drugs with no rule-of-5 violations regarding both target-drug sets*

| Set | Candidate protein ID | Candidate protein neighbors | E-value scores between candidate protein and its neighbors | Tc scores between candidate protein and its neighbors | Candidate drug ID | RSSSTr |
|---|---|---|---|---|---|---|
| CDK | Q00526 | P24941 | $2.18\times10^{-175}$ | 0.885 | CHEMBL3298309 | 100% |
| | Q00534 | P06493, P24941, P50750, P11802, P50613, Q00526 | $4.56\times10^{-91}$, $4\times10^{-95}$, $7.09\times10^{-62}$, $2.79\times10^{-154}$, $1.61\times10^{-63}$, $1.26\times10^{-96}$ | 0.830, 0.851, 0.806, 0.864, 0.835, 0.856 | CHEMBL4282338 | 99.9 % |
| | Q92772 | P06493, P24941, P50750, P11802, P50613, Q00526 | $1.29\times10^{-74}$, $5.6\times10^{-77}$, $6.3\times10^{-60}$, $8.44\times10^{-49}$, $2.55\times10^{-61}$, $4.65\times10^{-76}$ | 0.829, 0.849, 0.829, 0.824, 0.816, 0.839 | CHEMBL4282338 | 99.9 % |
| | Q00535 | P06493, P24941, P50750, P11802, P50613, Q00526 | $5\times10^{-120}$, $2.45\times10^{-128}$, $2.63\times10^{-72}$, $1.35\times10^{-76}$, $5.57\times10^{-91}$, $3.22\times10^{-133}$ | 0.878, 0.880, 0.814, 0.823, 0.834, 0.850 | CHEMBL4442620 | 99.9 % |
| | Q5MAI5 | P06493, P24941, P11802, Q00535, P50613 | $1.19\times10^{-72}$, $1.16\times10^{-73}$, $1.11\times10^{-52}$, $4.82\times10^{-70}$, $8.02\times10^{-67}$ | 0.817, 0.843, 0.832, 0.813, 0.828 | CHEMBL2103840 | 99.8 % |
| GPCR | P32249 | Q9NQS5 | $9.72\times10^{-09}$ | 0.802 | CHEMBL3718521 | 100% |
| | Q9NQS5 | P46095 | $1.62\times10^{-08}$ | 0.867 | CHEMBL3982858 | 100% |
| | P32249 | Q9NQS5 | $9.72\times10_{-09}$ | 0.802 | CHEMBL4060589 | 100% |
| | Q9HC97 | P32249 | $9.01\times10^{-29}$ | 0.849 | CHEMBL3560529 | 99.9 % |
| | Q9Y2T6 | Q9HC97 | $1.17\times10^{-39}$ | 0.822 | CHEMBL4876009 | 67.1 % |

Moving on to the CDK set's drug-based candidate positive interactions that revolve around 10 protein sequences and a total of 727 drugs (with 635, 80, and 12 candidate drugs possessing 0,

1, and 2 violations to the rule-of-5 respectively), with 733 candidate positive interaction being deduced based on drugs that are also characterized by no violations to the rule-of-5, such as the candidate that involves the CDK protein family member 'P11802' (Cyclin-dependent kinase 4), an unfavorable prognostic marker in both liver and renal cancers [433] that is mainly located in a cell's cytoplasm, nucleus, or nucleus membrane [434] where it contributes in cell's cycle [430] , involved in the 'RB1 dysfunction' and 'Cellular Senescence' pathways [430], alongside being associated with several Melanoma diseases [430]. Among the 657 drugs that 'P11802' has valid positive interactions with; 4 of the them (i.e., 'CHEMBL3644032', 'CHEMBL4442620', 'CHEMBL3644034', and 'CHEMBL3644033') are among the 18 neighbors that the candidate drug 'CHEMBL3644022' has in the CDK set (with which it share the Tc similarity values of 0.89, 0.72, 0.547, and 0.737 respectively), and with these 4 neighboring drugs possessing the $IC50$ values of 20, 5, 24, and $24nM$ for their interactions with the candidate protein sequence respectively; RSSSTr estimation that reached the value 99.63% regarding the candidate interaction being positive is further strengthened. Additional CDK set's drug-based candidate positive interactions are depicted in Table 4.22, where the top 5 most confident candidates that involve drugs with no rule-of-5 violations are depicted, which are deduced based on their neighboring drugs that are also in possession of no violations to the rule-of-5.

Focusing on the candidate positive interactions that are extracted from the two target-target sets, and specifically one of the 2 human protein-based candidates that are deduced from the COV set, where the human protein 'P27361' (Mitogen-activated kinase 3) that is known for its contribution in the regulation of numerous cellular processes while being part of both 'Prolactin Signaling' and 'TLR3 Cascade' pathways [430] is assumed to positively interact with 'P0DTC9', which is known for its fundamental roles in viral RNA transcription and viral replication [434]. This candidate interaction is based on the latter protein's valid positive interaction (characterized by a MI-score value of 0.53) with 'P19784' (Casein kinase II subunit alpha'), which is one of the two neighbors that 'P27361' has in the COV set, with their E-value score settling at the value of $6.14 \times 10^{-31}$ while their Tc similarity value is settling at 0.81, thus further supporting RSSSTr estimation concerning 'P27361' and 'P0DTC9' interaction being 98.02% positive. The last candidate we will be showcasing in this section is concerned with human protein 'O14734' (Acyl-coenzyme A thioesterase 8), a possible mediator in the Nef-induced down-regulation of CD4 [444] that we suspect to interact positively with the virus protein 'Q90VU7' as stated by RSSSTr (with a probability of 99.81%). Such assumption is based on the latter protein's neighbor 'P04601' that shares a valid positive interaction with the aforementioned human protein sequence (with a MI-score value of 0.71) [439], alongside the fact that both virus proteins possess significant E-value and Tc score values (settling at $3.67 \times 10^{-157}$ and 0.982 respectively), while sharing commune characteristics that range from their essentiality in viral replication [445], their ability to increase viral infectivity and avoid host defense [446], and their ability to down-regulate several infected CD4+ T-lymphocytes related molecules [447, 434], hence, making RSSSTr estimations more plausible, with Table 4.23 depicting COV and HIV sets top 5 most confident virus protein-based candidate positive interactions.

*Table 4.22 Top 5 most confident drug-based candidate positive interactions involving drugs with no rule-of-5 violations, deduced based on neighboring drugs with no rule-of-5 violations regarding CDK set*

| Set | Candidate protein ID | Candidate drug ID | Candidate drug neighbors | Tc scores between candidate drug and its neighbors | RSSSTr |
|---|---|---|---|---|---|
| CDK | P06493 | CHEMBL309573 | CHEMBL72548, CHEMBL1241849, CHEMBL4584344, CHEMBL73243 | 0.798, 0.44, 0.467, 0.894 | 100 % |
| | | CHEMBL484571 | CHEMBL72548, CHEMBL4584344, CHEMBL73243 | 0.656, 0.515, 0.524 | 100 % |
| | P50750 | CHEMBL3648095 | CHEMBL4741442 | 0.595 | 100 % |
| | Q00526 | CHEMBL3644032 | CHEMBL4442620 | 0.779 | 100 % |
| | P11802 | CHEMBL3648110 | CHEMBL4741442 | 0.55 | 100 % |

*Table 4.23 Top 5 most confident virus-protein-based candidate positive interactions across both target-target sets*

| Set | Candidate virus protein ID | Candidate virus protein neighbors | E-value scores between candidate virus protein and its neighbors | Tc scores between candidate virus protein and its neighbors | Candidate human protein ID | RSSSTr |
|---|---|---|---|---|---|---|
| COV | P0C6X7-PRO_0000037309 | P0DTD1-PRO_0000449619 | $1.69\times10^{-113}$ | 0.907 | P23396 | 100 % |
| | P0C6X7-PRO_0000037309 | P0DTD1-PRO_0000449619 | $1.69\times10^{-113}$ | 0.907 | P08865 | 100 % |
| | P59637 | P0DTC4 | $3.3\times10^{-46}$ | 0.953 | O75360 | 100 % |
| | P0C6X7-PRO_0000037309 | P0DTD1-PRO_0000449619 | $1.69\times10^{-113}$ | 0.907 | Q14257 | 100 % |
| | K9N7C7-PRO_0000422444 | P0DTD1-PRO_0000449624 | $5.48\times10^{-52}$ | 0.81 | Q9P0I2 | 99.99% |
| HIV | P04608 | P04326 | $3.64\times10^{-53}$ | 0.969 | P53999 | 100 % |
| | P04585-PRO_0000042447 | Q76353 | 0.0 | 0.991 | P31946 | 99.86% |
| | Q90VU7 | P04601 | $3.67\times10^{-157}$ | 0.982 | O14734 | 99.81% |
| | Q76353 | P04585-PRO_0000042447 | 0.0 | 0.991 | Q12824 | 99.74% |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Q72500 | P12520 | $3.67\times10^{-70}$ | 0.983 | Q9H4A6 | 99.67% |

## B. Matrix expansion task

Moving on to the main focus of this section that is the cooperation of RSSSTr and TSSTr to accomplish matrix expansion task (i.e., fully automated drug discovery task), that is simply the concatenation of a generative sub-task that is monitored via a recommendation one, through which both models enable the incorporation of novel relevant drug candidates into interaction matrices (hence the name matrix expansion), as depicted in Figure 4.34.

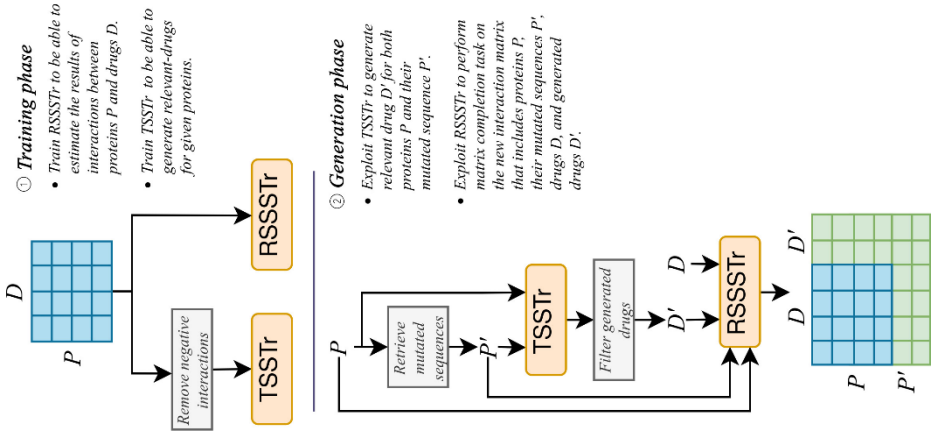



*Figure 4.34 Matrix expansion task work flow*

Focusing on the generative sub-task via which we introduce novel individuals to interaction matrices, and to ensure the provision of real-world exploitable results; the new protein sequences P′ that are integrated in an interaction matrix as new users (supposedly generated via TSSTr by processing an interaction matrix's drug residents D) are substituted by the mutated variants of those already residing in the same interaction matrix (retrieved from Uniprot database [37]) due to the fact that (1) there is no point in trying to process imaginary protein sequences that a studied organism cannot encode, (2) trying to incorporate protein sequences from other protein families would be less practical (different functionalities and locations), since RSSSTr only focuses on one protein family during a training process, thereby it wouldn't be able to generalize its knowledge efficiently, (3) ensuring that new targets possess at least one neighbor that has already been processed by RSSSTr (ensuring the ability of RSSSTr to generalize its results among homogeneous individuals), and (4) providing the opportunity to further study and understand the effects of mutations on the behaviors of protein sequences. Note that since we are not interested in generating protein sequences via TSSTr; both target-target sets are ignored in this section.

Drug generation sub-task on the other hand is performed via TSSTr, knowing that transformer architectures had already proven their potentials at generating said molecules by successfully learning their chemical properties through processing their sequential representations [448]. Moreover, to further prevent the generation of drugs that are either invalid (a drug that cannot physically exist), redundant (already exist), or harmful to use (e.g., possess high toxicity); generated drugs are further filtered out based on their properties regarding eight molecular

descriptors that most approved drugs abide to, including molecular weight (MolWt), count of H-bond donors (HBD), count of H-bond acceptors (HBA), molecular logarithm of the octanol-water partition coefficient (MolLogP) [334], rotatable bonds count (RotB), topological polar surface area (TPSA) [449], the count of sp3 hybridized carbons over the total carbon count (FSP3) [450], and the count of aromatic rings (#ArRNG) [451]. Consequently, matrix expansion task can be considered as the fully-automated drug discovery task listed in Eq (4.39), where a generated drug $drug_{new} \in D'$ can only be considered as a novel candidate one only if it was generated via TSSTr by translating a protein sequence $p_i$ (either a normal or a mutated sequence), while being valid, nonexistent (we only checked in ChEMBL database [41]), and abiding to all eight molecular descriptors threshold values set according to the $90^{th}$ and $10^{th}$ percentiles of Shultz study regarding approved new chemical entities (NCEs) between the years of 1998 and 2017 [452]. Note that during the drug generation sub-task, the exploitation of TSSTr in its inference state would result in the generation of one drug per unique target sequence, and to render it capable of generating larger numbers drugs for the same finite set of protein sequences (would still be limited by its own cognitive abilities though); TSSTr is deployed with and without dropout values (when deployed, dropout values are set to 0.3 for all dropout layers present in the TSSTr).

$$
\begin{aligned}
is_valid_novel_drug(drug_{new})\ if: \quad & drug_{new} = TSSTr(p_i) \quad for \quad p_i \in \{P \cup P'\} \\
And \quad & is_valid(drug_{new}) = True \\
And \quad & exist_in_ChEMBL(drug_{new}) = False \\
And \quad & 213 \leq MolWt(Drug_{new}) \leq 571 \\
And \quad & 0 \leq NumHDonors(Drug_new) \leq 4 \\
And \quad & 2 \leq NumHAcceptors(Drug_new) \leq 10 \\
And \quad & -0.1 \leq MolLogP(Drug_new) \leq 4.9 \\
And \quad & 2 \leq NumRotatableBonds(Drug_new) \leq 11 \\
And \quad & 33.9 \leq TPSA(Drug_new) \leq 144.1 \\
And \quad & 0.17 \leq FSP(Drug_new) \leq 0.75 \\
And \quad & 0 \leq NumAromaticRings(Drug_new) \leq 4
\end{aligned}
\qquad \text{Eq (4.39)}
$$

Starting with the drug-generative sub-task concerned with the GPCR set, where TSSTr is exploited for a total of 11 epochs (1 with no dropout values, and 10 with all dropout values set to 0.3) to process GPCR set's 6 protein sequences P and their 2162 mutated sequences P′. This process resulted in the generation of a total of 23848 drugs, which were reduced to a total of 334 (resemble 1.4% of the total generated drugs, with only 1 drug being generated by the non-dropout-dependent TSSTr while the rest 333 were generated by the dropout-dependent TSSTr) after the application of Eq (4.39) (while converting to canonical SMILES if necessary), with 229, 74, 26, and 5 being characterized by having 0, 1, 2, and 3 violations to the eight molecular descriptors listed in Eq (4.39) respectively. On the other hand, the application of the same procedure on the CDK set, where all 13 protein sequences P alongside their 5852 mutated sequences P′ are processed via TSSTr resulted in the generation of total of 64515 drugs, which were reduced to a total of 1514 valid drugs (resemble 2.35% of the total generated drugs, with 7 drugs being generated by the non-dropout-dependent TSSTr while 1507 were generated by the dropout-dependent TSSTr) after the application of Eq (4.39), with 1133, 312, 62, and 7 being characterized by 0, 1, 2, and 3 violations to the eight molecular descriptors listed in Eq (4.39)

respectively. General statistics for our newly generated drugs regarding all eight molecular descriptors listed in Eq (4.39) can be witnessed in Table 4.24 for both GPCR and CDK sets.

*Table 4.24 Statistics of the newly generated valid drugs across both target-drug sets regarding five molecular descriptors listed in Eq (4.39), with values in parentheses referencing the statistics of FDA approved NCEs from 1998 to 2017*

| | | NumHDonors | NumHAcceptors | MolWt | MolLogP | NumRotatableBonds | NumAromaticRings | TPSA | FSP | num_ro5_violations |
|---|---|---|---|---|---|---|---|---|---|---|
| GPCR | 90th percentile | 2 (4) | 8 (10.2) | 492.438 (571.0) | 4.597 | 6 (11) | 3 (4) | 108.696 (144.1) | 0.545 (0.75) | 1 |
| | Median | 1 (2) | 6 (6) | 422.887 (383.4) | 3.317 | 4 (6) | 2 (2) | 71.625 (77.2) | 0.25 (0.4) | 0 |
| | Mean±std | 0.772±0.784 (2.0) | 6.213±1.794 (6.4) | 423.004±58.956 (400.0) | 3.333±0.978 | 4.452±1.354 (6.5) | 2.162±0.869 (2.0) | 73.94±27.321 (88.1) | 0.284±0.168 (0.43) | 0.422±0.701 |
| | 10th percentile | 0 (0) | 4 (2) | 351.464 (213.0) | 2.225 | 3 (2.0) | 1 (0) | 40.62 (33.9) | 0.054 (0.17) | 0 |
| CDK | 90th percentile | 4 (4) | 9 (10.2) | 502.623 (571.0) | 4.97 | 8 (11) | 4 (4) | 113.65 (144.1) | 0.519 (0.75) | 1 |
| | Median | 3 (2) | 7 (6) | 414.502 (383.4) | 3.547 | 6 (6) | 3 (2) | 87.215 (77.2) | 0.429 (0.4) | 0 |
| | Mean±std | 2.641±1.274 (2.0) | 7.423±1.467 (6.4) | 418.019±60.316 (400) | 3.551±1.041 | 6.003±2.007 (6.5) | 2.861±0.831 (2.0) | 88.896±19.032 (88.1) | 0.403±0.109 (0.43) | 0.302±0.566 |
| | 10th percentile | 1 (0) | 6 (2) | 348.769 (213) | 2.269 | 4 (2) | 2 (0) | 66.28 (33.9) | 0.25 (0.17) | 0 |

Concluding matrix expansion task with the exploitation of RSSSTr to perform matrix completion sub-task on the expanded interaction matrices, which incorporate both protein sequences P and their mutated variants P′ as users, alongside drugs D and novel drugs D′ as items, with candidate positive interactions that are concerned with the newly incorporated individuals being retrieved by applying either Eq 4.40 or 4.41 (with the former being concerned with retrieving candidate positive interactions that involve novel drugs, while the latter being concerned with those that involve mutated protein sequences) on the expanded interaction matrices, where an interaction that incorporates a new individual $d_{new} \in D'$ or $p_{new} \in P'$ is considered as a candidate positive one only if it was estimated as a positive interaction by RSSSTr, while having at least one valid positive interaction that involves one of the new molecule's neighbors and the other molecule participating in the same candidate interaction being correctly estimated as a positive interaction by RSSSTr (note that candidate positive interactions that simultaneously involve mutated

sequences and novel drugs are ignored, since both molecules possess no valid positive interactions to support such assumptions).

$$\begin{aligned} \exists candidate_positive_interaction(p_i, d_{new})\ \ if: \quad & RSSSTr(p_i, d_{new}) \geq th_{RS} \\ And \quad & \exists d_n \in D: Tc\big(ECFP_6(d_{new}), ECFP_6(d_n)\big) \geq th_{Tc_D} \\ And \quad & \exists valid_positive_interaction(p_i, d_n) \\ And \quad & RSSSTr(p_i, d_n) \geq th_{RS}, for\ p_i \in P \end{aligned} \qquad \text{Eq (4.40)}$$

$$\begin{aligned} \exists candidate_positive_interaction(p_{new}, d_j)\ \ if: \quad & RSSSTr(p_{new}, d_j) \geq th_{RS} \\ And \quad & \exists p_n \in P: Tc\big(ECFP_6(p_{new}), ECFP_6(p_n)\big) \geq th_{Tc_P} \\ And \quad & Evalue(p_{new}, p_n) \leq th_{Evalue} \\ And \quad & \exists valid_positive_interaction(p_n, d_j) \\ And \quad & RSSSTr(p_n, d_j) \geq th_{RS}, for\ d_j \in D \end{aligned} \qquad \text{Eq (4.41)}$$

The application of Eq (4.40) on the GPCR set's expanded matrix resulted in a total of 173 drug-based candidate positive interactions, incorporating 3 protein sequences and 173 novel drug candidates that are characterized by having no violations to the eight molecular descriptors listed in Eq (4.39), such as our novel drug that is identified by the canonical SMILES 'CC(C)Nc1nc2c(nc1N1CCC(Oc3ccc(F)cc3F)CC1)CCN(C(= O)C(C)F)C2' (characterized by a *MolWt* value of 477.531 while having 1 H-bond donors, 7 H-bond acceptors, 6 rotatable bonds, 2 aromatic rings, a *MolLogP* value of 3.866, a *TPSA* value of 70.59, and a *FSP* value of 0.542) and its candidate positive interaction with 'P46095' ('G protein-coupled receptor 6' that's associated with 'Lennox-Gastaut Syndrome' and 'Thyroid Dyshormonogenesis' diseases, while being involved in 'Class A/1 (Rhodopsin-like receptors)' pathway [430]), that is estimated to be 100% positive by RSSSTr. Such claim is further strengthened by the novel drug's 136 neighbors (possess 137 neighbors in the GPCR set) which possess valid positive interactions with the candidate protein with $IC50$ values that range between 0.0066 and $60nM$, while having Tc values that range between 0.421 and 0.884 regarding their similarity values with the new drug. Whereas the application of Eq (4.40) on the CDK set's expanded interaction matrix resulted in a total of 1314 drug-based candidate positive interactions that involved 9 protein sequences and 627 of our newly generated drugs that are characterized by having no violations to the molecular descriptors listed in Eq (4.39) such as 'CC(C)N1CCN(c2ccc(Nc3ncc4c5ccncc5n(C5CCCC5)c4n3)nn2)CC1' (characterized by a *MolWt* value of 457.586 while having 1 H-bond donors, 9 H-bond acceptors, 5 rotatable bonds, 4 aromatic rings, a *MolLogP* value of 4.159, a *TPSA* value of 87.89, and a *FSP* value of 0.48) and its candidate positive interaction with 'P11802' ('Cyclin dependent kinase 4' that's considered as an unfavorable prognostic marker in both liver and renal cancers [433], while being associated with both 'Melanoma' and 'cellular senescence' diseases [430]) which is estimated to be 100% positive by RSSSTr, with the new drug's 278 neighbors (possess 346 neighbors in the CDK set) that possess valid positive interactions with the candidate protein with $IC50$ values that range between 0.13 and $100nM$, while having Tc values that range between 0.421 and 0.815 regarding their similarity values with the new drug further supporting such assumption. With top 5 most confident drug-based candidate positive interactions regarding our

novel drugs that are characterized by no violations to the molecular descriptors listed in Eq (4.39) and protein sequences of both GPCR and CDK sets being depicted in Table 4.25.

*Table 4.25 Top 5 most confident drug-based candidate positive interactions that involve generated drugs (with no violations to the 8 molecular descriptors listed in Eq (4.39)) and non-mutated protein sequences across both target-drug sets*

| Set | Candidate protein | New drug SMILES sequence | Count(drug supportive neighbors) | Tc scores between new drug and its neighbors | IC50(candidate protein, supportive neighbors) | RSSSTr(candidate protein, new drug) |
|---|---|---|---|---|---|---|
| GPCR | Q9NQS5 | O=c1nc(OCC2COc3ncccc3O2)cc2n1CCc1cc(C#CCNCc3ccccc3O)ccc1-2 | 128 out of 148 | 0.422 ~ 0.817 | 0.01 ~ 100.0 | 100% |
| | P46095 | CC(C)Nc1nc2c(nc1N1CCC(Oc3ccc(F)cc3F)CC1)CCN(C(=O)C(C)F)C2 | 136 out of 137 | 0.421 ~ 0.884 | 0.007 ~ 60.0 | 100% |
| | P32249 | O=C(/C=C/c1ccc(F)cc1)N1CCN(S(=O)(=O)Cc2ccccc2F)CC1 | 26 out of 27 | 0.461 ~ 0.69 | 1.15 ~ 81.0 | 100% |
| | P46095 | CC(C)Nc1nc2c(nc1N1CCC(Oc3ccc(F)cc3)CC1)CCN(C(=O)C1CC1)C2 | 64 out of 64 | 0.422 ~ 0.723 | 0.01 ~ 0.391 | 100% |
| | Q9NQS5 | O=c1nc(OCC2COc3ncccc3O2)cc2n1CCc1cc(C#CCN3CCCCC3)ccc1-2 | 134 out of 156 | 0.421 ~ 0.848 | 0.01 ~100.0 | 100% |
| CDK | P11802 | CC(C)N1CCN(Cc2ccc(Nc3ncc4c5ccncc5n(C5CCCC5)c4n3)nc2)CC1 | 245 out of 274 | 0.421 ~ 0.781 | 0.13 ~ 100.0 | 100% |
| | P06493 | CC(C)CC(=O)c1ccc(Nc2ncc(Br)c(N[C@H](CO)C(C)(C)O)n2)cc1 | 25 out of 27 | 0.423 ~ 0.506 | 3.0 ~ 96.0 | 100% |
| | Q00534 | CC(C)NC1CCN(c2ccc(Nc3ncc4c5ccncc5n(C5CCCC5)c4n3)nc2)C1 | 183 out of 383 | 0.42 ~ 0.83 | 1.1 ~ 92.0 | 100% |
| | P24941 | CC(C)(C)c1cccc2c1/C(=N/Nc1ccc(S(N)(=O)=O)cc1)C(=O)N2 | 38 out of 39 | 0.424 ~ 0.652 | 1.2 ~ 93.0 | 100% |
| | P50750 | COc1ccc(F)cc1-c1ccnc2[nH]c(C3CCN(CC(=O)CO)CC3)cc12 | 111 out of 160 | 0.42 ~ 0.828 | 1.2 ~ 100.0 | 99.99 % |

As for the protein-based candidate positive interactions that are concerned with the newly integrated mutated sequences; the application of Eq (4.41) on the GPCR set resulted in a total of 378416 candidate positive interactions that involved 1828 mutated protein sequences and 652 drugs (530, 95, and 27 have 0, 1, and 2 rule-of-5 violations respectively), such as the candidate positive interaction that involves both 'Q9Y2T6's mutate sequence (characterized by a mutation in the $314^{th}$ codon that altered ACC to ATC, resulting in the substitution of Threonine with Isoleucine) that is associated with 'Malignant tumor of prostate' [434] and 'CHEMBL4876009', which is based on the high similarity between the mutated sequence and 'Q9HC97' (with an E-value score of $9.83 \times 10^{-38}$ and a Tc similarity of 0.822) alongside the fact that the neighboring protein sequence possess a valid positive interaction with the candidate drug with an $IC50$ value of $3.52nM$, further supporting RSSSTr estimation regarding said candidate interaction being

positive that reached the value of 65.87%. Concluding with the protein-based candidate positive interactions that are deduced by applying Eq (4.41) on the CDK set, which resulted in a total of 6905254 candidate positive interactions that involved 5173 mutated protein sequences and 1863 drugs (1479, 323, 60, and 1 have 0, 1, 2, and 3 rule-of-5 violations respectively), such as the candidate positive interaction that involves both P11802's mutated sequence (characterized by a mutation in the $184^{th}$ codon that altered GAA to AAA, resulting in the substitution of Glutamic acid with Lysine) that is associated with 'CDK4-related condition', 'Familial melanoma', and 'Hereditary cancer-predisposing syndrome' [434] and 'CHEMBL4442620'. Such candidate interaction is based on the high similarity between the mutated sequence and its neighboring proteins 'P11802', 'Q00526', 'P24941', 'P06493', 'P50613', and 'P50750' (with their E-value scores ranging between 0.0 and $1 \times 10^{-46}$ while their Tc similarity values range between 0.81 and 1.0), alongside the fact that these neighboring proteins are in possession of valid positive interactions with the candidate drug with all their $IC50$ values settling at $5nM$, further strengthening RSSSTr assumption regarding said candidate interaction being positive that reached the value of 99.999%. Other protein-based candidate positive interactions that involve mutated protein sequences and drugs with no rule-of-5 violations concerning both GPCR and CDK sets are depicted in Table 4.26.

*Table 4.26 Top 5 protein-based candidate positive interactions that involve mutated protein sequences and existing drugs with no rule-of-5 violations regarding both target-drug sets*

| | Mutated sequence | Mutation cause and location | Mutated sequence neighbors | E-value scores between mutated sequence and its neighbors | Tc scores between mutated sequence and its neighbors | IC50(mutated sequence neighbors, candidate drug) | Candidate drug | RSSSTr |
|---|---|---|---|---|---|---|---|---|
| GPCR | Q9HC97 | TTC→TTG at 170 | P32249 | $6.52\times10^{-23}$ | 0.849 | 18.0 | CHEMBL3262891 | 99.94% |
| | P32249 | CGC→TGC at 138 | Q9NQS5 | $1.1\times10^{-08}$ | 0.802 | 0.01 | CHEMBL3719012 | 100.0% |
| | Q9NQS5 | CTC→CCC at 72 | P32249 | $9.01\times10^{-07}$ | 0.81 | 6.4 | CHEMBL3559986 | 99.79% |
| | P46095 | TTC→TCC at 131 | Q9NQS5 | $7.53\times10^{-07}$ | 0.867 | 0.01 | CHEMBL3718528 | 99.9% |
| | Q9Y2T5 | CTG→CGG at 152 | P46095 | $1.48\times10^{-07}$ | 0.812 | 30.0 | CHEMBL3715787 | 99.99% |
| CDK | P11802 | TCG→TTG at 259 | P11802, P24941, P06493, Q00535, P50613, P50750 | 0.0, $4.59\times10^{-82}$, $1.9\times10^{-77}$, $2.02\times10^{-75}$, $3.56\times10^{-60}$, $1.87\times10^{-54}$ | 1.0, 0.81, 0.812, 0.823, 0.847, 0.832 | 4.2, 14.0, 24.0, 34.0, 20.0, 2.5 | CHEMBL424696 | 94.67% |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Q00534 | GCC→ ACC at 197 | P11802, Q00526, P24941, P06493, P50613, P50750 | $3.25\times10^{-154}$, $6.47\times10^{-96}$, $9.02\times10^{-94}$, $1.2\times10^{-89}$, $8.9\times10^{-62}$, $3.71\times10^{-60}$ | 0.861, 0.853, 0.849, 0.837, 0.833, 0.809 | 5.0, 5.0, 5.0, 5.0, 5.0, 5.0 | CHEMBL 4442620 | 100.0% |
| | P06493 | GTT→ GCT at 105 | P06493, P24941, Q00535, P50613, P11802, P50750 | 0.0, $2.15\times10^{-145}$, $7.55\times10^{-118}$, $1.38\times10^{-85}$, $2.19\times10^{-78}$, $1.1\times10^{-73}$ | 1.0, 0.874, 0.878, 0.809, 0.812, 0.836 | 1.0, 1.0, 1.0, 42.0, 1.0, 1.0 | CHEMBL 2103840 | 100.0% |
| | Q92772 | CCC→ ACC at 200 | P24941, P06493, Q00535, P50613, P50750, P11802 | $7.08\times10^{-76}$, $5.5\times10^{-72}$, $1.46\times10^{-67}$, $1.1\times10^{-58}$, $3.68\times10^{-57}$, $1.0\times10^{-45}$ | 0.852, 0.831, 0.827, 0.818, 0.831, 0.826 | 14.0, 24.0, 34.0, 20.0, 2.5, 4.2 | CHEMBL 424696 | 99.73% |
| | Q5MAI5 | TTT→ TCT at 316 | P24941, P06493, Q00535, P50613, P50750, P11802 | $3.22\times10^{-72}$, $3.63\times10^{-71}$, $1.276\times10^{-68}$, $2.98\times10^{-65}$, $6.5\times10^{-56}$, $7.21\times10^{-51}$ | 0.834, 0.822, 0.804, 0.828, 0.803, 0.822 | 14.0, 24.0, 34.0, 20.0, 2.5, 4.2 | CHEMBL 424696 | 99.08% |

## 4.5. Conclusion

The DL-based approaches discussed in this chapter establish a comprehensive computational framework for advancing intermolecular interaction prediction and *de novo* drug design through modern RS paradigms. By formulating DTI prediction, drug repurposing, and molecular generation as matrix completion and matrix expansion problems; our study bridges theoretical ML constructs with biologically meaningful objectives. The integration of CNN-RNN architectures, and transformer-based models demonstrates how sequential, structural, and contextual dependencies within biological entities can be effectively captured, even in challenging scenarios involving sparse data and orphan protein sequences. The consideration of diverse and clinically relevant biological targets, including cytokines, CDKs, GPCRs, HIV, and SARS-CoV-2, highlights the applicability of the proposed systems across heterogeneous therapeutic landscapes. Rigorous training, evaluation, and biological validation criteria further ensure that predictive performance aligns with domain-specific relevance rather than purely statistical optimization. Collectively, the presented methodologies illustrate the potential of DL-driven recommender and translation systems in enhancing interaction discovery, supporting systematic drug repurposing, and enabling generative exploration of novel compounds, thereby providing scalable and adaptable solutions to contemporary challenges in computational drug discovery in bioinformatics.

# Conclusion and Future Work

In summary, this thesis focused on the challenges arising from the growing scale and complexity of modern biological data, alongside the limited generalization capabilities of existing computational models across intermolecular interaction prediction scenarios. The rapid increase of heterogeneous, high-dimensional, and sparse biological datasets has exposed the weaknesses of approaches that rely on global features, fixed descriptors, and strong dependence on historical interaction records alone. Within this context, researchers were driven by the need to design computational frameworks that can leverage the sequential and structural molecular representations of biological entities to provide more dependable estimations, all while deploying prominent techniques that enable the efficient mitigation of data sparsity across diverse intermolecular interaction settings, involving orphan and mutated protein sequences, alongside newly developed drug candidates. More broadly, this work positions recommender systems as a unifying computational paradigm for bridging biological complexity and heterogeneity, and scalable learning in modern drug discovery workflows.

The first major contribution of this thesis is the development and validation of the RSCRDL and its greedy variant RSCRDL_F as problem-specific, deep learning-based recommender systems tailored to intermolecular interaction prediction. Through the direct exploitation of the sequential representations of proteins and drugs; RSCRDL departs from conventional reliance on global or handcrafted molecular features by demonstrating that richer latent representations can be learned through convolutional and recurrent architectures applied on the aforementioned representations in a natural-language-processing fashion. The integration of content-based components in RSCRDL_F further addresses typical recommender system limitations, such as cold-start and poor generalization, while preserving collaborative filtering strengths. Experimental results on multiple datasets show that sequence-aware recommendation can outperform general-purpose state-of-the-art models, while remaining effective even when applied to orphan protein sequences, highlighting the suitability of the proposed approach for biologically constrained and data-sparse environments.

The second core contribution extends this paradigm toward more advanced predictive and generative settings through the introduction of the transformer-based RSSSTr recommender system and the TSSTr molecular translator. These models jointly exploit sequential and structural molecular representations, enabling the recommendation of both existing and novel drug candidates for regular, mutated, and previously unobserved protein sequences. The proposed FP_indexation mechanism and enhanced encoder architecture improve structural feature extraction and training stability, while the formulation of intermolecular interaction prediction as both a matrix completion and a matrix expansion task opens the door to *de novo* drug design within recommendation frameworks. By coupling generation and relevance-assessment, this contribution demonstrates how recommenders and translators can be synergistically combined to produce chemically valid and biologically meaningful candidate molecules.

Looking forward, several promising research directions emerge from this work. Future efforts may focus on developing more biologically informed recommender systems that integrate deeper domain knowledge with machine learning techniques, ultimately enabling models not only retrieve relevant candidates, but actively generate suitable molecular structures in a manner analogous to

generative adversarial frameworks. Additionally, refining feature extraction mechanisms to localize and model specific molecular interaction sites represents a critical step toward improving interpretability and biological fidelity. Leveraging such fine-grained interaction knowledge could support the targeted generation of molecules that explicitly incorporate functional binding motifs, thereby enhancing the efficiency and relevance of computational drug discovery pipelines.

# References


[1] Hesper, B., & Hogeweg, P. (2021). Bio-informatics: a working concept. A translation of" Bio-informatica: een werkconcept" by B. Hesper and P. Hogeweg. arXiv preprint arXiv:2111.11832.

[2] Huerta, M., Downing, G., Haseltine, F., Seto, B., & Liu, Y. (2000). NIH working definition of bioinformatics and computational biology. US National Institute of Health, 1-1.

[3] Singh, V., & Kumar, A. (Eds.). (2021). Advances in Bioinformatics. Springer.

[4] HERSHEY, A. D., & CHASE, M. (1952). Independent functions of viral protein and nucleic acid in growth of bacteriophage. The Journal of general physiology, 36(1), 39–56. https://doi.org/10.1085/jgp.36.1.39

[5] Watson, J. D., & Crick, F. H. (1953). Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. Nature, 171(4356), 737-738.

[6] Jawdat, D. (2006, April). The era of bioinformatics. In 2006 2nd International Conference on Information & Communication Technologies (Vol. 1, pp. 1860-1865). IEEE.

[7] Chang, M. A., Dayhoff, M. O., Eck, R. V., & Sochard, M. R. (1965). Atlas of protein sequence and structure (No. NASA-CR-71805).

[8] Bellman, R. (1966). Dynamic programming. science, 153(3731), 34-37.

[9] Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48(3), 443-453.

[10] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. Journal of molecular biology, 215(3), 403-410.

[11] Russell, R. M. (1978). The CRAY-1 computer system. Communications of the ACM, 21(1), 63-72.

[12] Gropp, W., Lusk, E., & Skjellum, A. (1999). Using MPI: portable parallel programming with the message-passing interface (Vol. 1). MIT press.

[13] Kernighan, B. W., & Ritchie, D. M. (1988). The C programming language. prentice-Hall.

[14] Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387.

[15] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

[16] Qian, N., & Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. Journal of molecular biology, 202(4), 865-884.

[17] Cerf, V., & Kahn, R. (1974). A protocol for packet network intercommunication. IEEE Transactions on communications, 22(5), 637-648.

[18] Olson, M. V. (1993). The human genome project. Proceedings of the National Academy of Sciences, 90(10), 4338-4344.

[19] Hadiby, S. (2023). La fouille de données à la découverte de motif [Thèse de Doctorat, Université Badji Mokhtar - Annaba].

[20] Zekri, M. (2015). Approches Bio-inspirées Pour La Fouille De Données En Bioinformatique [Thèse de Doctorat, Université Badji Mokhtar - Annaba].

[21] Lesk, A. M. (2025, January 2). bioinformatics. Encyclopedia Britannica. https://www.britannica.com/science/bioinformatics

[22] Trefil, J. , Bertsch, . George F. and McGrayne, . Sharon Bertsch (2025, February 7). atom. Encyclopedia Britannica. https://www.britannica.com/science/atom

[23] Reece, J. B., Urry, L. A., Cain, M. L., Wasserman, S. A., Minorsky, P. V., & Jackson, R. B. (2014). Campbell biology. Pearson.

[24] Chandra, S. (2005). Comprehensive Inorganic Chemistry. New Age International.

[25] Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., Roberts, K., & Walter, P. (2014). Molecular biology of the cell 6th ed (new york, ny: Garland science).

[26] Helmenstine, A. (2024, January 24). Macromolecules - definition, types, examples. Science Notes and Projects. https://sciencenotes.org/macromolecules-definition-types-examples/

[27] Minkin, V. I. (1999). Glossary of terms used in theoretical organic chemistry. Pure and Applied Chemistry, 71(10), 1919-1981.

[28] Whitten, Kenneth W.; Gailey, Kenneth D.; Davis, Raymond E. (1992). "7-3 Formation of covalent bonds". General Chemistry (4th ed.). Saunders College Publishing. p. 264. ISBN 0-03-072373-6.

[29] Brown, T. L. (2009). Chemistry: The Central Science. Pearson Education.

[30] Mahan, G. D. (2025, January 4). crystal. Encyclopedia Britannica. https://www.britannica.com/science/crystal

[31] Britannica, T. Editors of Encyclopaedia (2025, February 4). DNA. Encyclopedia Britannica. https://www.britannica.com/science/DNA

[32] Lodish, H. (2016). Molecular Cell Biology 8th ed. W. H. Freeman.

[33] DNA polymerase structure diagram Biochem Nucleic acids struc. User Guide and Engine Fix Full List. (2024, May 27). https://thyiweassuser.z14.web.core.windows.net/dna-polymerase-structure-diagram.html

[34] Mokobi, F. (2022, May 23). Immune Response- definition, types, factors, examples. Microbe Notes; Sagar Aryal. https://microbenotes.com/immune-response/

[35] Clifton, J. (2021, November 10). What Are Amino Acids? | The Science Blog. ReAgent. https://www.reagent.co.uk/blog/what-are-amino-acids/

[36] Rastogi, S. C., Rastogi, P., & Mendiratta, N. (2022). Bioinformatics: Methods and applications: Genomics, proteomics and drug discovery (5th ed.). PHI Learning.

[37] Stryer, L., Berg, J., Tymoczko, J., & Gatto, G., Jr. (2019). Biochemistry (9th ed.). Macmillan Higher Education.

[38] Understanding protein structure: From primary to Quaternary levels. (2025, January 9). BiologyInsights. https://biologyinsights.com/understanding-protein-structure-from-primary-to-quaternary-levels/

[39] Lesk, A. M. (2013). Introduction to Bioinformatics (4th ed.). Oxford University Press.

[40] Zvelebil, M. J., & Baum, J. O. (2007). Understanding Bioinformatics. CRC Press.

[41] Gaulton, A., Hersey, A., Nowotka, M. L., Patricia Bento, A., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrian-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magarinos, M. P., Overington, J. P., Papadatos, G., Smit, I., & Leach, A. R. (2017). The ChEMBL database in 2017. Nucleic Acids Research, 45(D1). https://doi.org/10.1093/nar/gkw1074

[42] Yates, A. D., Achuthan, P., Akanni, W., Allen, J., Allen, J., Alvarez-Jarreta, J., Amode, M. R., Armean, I. M., Azov, A. G., Bennett, R., Bhai, J., Billis, K., Boddu, S., Marugán, J. C., Cummins, C., Davidson, C., Dodiya, K., Fatima, R., Gall, A., … Flicek, P. (2020). Ensembl 2020. Nucleic Acids Research, 48(D1), D682–D688. https://doi.org/10.1093/nar/gkz966

[43] Piñero, J., Bravo, À., Queralt-Rosinach, N., Gutiérrez-Sacristán, A., Deu-Pons, J., Centeno, E., ... & Furlong, L. I. (2016). DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. Nucleic acids research, gkw943.

[44] Falagas, M. E., Pitsouni, E. I., Malietzis, G. A., & Pappas, G. (2008). Comparison of PubMed, Scopus, web of science, and Google scholar: strengths and weaknesses. The FASEB journal, 22(2), 338-342.

[45] GenBank overview. (n.d.). Nih.gov. Retrieved March 8, 2025, from https://www.ncbi.nlm.nih.gov/genbank/

[46] Ensembl genome browser 113. (n.d.). Ensembl.org. Retrieved March 8, 2025, from https://www.ensembl.org/index.html

[47] DDBJ. (n.d.). Nig.Ac.Jp. Retrieved March 8, 2025, from https://www.ddbj.nig.ac.jp/index-e.html

[48] UniProt. (n.d.). UniProt. Retrieved March 8, 2025, from https://www.uniprot.org/

[49] RefSeq: NCBI reference sequence database. (n.d.). Nih.gov. Retrieved March 8, 2025, from https://www.ncbi.nlm.nih.gov/refseq/

[50] RCSB Protein Data Bank. (n.d.). RCSB PDB: Homepage. Rcsb.org. Retrieved March 8, 2025, from https://www.rcsb.org/

[51] The Cambridge Structural Database. (n.d.). Cam.ac.uk. Retrieved March 8, 2025, from https://www.ccdc.cam.ac.uk/solutions/software/csd/

[52] Nucleic acid knowledge base. (n.d.). Nakb.org. Retrieved March 8, 2025, from https://www.nakb.org/

[53] Home - GEO - NCBI. (n.d.). Nih.gov. Retrieved March 8, 2025, from https://www.ncbi.nlm.nih.gov/geo/

[54] BioStudies. (n.d.). BioStudies < the European bioinformatics institute < EMBL-EBI. Ebi.ac.uk. Retrieved March 8, 2025, from https://www.ebi.ac.uk/biostudies/arrayexpress

[55] Stanford Microarray Database [defunct]. (n.d.). Princeton.edu. Retrieved March 8, 2025, from http://smd.princeton.edu/

[56] KEGG: Kyoto Encyclopedia of Genes and genomes. (n.d.). Genome.Jp. Retrieved March 8, 2025, from https://www.genome.jp/kegg/

[57] (N.d.-a). Metacyc.org. Retrieved March 8, 2025, from https://metacyc.org/

[58] Reactome pathway database. (n.d.). Reactome.org. Retrieved March 8, 2025, from https://curator.reactome.org/

[59] Biocyc.org. Retrieved March 8, 2025, from https://www.biocyc.org/

[60] Home. Wikipathways.org. Retrieved March 8, 2025, from https://www.wikipathways.org/

[61] PubChem. PubChem. Nih.gov. Retrieved March 8, 2025, from https://pubchem.ncbi.nlm.nih.gov/

[62] ChEMBL - ChEMBL. (n.d.). Ebi.ac.uk. Retrieved March 8, 2025, from https://www.ebi.ac.uk/chembl/

[63] ChemSpider: Search and share chemistry - homepage. (n.d.). Chemspider.com. Retrieved March 8, 2025, from https://www.chemspider.com/

[64] Drugbank.com. Retrieved March 8, 2025, from https://go.drugbank.com/

[65] Genomes. Nih.gov. Retrieved March 8, 2025, from https://www.ncbi.nlm.nih.gov/home/genomes/

[66] Home - SNP - NCBI. (n.d.). Nih.gov. Retrieved March 8, 2025, from https://www.ncbi.nlm.nih.gov/snp/

[67] EMBL-EBI. (n.d.-b). PRIDE - Proteomics Identification Database. Ebi.ac.uk. Retrieved March 8, 2025, from https://www.ebi.ac.uk/pride/

[68] PeptideAtlas. (n.d.). Peptideatlas.org. Retrieved March 8, 2025, from https://peptideatlas.org/

[69] STRING: functional protein association networks. (n.d.). String-db.org. Retrieved March 8, 2025, from https://string-db.org/

[70] Lab, M. T. (n.d.). BioGRID interaction database. Thebiogrid.org. Retrieved March 8, 2025, from https://thebiogrid.org/

[71] EMBL-EBI. (n.d.-a). IntAct portal. Ebi.ac.uk. Retrieved March 8, 2025, from https://www.ebi.ac.uk/intact/home

[72] STITCH: chemical association networks. (n.d.). Embl.de. Retrieved March 8, 2025, from http://stitch.embl.de/

[73] Liu, T. (n.d.). Binding database home. Bindingdb.org. Retrieved March 8, 2025, from https://www.bindingdb.org/rwd/bind/index.jsp

[74] The Cancer Genome Atlas program (TCGA). (2022, May 13). Cancer.gov. https://www.cancer.gov/ccg/research/genome-sequencing/tcga

[75] ClinVar. (n.d.). ClinVar. Nih.gov. Retrieved March 8, 2025, from https://www.ncbi.nlm.nih.gov/clinvar/

[76] Home - OMIM. (n.d.). Omim.org. Retrieved March 8, 2025, from https://www.omim.org/

[77] Orphanet. (n.d.). Orpha.net. Retrieved March 8, 2025, from https://www.orpha.net/

[78] Nih.gov. Retrieved March 8, 2025, from https://pubmed.ncbi.nlm.nih.gov/

[79] Scopus.com. Retrieved March 8, 2025, from https://www.scopus.com/home.uri;

[80] Clarivate. (n.d.). Clarivate.com. Retrieved March 8, 2025, from https://access.clarivate.com/login?app=wos&alternative=true&shibShireURL=https:%2F%2Fwww.webofknowledge.com%2F%3Fauth%3DShibboleth&shibReturnURL=https:%2F%2Fwww.webofknowledge.com%2F%3Fmode%3DNextgen%26action%3Dtransfer%26path%3D%252Fwos%26DestApp%3DUA&referrer=mode%3DNextgen%26path%3D%252Fwos%26DestApp%3DUA%26action%3Dtransfer&roaming=true

[81] Mount, D. W., & Mount, D. W. (2001). Bioinformatics: sequence and genome analysis (Vol. 564). Cold Spring Harbor, NY: Cold spring harbor laboratory press.

[82] Mount, D. W. (2004). Bioinformatics: Sequence and Genome Analysis (2nd ed.). Cold Spring Harbor Laboratory Press.

[83] Al-Amrani, S., Al-Jabri, Z., Al-Zaabi, A., Alshekaili, J., & Al-Khabori, M. (2021). Proteomics: Concepts and applications in human medicine. World journal of biological chemistry, 12(5), 57–69. https://doi.org/10.4331/wjbc.v12.i5.57

[84] De Filippo, C., Ramazzotti, M., Fontana, P., & Cavalieri, D. (2012). Bioinformatic approaches for functional annotation and pathway inference in metagenomics data. Briefings in Bioinformatics, 13(6), 696–710. https://doi.org/10.1093/bib/bbs070

[85] Gasteiger, J., & Engel, T. (Eds.). (2006). Chemoinformatics: a textbook. John Wiley & Sons.

[86] Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. Nature Reviews. Genetics, 16(6), 321–332. https://doi.org/10.1038/nrg3920

[87] O'Donoghue, S. I., Gavin, A.-C., Gehlenborg, N., Goodsell, D. S., Hériché, J.-K., Nielsen, C. B., North, C., Olson, A. J., Procter, J. B., Shattuck, D. W., Walter, T., & Wong, B. (2010). Visualizing biological data-now and in the future. Nature Methods, 7(3 Suppl), S2-4. https://doi.org/10.1038/nmeth.f.301

[88] Basic Local Alignment Search Tool. (n.d.). Nih.gov. Retrieved March 16, 2025, from https://blast.ncbi.nlm.nih.gov/Blast.cgi

[88] Dineen, D. (n.d.). Clustal Omega - fast, accurate, scalable multiple sequence alignment for proteins. Clustal.org. Retrieved March 16, 2025, from http://www.clustal.org/omega/

[90] HMMER. (n.d.). Hmmer.org. Retrieved March 16, 2025, from http://hmmer.org/

[91] Spades: SPAdes genome assembler. (n.d.).

[92] GATK. (n.d.). Broadinstitute.org. Retrieved March 16, 2025, from https://gatk.broadinstitute.org/hc/en-us

[93] Stanke, M. (2003). The AUGUSTUS gene prediction tool. https://bioinf.uni-greifswald.de/augustus/

[94] SwissDock. (n.d.). Swissdock.Ch. Retrieved March 16, 2025, from https://www.swissdock.ch/

[95] AlphaFold Protein Structure Database. (n.d.). AlphaFold protein structure database. Alphafold.com. Retrieved March 16, 2025, from https://alphafold.com/

[96] PyMOL. (n.d.). Pymol.org. Retrieved March 16, 2025, from https://pymol.org/

[97] KEGG mapper. (n.d.). Genome.Jp. Retrieved March 16, 2025, from https://www.genome.jp/kegg/mapper/

[98] DAVID Functional Annotation Bioinformatics Microarray Analysis. (n.d.). Nih.gov. Retrieved March 16, 2025, from https://davidbioinformatics.nih.gov/

[99] STRING: functional protein association networks. (n.d.). String-db.org. Retrieved March 16, 2025, from https://string-db.org/

[100] AutoDock. (n.d.). AutoDock. Retrieved March 16, 2025, from https://autodock.scripps.edu/

[101] Schrödinger - physics-based software platform for molecular discovery & design. (2023, July 26). Schrödinger. https://www.schrodinger.com/

[102] Landrum, G. (n.d.). RDKit. Rdkit.org. Retrieved March 16, 2025, from https://www.rdkit.org/

[103] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234-1240.

[104] Scikit-learn. (n.d.). Scikit-learn.org. Retrieved March 16, 2025, from https://scikit-learn.org/stable/

[105] UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction — umap 0.5.8 documentation. (n.d.). Readthedocs.Io. Retrieved March 16, 2025, from https://umap-learn.readthedocs.io/en/latest/

[106] Ono, K. (n.d.). Cytoscape. Cytoscape.org. Retrieved March 16, 2025, from https://cytoscape.org/

[107] IGV: Integrative genomics viewer. (n.d.). Igv.org. Retrieved March 16, 2025, from https://igv.org/

[108] Ggplot2. (n.d.). Tidyverse.org. Retrieved March 16, 2025, from https://ggplot2.tidyverse.org/

[109] QIIME 2 homepage. (n.d.). QIIME 2. Retrieved March 16, 2025, from https://qiime2.org/

[110] Krueger, F., & Andrews, S. R. (2011). Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. Bioinformatics (Oxford, England), 27(11), 1571–1572. https://doi.org/10.1093/bioinformatics/btr167

[111] Benton, H. P., Wong, D. M., Trauger, S. A., & Siuzdak, G. (2008). XCMS2: processing tandem mass spectrometry data for metabolite identification and structural characterization. Analytical Chemistry, 80(16), 6382–6389. https://doi.org/10.1021/ac800795f

[112] The role of bioinformatics in genomic research. (n.d.). Falconediting.com. Retrieved March 18, 2025, from https://falconediting.com/en/blog/the-role-of-bioinformatics-in-genomic-research/

[113] Beeton-Kempen, N. (2020, December 9). Proteomics: Principles, techniques and applications. Proteomics & Metabolomics from Technology Networks; Technology Networks. https://www.technologynetworks.com/proteomics/articles/proteomics-principles-techniques-and-applications-343804

[114] Lowe, R., Shirley, N., Bleackley, M., Dolan, S., & Shafee, T. (2017). Transcriptomics technologies. PLoS Computational Biology, 13(5), e1005457. https://doi.org/10.1371/journal.pcbi.1005457

[115] Kaari, M., Joseph, J., Manikkam, R., Shamya, M., & Aruni, W. (2022). Application of bioinformatic tools for phylogenetic analysis. In Methods in Actinobacteriology (pp. 187–191). Springer US.

[116] Yang, Z., & Rannala, B. (2012). Molecular phylogenetics: principles and practice. Nature Reviews. Genetics, 13(5), 303–314. https://doi.org/10.1038/nrg3186

[117] Somda, D., Wilson Kpordze, S., Jerpkorir, M., Chantelle Mahora, M., Wanjiru Ndungu, J., Wambui Kamau, S., … Elbasyouni, A. (2024). The Role of Bioinformatics in Drug Discovery: A Comprehensive Overview. IntechOpen. doi: 10.5772/intechopen.113712

[118] Ekins, S., Mestres, J., & Testa, B. (2007). In silico pharmacology for drug discovery: applications to targets and beyond. British Journal of Pharmacology, 152(1), 21–37. https://doi.org/10.1038/sj.bjp.0707306

[119] Yang, Y., Xie, B., & Yan, J. (2014). Application of next-generation sequencing technology in forensic science. Genomics, Proteomics & Bioinformatics, 12(5), 190–197. https://doi.org/10.1016/j.gpb.2014.09.001

[120] Sharma, V. (2020). Bioinformatics and its applications in environmental science and health and its applications in other disciplines. Sambodhi J, 4, 88-93.

[121] Noor, H. (2023, November 23). Role of bioinformatics in Environmental Science studies. Envpk.com - Best Environmental Website Of Pakistan. https://www.envpk.com/role-of-bioinformatics-in-environmental-science-studies/

[122] Khalid, M. N., & Joint International Research Laboratory of Agriculture and Agri-Product Safety, College of Agriculture, Yangzhou University, China. (2021). Application and potential use of advanced bioinformatics techniques in agriculture and animal sciences. Indian Journal of Pure & Applied Biosciences, 9(3), 237–246. https://doi.org/10.18782/2582-2845.8702

[123] Kavi Kishor, P. B., Bandyopadhyay, R., & Suravajhala, P. (Eds.). (2014). Agricultural Bioinformatics (2014th ed.). Springer.

[124] Bioinformatics and its application. (2024, April 26). GeeksforGeeks. https://www.geeksforgeeks.org/bioinformatics-application/

[125] Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In Journal of physics: conference series (Vol. 1142, p. 012012). IOP Publishing.

[126] IBM. (2021, September 22). What Is Machine learning? Ibm.com; IBM. https://www.ibm.com/think/topics/machine-learning

[127] Rani, P., Kumar, R., Jain, A., & Lamba, R. (2020). Taxonomy of machine learning algorithms and its applications. Journal of Computational and Theoretical Nanoscience, 17(6), 2508–2513. https://doi.org/10.1166/jctn.2020.8922

[128] Topol, E. (2019). Deep medicine: How artificial intelligence can make healthcare human again. Basic Books.

[129] Dal Pozzolo, A. (2015). Machine Learning for Credit Card Fraud Detection. Elsevier.

[130] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to end learning for self-driving cars. In arXiv [cs.CV]. http://arxiv.org/abs/1604.07316

[131] Francesco Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender Systems Handbook.

[132] Köseoğlu, E. D. (2024, September 9). Machine Learning vs. Traditional Programming: What's the difference? Medium. https://medium.com/@ekinkoseoglu/machine-learning-vs-traditional-programming-whats-the-difference-0ee6db6bee59

[133] Narsingh. (2024, August 23). Machine learning vs traditional programming: Key differences and benefits. TechInfer. https://techinfer.com/machine-learning-vs-traditional-programming/

[134] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3), 160.

[135] Aggarwal, C. C. (2015). Data mining: the textbook (Vol. 1, No. 3). New York: springer.

[136] Nasteski, V. (2017). An overview of the supervised machine learning methods. Horizons. b, 4(51-62), 56.

[137] Shmueli, G. (2010). To explain or to predict?.

[138] Raj, R. (n.d.). Classification and regression in machine learning. Enjoyalgorithms.com. Retrieved April 15, 2025, from https://www.enjoyalgorithms.com/blogs/classification-and-regression-in-machine-learning

[139] Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer. https://doi.org/10.1007/b94608

[140] James, G., Witten, D., & Hastie, T. (2013). An introduction to statistical learning: With applications in R. Springer.

[141] Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

[142] Datacamp.com. Retrieved April 19, 2025, from https://www.datacamp.com/blog/classification-machine-learning

[143] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284. https://doi.org/10.1109/tkde.2008.239

[144] Zhang, M.-L., & Zhou, Z.-H. (2014). A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 26(8), 1819–1837. https://doi.org/10.1109/tkde.2013.39

[145] vandita2t53 Improve, V. (2025, May 17). Hierarchical classification. GeeksforGeeks. https://www.geeksforgeeks.org/hierarchical-classification/

[146] Silla, C. N., & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. Data mining and knowledge discovery, 22, 31-72.

[147] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. The Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

[148] Javed, T. A., Shahzad, W., & Arshad, U. (2021). Hierarchical Text Classification of Urdu news using deep neural network. In arXiv [cs.CL]. https://doi.org/10.48550/ARXIV.2107.03141

[149] Shyam, R., & Singh, R. (2021). A taxonomy of machine learning techniques. Journal of Advancements in Robotics, 8(3), 18-25p.

[150] Zimek, A., Schubert, E., & Kriegel, H. P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5), 363-387.

[151] Ngai, E. W., Xiu, L., & Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. Expert systems with applications, 36(2), 2592-2602.

[152] Intercluster and intracluster distance. (2019, May 10). GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/ml-intercluster-and-intracluster-distance/

[153] Rokach, L., & Maimon, O. (2006). Clustering Methods. In Data Mining and Knowledge Discovery Handbook (pp. 321–352). Springer-Verlag.

[154] AnalytixLabs. (2020, July 5). What is Clustering in Machine Learning: Types and Methods. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms/

[155] J, E. (2023, September 25). 6 Different Types of Clustering: All You Need To Know! Datarundown. https://datarundown.com/types-of-clustering/

[156] OpenAI. (2025). Diagram illustrating three types of unsupervised learning. ChatGPT. https://chat.openai.com/

[157] Nisbet, R. (2016). Handbook of statistical analysis and data mining applications. Academic Press.

[158] Tripathi, A. (2024, January 22). Apriori Algorithm In Data Mining: Implementation, Examples, and More. Analytixlabs. https://www.analytixlabs.co.in/blog/apriori-algorithm-in-data-mining/

[159] IBM. (2021b, September 23). Unsupervised Learning. Ibm.com. https://www.ibm.com/think/topics/unsupervised-learning

[160] Van Der Maaten, L., Postma, E. O., & Van Den Herik, H. J. (2009). Dimensionality reduction: A comparative review. Journal of machine learning research, 10(66-71), 13.

[161] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[162] Abid Ali Awan. (2023, August 22). What is a Generative Model? Datacamp.com; DataCamp. https://www.datacamp.com/blog/what-is-a-generative-model

[163] Chapelle, O., Schölkopf, B., & Zien, A. (2006). Semi-supervised learning MIT Press Cambridge.

[164] Bergmann, D. (2023, December 8). What is semi-supervised learning? IBM. https://www.ibm.com/think/topics/semi-supervised-learning

[165] GeeksforGeeks. (2019, May 28). Semi-Supervised Learning in ML. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/ml-semi-supervised-learning/

[166] Lee, D., Seo, H., & Jung, M. W. (2012). Neural basis of reinforcement learning and decision making. Annual review of neuroscience, 35(1), 287-308.

[167] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction 2nd ed. MIT press Cambridge, 1(2), 25.

[168] GeeksforGeeks. (2018, April 25). Reinforcement Learning. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/what-is-reinforcement-learning/

[169] Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

[170] AskDataScience. (2020, February 19). What are the most common data types in data science? - Ask Data Science! Askdatascience.com. https://askdatascience.com/834/what-are-the-most-common-data-types-in-data-science

[171] Wulff, S. S. (2017). Time series analysis: Forecasting and control.

[172] Kelleher, J. D., Mac Namee, B., & D'arcy, A. (2020). Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT press.

[173] Badman, A., & Kosinski, M. (2024, October 3). What is data? IBM. https://www.ibm.com/think/topics/data

[174] Abiteboul, S., Buneman, P., & Suciu, D. (2000). Data on the web: from relations to semistructured data and XML. Morgan Kaufmann.

[175] Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. IEEE Data Eng. Bull., 23(4), 3-13.

[176] Tenzing Lagyal. (2024, October 31). A Comprehensive Guide to Feature Encoding in Machine Learning. Medium. https://medium.com/@lagyal/a-comprehensive-guide-to-feature-encoding-in-machine-learning-e799db6d6a30

[177] Schütze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. Introduction to information retrieval. Vol. 39. Cambridge: Cambridge University Press, 2008.

[178] Gonzalez, R. C., & Woods, R. E. (2017). Digital Image Processing (4th ed.). Pearson.

[179] Kotsiantis, S., & Kanellopoulos, D. (2006). Discretization techniques: A recent survey. GESTS International Transactions on Computer Science and Engineering, 32(1), 47-58.

[180] Nair, A. (2022, October 4). An Intro to Discretization Techniques for Machine Learning. Towards Data Science. https://towardsdatascience.com/an-intro-to-discretization-techniques-for-machine-learning-93dce1198e68/

[181] Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.

[182] Belcic, I., & Stryker, C. (Eds.). (2023, October 14). What is an ML pipeline? IBM. https://www.ibm.com/think/topics/machine-learning-pipeline

[183] GeeksforGeeks. (2025, March 27). Model Selection for Machine Learning. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/model-selection-for-machine-learning/

[184] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. The journal of machine learning research, 13(1), 281-305.

[185] Paleyes, A., Urma, R. G., & Lawrence, N. D. (2022). Challenges in deploying machine learning: a survey of case studies. ACM computing surveys, 55(6), 1-29.

[186] Gupta, A., Sharma, A., & Goel, A. (2017). Review of regression analysis models. Int. J. Eng. Res. Technol, 6(08), 58-61.

[187] Seber, G. A., & Lee, A. J. (2003). Linear regression analysis. John Wiley & Sons.

[188] Zach Bobbitt. (2020, January 8). The Four Assumptions of Linear Regression. Statology. https://www.statology.org/linear-regression-assumptions/

[189] Kutner. (2004). Applied linear statistical models (5th ed.). McGraw-Hill.

[190] Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). Logistic regression (p. 536). New York: Springer-Verlag.

[191] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

[192] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1), 267-288.

[193] Lawton, G. (2024, July 30). What is regression in machine learning? Search Enterprise AI; TechTarget. https://www.techtarget.com/searchEnterpriseAI/feature/What-is-regression-in-machine-learning

[194] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[195] Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24(12), 1565-1567.

[196] scikit-learn. (2018). Support Vector Machines. Scikit-Learn. https://scikit-learn.org/stable/modules/svm.html

[197] Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). Classification and regression trees. Chapman and Hall/CRC.

[198] Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. Data mining and knowledge discovery, 2(4), 345-389.

[199] Misra, S. (2023, August 16). Interpretable AI: Decision Trees. Medium. https://medium.com/@shrutimisra/interpretable-ai-decision-trees-f9698e94ef9b

[200] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[201] Shakhnarovich, G., Darrell, T., & Indyk, P. (2008). Nearest-neighbor methods in learning and vision. IEEE Trans. Neural Networks, 19(2), 377.

[202] Lewis, D. D. (1998, April). Naive (Bayes) at forty: The independence assumption in information retrieval. In European conference on machine learning (pp. 4-15). Berlin, Heidelberg: Springer Berlin Heidelberg.

[203] Zhang, H. (2004). The optimality of naive Bayes. Aa, 1(2), 3.

[204] GeeksforGeeks. (2023, December 14). How to Find The Optimal Value of K in KNN. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/how-to-find-the-optimal-value-of-k-in-knn/

[205] Raizada, R. D., & Lee, Y. S. (2013). Smoothness without smoothing: why Gaussian naive Bayes is not naive for multi-subject searchlight studies. PloS one, 8(7), e69566.

[206] Jolliffe, I. T. (2002). Principal component analysis for special types of data (pp. 338-372). Springer New York.

[207] Marridi, Z. (2016, November 7). Principal Component Analysis (PCA) 101. NumXL. https://numxl.com/blogs/principal-component-analysis-pca-101/

[208] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

[209] Kaneko Laboratory. (2018, June 3). t-distributed Stochastic Neighbor Embedding (t-SNE). Https://Datachemeng.com/Tsne/. https://datachemeng.com/tsne/

[210] Debnath, P. (2025, July 2). What Are Deep Fakes? Understanding the Growing Threat in the Digital Age. Cloudsek.com. https://www.cloudsek.com/knowledge-base/what-are-deep-fakes-understanding-the-growing-threat-in-the-digital-age

[211] Kingma, D. P., & Welling, M. (2013, December). Auto-encoding variational bayes.

[212] Kurtis Pykes. (2023, August 22). Variational Autoencoders: How They Work and Why They Matter. Datacamp.com; DataCamp. https://www.datacamp.com/blog/what-is-a-generative-model

[213] Shi, J. (2017). Variational-autoencoder. GitHub. https://github.com/shijy07/variational-autoencoder-1

[214] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666.

[215] Arthur, D., & Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Stanford.

[216] Johnson, W. (2015). K-Means Clustering – What it is and How it Works. Learn by Marketing. https://www.learnbymarketing.com/methods/k-means-clustering/

[217] Srikant, R. (1996). Fast algorithms for mining association rules and sequential patterns. The University of Wisconsin-Madison.

[218] Borgelt, C. (2012). Frequent item set mining. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2(6), 437-456.

[219] Jabeen, H. (2018, August 21). Market Basket Analysis using R. Data Camp. https://www.datacamp.com/tutorial/market-basket-analysis-r

[220] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[221] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[222] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1, No. 2). Cambridge: MIT press.

[223] Pop, A. C., Cai, Z., & Gyselinck, J. J. (2023). Machine-learning aided multiobjective optimization of electric machines—geometric-feasibility and enhanced regression models. IEEE Journal of Emerging and Selected Topics in Industrial Electronics, 4(3), 844-854.

[224] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.

[225] Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256). JMLR Workshop and Conference Proceedings.

[226] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Cham: Springer International Publishing.

[227] Boureau, Y. L., Ponce, J., & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 111-118).

[228] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[229] Zone, S. (2020, May 22). The Convolution Parameters Calculation. Medium. https://medium.com/@shahzaibnoor40/the-convolution-parameters-calculation-b2394da8dd59

[230] Zgallai, W. A., & Ozsahin, D. U. (Eds.). (2024). Artificial intelligence and image processing in medical imaging. Elsevier.

[231] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[232] Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. Design and applications, 5(64-67), 2.

[233] Pascanu, R., Mikolov, T., & Bengio, Y. (2013, May). On the difficulty of training recurrent neural networks. In International conference on machine learning (pp. 1310-1318). Pmlr.

[234] Introduction to recurrent neural networks. (2018, October 3). GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/introduction-to-recurrent-neural-network/

[235] Sayed, H. M., ElDeeb, H. E., & Taie, S. A. (2023). Bimodal variational autoencoder for audiovisual speech recognition. Machine Learning, 112(4), 1201-1226.

[236] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[237] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[238] Anishnama. (2023, May 4). Understanding Gated Recurrent Unit (GRU) in Deep Learning. Medium. https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2

[239] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[240] Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2023). Efficient Transformers: A survey. ACM Computing Surveys, 55(6), 1–28. https://doi.org/10.1145/3530811

[241] Vig, J. (2019). A multiscale visualization of attention in the transformer model. arXiv preprint arXiv:1906.05714.

[242] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.

[243] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103).

[244] Jimenez, M., Torres, M. T., John, R., & Triguero, I. (2020). Galaxy image classification based on citizen science data: A comparative study. IEEE Access, 8, 47232-47246.

[245] Mangini Jagadish. (2025, February 18). Advantages and Disadvantages of Autoencoders. Online Tutorial Hub. https://onlinetutorialhub.com/machine-learning-tutorial/advantages-and-disadvantages-of-autoencoders/

[246] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[247] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine learning, 63(1), 3-42.

[248] Louppe, G. (2014). Understanding random forests: From theory to practice. Universite de Liege (Belgium).

[249] David, D. (2020, August 6). Random Forest Classifier Tutorial: How to Use Tree-Based Algorithms for Machine Learning. FreeCodeCamp. https://www.freecodecamp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/

[250] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

[251] Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.

[252] Mason, L., Baxter, J., Bartlett, P., & Frean, M. (1999). Boosting algorithms as gradient descent. Advances in neural information processing systems, 12.

[253] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139.

[254] Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications, 6, 100134.

[255] Behzadi, F. (2015). Natural language processing and machine learning: A review. International Journal of Computer Science and Information Security, 13(9), 101.

[256] Egon, A., & Klinton, B. (2024). APPLICATIONS OF ML IN NATURAL LANGUAGE PROCESSING (NLP). Available at SSRN.

[257] Dash, T. (2017). A study on intrusion detection using neural networks trained with evolutionary algorithms. Soft Computing, 21(10), 2687-2700.

[258] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., Chen, S., Liu, D., & Li, J. (2020). Performance comparison and current challenges of using machine learning techniques in cybersecurity. Energies, 13(10), 2509.

[259] Covington, P., Adams, J., & Sargin, E. (2016, September). Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM conference on recommender systems (pp. 191-198).

[260] Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., & Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy. Knowledge and Information Systems, 64(1), 35-74.

[261] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). A guide to deep learning in healthcare. Nature medicine, 25(1), 24-29.

[262] Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep learning for healthcare: review, opportunities and challenges. Briefings in bioinformatics, 19(6), 1236-1246.

[263] Shah, J., Vaidya, D., & Shah, M. (2022). A comprehensive review on multiple hybrid deep learning approaches for stock prediction. Intelligent Systems with Applications, 16, 200111.

[264] Nandi, B., Jana, S., & Das, K. P. (2023). Machine learning-based approaches for financial market prediction: A comprehensive review. Journal of AppliedMath, 1(2.1), 134-134.

[265] Bertolini, M., Mezzogori, D., Neroni, M., & Zammori, F. (2021). Machine Learning for industrial applications: A comprehensive literature review. Expert Systems with Applications, 175, 114820.

[266] Gourisaria, M. K., Agrawal, R., Harshvardhan, G. M., Pandey, M., & Rautaray, S. S. (2021). Application of machine learning in industry 4.0. In Machine learning: Theoretical foundations and practical applications (pp. 57-87). Singapore: Springer Singapore.

[267] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and electronics in agriculture, 147, 70-90.

[268] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.

[269] Shalu, S. G. (2023). Environmental monitoring with machine learning. EPRA International Journal of Multidisciplinary Research (IJMR), 208-212.

[270] Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., ... & Bengio, Y. (2022). Tackling climate change with machine learning. ACM Computing Surveys (CSUR), 55(2), 1-96.

[271] Fatima, A. (2024, March 5). Overfitting, Underfitting and Model's Capacity in Deep Learning. Machine Mindscape. https://machinemindscape.com/overfitting-underfitting-and-models-capacity-in-deep-learning/

[272] Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

[273] Bottou, L., & Bousquet, O. (2008). The tradeoffs of large scale learning, advances in neural information processing systems, 20.

[274] Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78-87.

[275] Altman, N., & Krzywinski, M. (2018). The curse (s) of dimensionality. Nat Methods, 15(6), 399-400.

[276] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

[277] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence, 1(5), 206-215.

[278] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

[179] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., ... & Dean, J. (2021). Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350.

[280] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.

[281] Eubanks, V. (2018). Automating inequality: How high-tech tools profile, police, and punish the poor. St. Martin's Press.

[282] O'neil, C. (2017). Weapons of math destruction: How big data increases inequality and threatens democracy. Crown.

[283] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017, May). Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP) (pp. 3-18). IEEE.

[284] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017, April). Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security (pp. 506-519).

[285] Copeland, R. A. (2013). Evaluation of enzyme inhibitors in drug discovery: a guide for medicinal chemists and pharmacologists. John Wiley & Sons.

[286] Hughes, J. P., Rees, S., Kalindjian, S. B., & Philpott, K. L. (2011). Principles of early drug discovery. British journal of pharmacology, 162(6), 1239-1249.

[287] C, B. G. (2023, April 7). DRUG DISCOVERY AND DEVELOPMENT: A STEP-BY-STEP PROCESS. The Science Notes. https://thesciencenotes.com/drug-discovery-process-step-by-step/

[288] Schenone, M., Dančík, V., Wagner, B. K., & Clemons, P. A. (2013). Target identification and mechanism of action in chemical biology and drug discovery. Nature chemical biology, 9(4), 232-240.

[289] Macarron, R., Banks, M. N., Bojanic, D., Burns, D. J., Cirovic, D. A., Garyantes, T., ... & Sittampalam, G. S. (2011). Impact of high-throughput screening in biomedical research. Nature reviews Drug discovery, 10(3), 188-195.

[290] Waring, M. J., Arrowsmith, J., Leach, A. R., Leeson, P. D., Mandrell, S., Owen, R. M., ... & Weir, A. (2015). An analysis of the attrition of drug candidates from four major pharmaceutical companies. Nature reviews Drug discovery, 14(7), 475-486.

[291] van der Worp, H. B., Howells, D. W., Sena, E. S., Porritt, M. J., Rewell, S., O'Collins, V., & Macleod, M. R. (2010). Can animal models of disease reliably inform human studies?. PLoS medicine, 7(3), e1000245.

[292] Friedman, L. M., Furberg, C. D., DeMets, D. L., Reboussin, D. M., & Granger, C. B. (2015). Fundamentals of clinical trials. springer.

[293] FDA. (2022, January 21). New Drug Application (NDA). U.S. Food and Drug Administration. https://www.fda.gov/drugs/types-applications/new-drug-application-nda

[294] World Health Organization, & Who Collaborating Centre For International Drug Monitoring. (2002). The importance of pharmacovigilance. World Health Organization.

[295] DiMasi, J. A., Grabowski, H. G., & Hansen, R. W. (2016). Innovation in the pharmaceutical industry: new estimates of R&D costs. Journal of health economics, 47, 20-33.

[296] Pushpakom, S., Iorio, F., Eyers, P. A., Escott, K. J., Hopper, S., Wells, A., ... & Pirmohamed, M. (2019). Drug repurposing: progress, challenges and recommendations. Nature reviews Drug discovery, 18(1), 41-58.

[297] Nosengo, N. (2016). Can you teach old drugs new tricks?.

[298] Campesi, I., Racagni, G., & Franconi, F. (2021). Just a reflection: does drug repurposing perpetuate sex-gender bias in the safety profile?. Pharmaceuticals, 14(8), 730.

[299] Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., ... & Zhao, S. (2019). Applications of machine learning in drug discovery and development. Nature reviews Drug discovery, 18(6), 463-477.

[300] Ekins, S., Puhl, A. C., Zorn, K. M., Lane, T. R., Russo, D. P., Klein, J. J., ... & Clark, A. M. (2019). Exploiting machine learning for end-to-end drug discovery and development. Nature materials, 18(5), 435-441.

[301] Szklarczyk, D., Gable, A. L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., ... & Mering, C. V. (2019). STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic acids research, 47(D1), D607-D613.

[302] Keskin, O., Gursoy, A., Ma, B., & Nussinov, R. (2008). Principles of protein− protein interactions: what are the preferred ways for proteins to interact?. Chemical reviews, 108(4), 1225-1244.

[303] Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., & Kanehisa, M. (2008). Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. Bioinformatics, 24(13), i232-i240.

[304] Nooren, I. M., & Thornton, J. M. (2003). Diversity of protein–protein interactions. The EMBO journal.

[305] Ryu, J. Y., Kim, H. U., & Lee, S. Y. (2018). Deep learning improves prediction of drug–drug and drug–food interactions. Proceedings of the national academy of sciences, 115(18), E4304-E4311.

[306] Luscombe, N. M., Austin, S. E., Berman, H. M., & Thornton, J. M. (2000). An overview of the structures of protein-DNA complexes. Genome biology, 1(1), reviews001-1.

[307] Sanseau, P., Agarwal, P., Barnes, M. R., Pastinen, T., Richards, J. B., Cardon, L. R., & Mooser, V. (2012). Use of genome-wide association studies for drug repositioning. Nature biotechnology, 30(4), 317-320.

[308] Singhal, A., & Sinha, N. (2024). Drug Target Interaction. Askiitians.com. https://www.askiitians.com/iit-jee-chemistry/organic-chemistry/chemistry-in-everyday-life/drug-target-interaction/

[309] Ricci, F., Rokach, L., & Shapira, B. (2010). Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). Boston, MA: springer US.

[310] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6), 734-749.

[311] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems: An introduction–cambridge university press. New York, 352.

[312] Aggarwal, C. C. (2016). Recommender systems (Vol. 1). Cham: Springer International Publishing.

[313] Adomavicius, G., & Tuzhilin, A. (2010). Context-aware recommender systems. In Recommender systems handbook (pp. 217-253). Boston, MA: Springer US.

[314] Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE international conference on data mining (pp. 263-272). Ieee.

[315] Koren, Y., Rendle, S., & Bell, R. (2021). Advances in collaborative filtering. Recommender systems handbook, 91-142.

[316] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.

[317] Areeb, Q. M., Nadeem, M., Sohail, S. S., Imam, R., Doctor, F., Himeur, Y., ... & Amira, A. (2023). Filter bubbles in recommender systems: Fact or fallacy—A systematic review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 13(6), e1512.

[318] Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005, May). Improving recommendation lists through topic diversification. In Proceedings of the 14th international conference on World Wide Web (pp. 22-32).

[319] Adomavicius, G., & Kwon, Y. (2011). Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering, 24(5), 896-911.

[320] Zhang, Y., Callan, J., & Minka, T. (2002, August). Novelty and redundancy detection in adaptive filtering. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 81-88).

[321] Kaminskas, M., & Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Transactions on Interactive Intelligent Systems (TiiS), 7(1), 1-42.

[322] Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010, September). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In Proceedings of the fourth ACM conference on Recommender systems (pp. 257-260).

[323] Kotkov, D., Wang, S., & Veijalainen, J. (2016). A survey of serendipity in recommender systems. Knowledge-Based Systems, 111, 180-192.

[324] Kuanr, M., Mohapatra, P., & Choudhury, S. S. (2020). TSARS: A Tree-Similarity Algorithm-Based Agricultural Recommender System. Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries, 387-400.

[325] Lops, P., De Gemmis, M., & Semeraro, G. (2010). Content-based recommender systems: State of the art and trends. Recommender systems handbook, 73-105.

[326] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009(1), 421425.

[327] Ravulakollu, N. (2023, May 16). Types of recommendation systems: How they work & use cases. AlmaBetter. https://www.almabetter.com/bytes/articles/types-of-recommendation-systems-how-they-work-and-their-use-cases

[328] Burke, R. (2000). Knowledge-based recommender systems. Encyclopedia of library and information systems, 69(Supplement 32), 175-186.

[329] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.

[330] Kawashima, S., & Kanehisa, M. (2000). AAindex: amino acid index database. Nucleic acids research, 28(1), 374-374.

[331] Fout, A., Byrd, J., Shariat, B., & Ben-Hur, A. (2017). Protein interface prediction using graph convolutional networks. Advances in neural information processing systems, 30.

[332] Jiménez, J., Doerr, S., Martínez-Rosell, G., Rose, A. S., & De Fabritiis, G. (2017). DeepSite: protein-binding site predictor using 3D-convolutional neural networks. Bioinformatics, 33(19), 3036-3042.

[333] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017, July). Neural message passing for quantum chemistry. In International conference on machine learning (pp. 1263-1272). Pmlr.

[334] Lipinski, C. A., Lombardo, F., Dominy, B. W., & Feeney, P. J. (1997). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced drug delivery reviews, 23(1-3), 3-25.

[335] Ji, Y., Zhou, Z., Liu, H., & Davuluri, R. V. (2021). DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. Bioinformatics, 37(15), 2112-2120.

[336] Damaševicius, R. (2008, March). Splice site recognition in DNA sequences using k-mer frequency based mapping for support vector machine with power series kernel. In 2008 International Conference on Complex, Intelligent and Software Intensive Systems (pp. 687-692). IEEE.

[337] GTEx Consortium, Ardlie, K. G., Deluca, D. S., Segrè, A. V., Sullivan, T. J., Young, T. R., ... & Dermitzakis, E. T. (2015). The Genotype-Tissue Expression (GTEx) pilot analysis: multitissue gene regulation in humans. Science, 348(6235), 648-660.

[338] Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., & Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nature methods, 5(7), 621-628.

[339] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., ... & Sherlock, G. (2000). Gene ontology: tool for the unification of biology. Nature genetics, 25(1), 25-29.

[340] Jassal, B., Matthews, L., Viteri, G., Gong, C., Lorente, P., Fabregat, A., ... & D'Eustachio, P. (2020). The reactome pathway knowledgebase. Nucleic acids research, 48(D1), D498-D503.

[341] Piñero, J., Ramírez-Anguita, J. M., Saüch-Pitarch, J., Ronzano, F., Centeno, E., Sanz, F., & Furlong, L. I. (2020). The DisGeNET knowledge platform for disease genomics: 2019 update. Nucleic acids research, 48(D1), D845-D855.

[342] Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864).

[343] Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in neural information processing systems, 30.

[344] Zitnik, M., Agrawal, M., & Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics, 34(13), i457-i466.

[345] Ezzat, A., Zhao, P., Wu, M., Li, X. L., & Kwoh, C. K. (2016). Drug-target interaction prediction with graph regularized matrix factorization. IEEE/ACM transactions on computational biology and bioinformatics, 14(3), 646-656.

[346] Zeng, X., Ding, N., Rodríguez-Patón, A., & Zou, Q. (2017). Probability-based collaborative filtering model for predicting gene–disease associations. BMC medical genomics, 10(Suppl 5), 76.

[347] Luo, H., Li, M., Wang, S., Liu, Q., Li, Y., & Wang, J. (2018). Computational drug repositioning using low-rank matrix approximation and randomized algorithms. Bioinformatics, 34(11), 1904-1912.

[348] Ponselvakumar, A. P., Anandamurugan, S., Logeswaran, K., Nivashini, S., Showentharya, S. K., & Jayashree, S. S. (2021, February). Advancement in precision medicine and recommendation system for clinical trials using deep learning methods. In IOP conference series: materials science and engineering (Vol. 1055, No. 1, p. 012110). IOP Publishing.

[349] You, R., Zhang, Z., Xiong, Y., Sun, F., Mamitsuka, H., & Zhu, S. (2018). GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. Bioinformatics, 34(14), 2465-2473.

[350] Kulmanov, M., & Hoehndorf, R. (2020). DeepGOPlus: improved protein function prediction from sequence. Bioinformatics, 36(2), 422-429.

[351] Quince, C., Walker, A. W., Simpson, J. T., Loman, N. J., & Segata, N. (2017). Shotgun metagenomics, from sampling to analysis. Nature biotechnology, 35(9), 833-844.

[352] Soueidan, H., & Nikolski, M. (2015). Machine learning for metagenomics: methods and tools. arXiv preprint arXiv:1510.06621.

[353] Lu, Z., Wilbur, W. J., McEntyre, J. R., Iskhakov, A., & Szilagyi, L. (2009, November). Finding query suggestions for PubMed. In AMIA Annual Symposium Proceedings (Vol. 2009, p. 396).

[354] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

[355] Zheng, X., Ding, H., Mamitsuka, H., & Zhu, S. (2013, August). Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In Proceedings of

the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1025-1033).

[356] Liu, Y., Wu, M., Miao, C., Zhao, P., & Li, X. L. (2016). Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. PLoS computational biology, 12(2), e1004760.

[357] Xu, J., Meng, Y., Peng, L., Cai, L., Tang, X., Liang, Y., ... & Yang, J. (2022). Computational drug repositioning using similarity constrained weight regularization matrix factorization: A case of COVID-19. Journal of Cellular and Molecular Medicine, 26(13), 3772-3782.

[358] Singh, A. P., & Gordon, G. J. (2008, August). Relational learning via collective matrix factorization. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 650-658).

[359] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.

[360] Kabbur, S., Ning, X., & Karypis, G. (2013, August). Fism: factored item similarity models for top-n recommender systems. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 659-667).

[361] Wang, S., Sun, G., & Li, Y. (2020). SVD++ recommendation algorithm based on backtracking. Information, 11(7), 369.

[362] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web (pp. 173-182).

[363] He, X., Du, X., Wang, X., Tian, F., Tang, J., & Chua, T. S. (2018). Outer product-based neural collaborative filtering. arXiv preprint arXiv:1808.03912.

[364] Han, L., Wu, H., Hu, N., & Qu, B. (2019, October). Convolutional neural collaborative filtering with stacked embeddings. In Asian Conference on Machine Learning (pp. 726-741). PMLR.

[365] Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015, May). Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th international conference on World Wide Web (pp. 111-112).

[366] Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016, February). Collaborative denoising auto-encoders for top-n recommender systems. In Proceedings of the ninth ACM international conference on web search and data mining (pp. 153-162).

[367] Öztürk, H., Özgür, A., & Ozkirimli, E. (2018). DeepDTA: deep drug–target binding affinity prediction. Bioinformatics, 34(17), i821-i829.

[368] Karimi, M., Wu, D., Wang, Z., & Shen, Y. (2019). DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. Bioinformatics, 35(18), 3329-3338.

[369] Öztürk, H., Ozkirimli, E., & Özgür, A. (2019). WideDTA: prediction of drug-target binding affinity. arXiv preprint arXiv:1902.04166.

[370] Wan, F., Zhu, Y., Hu, H., Dai, A., Cai, X., Chen, L., ... & Zeng, J. (2019). DeepCPI: a deep learning-based framework for large-scale in silico drug screening. Genomics, proteomics & bioinformatics, 17(5), 478-495.

[371] Abdel-Basset, M., Hawash, H., Elhoseny, M., Chakrabortty, R. K., & Ryan, M. (2020). DeepH-DTA: deep learning for predicting drug-target interactions: a case study of COVID-19 drug repurposing. Ieee Access, 8, 170433-170451.

[372] Huang, K., Xiao, C., Glass, L. M., & Sun, J. (2021). MolTrans: molecular interaction transformer for drug–target interaction prediction. Bioinformatics, 37(6), 830-836.

[373] Luo, Y., Zhao, X., Zhou, J., Yang, J., Zhang, Y., Kuang, W., ... & Zeng, J. (2017). A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. Nature communications, 8(1), 573.

[374] Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S. (2019, July). Neural graph collaborative filtering. In Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval (pp. 165-174).

[375] Wan, F., Hong, L., Xiao, A., Jiang, T., & Zeng, J. (2019). NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. Bioinformatics, 35(1), 104-111.

[376] Torng, W., & Altman, R. B. (2019). Graph convolutional neural networks for predicting drug-target interactions. Journal of chemical information and modeling, 59(10), 4131-4149.

[377] Lim, J., Ryu, S., Park, K., Choe, Y. J., Ham, J., & Kim, W. Y. (2019). Predicting drug-target interaction using 3D structure-embedded graph representations from graph neural networks. arXiv preprint arXiv:1904.08144.

[378] Jiang, M., Li, Z., Zhang, S., Wang, S., Wang, X., Yuan, Q., & Wei, Z. (2020). Drug–target affinity prediction using graph neural network and contact maps. RSC advances, 10(35), 20701-20712.

[379] Nguyen, T., Le, H., Quinn, T. P., Nguyen, T., Le, T. D., & Venkatesh, S. (2021). GraphDTA: predicting drug–target binding affinity with graph neural networks. Bioinformatics, 37(8), 1140-1147.

[380] Son, J., & Kim, D. (2021). Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. PloS one, 16(4), e0249404.

[381] Jha, K., Saha, S., & Singh, H. (2022). Prediction of protein–protein interaction using graph neural networks. Scientific Reports, 12(1), 8360.

[382] Zhang, Y., Wang, Y., Wu, C., Zhan, L., Wang, A., Cheng, C., ... & Li, P. (2024). Drug–target interaction prediction by integrating heterogeneous information with mutual attention network. BMC bioinformatics, 25(1), 361.

[383] Yazdani-Jahromi, M., Yousefi, N., Tayebi, A., Kolanthai, E., Neal, C. J., Seal, S., & Garibay, O. O. (2022). AttentionSiteDTI: an interpretable graph-based model for drug-target interaction prediction using NLP sentence-level relation classification. Briefings in Bioinformatics, 23(4), bbac272.

[384] Bae, H., & Nam, H. (2022). GraphATT-DTA: attention-based novel representation of interaction to predict drug-target binding affinity. Biomedicines, 11(1), 67.

[385] Gönen, M., Khan, S., & Kaski, S. (2013, May). Kernelized Bayesian matrix factorization. In International conference on machine learning (pp. 864-872). PMLR.

[386] Nascimento, A. C., Prudêncio, R. B., & Costa, I. G. (2016). A multiple kernel learning algorithm for drug-target interaction prediction. BMC bioinformatics, 17(1), 46.

[387] Han, S. C., Lim, T., Long, S., Burgstaller, B., & Poon, J. (2021, October). Glocal-k: Global and local kernels for recommender systems. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 3063-3067).

[388] Hofmann, T. (2003, July). Collaborative filtering via gaussian probabilistic latent semantic analysis. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 259-266).

[389] Mnih, A., & Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. Advances in neural information processing systems, 20.

[390] Kim, H., Lu, X., Flaxman, S. R., & Teh, Y. W. (2016). Tucker gaussian process for regression and collaborative filtering. CoRR.

[391] Yang, Y., & Buettner, F. (2021, December). Multi-output gaussian processes for uncertainty-aware recommender systems. In Uncertainty in Artificial Intelligence (pp. 1505-1514). PMLR.

[392] He, T., Heidemeyer, M., Ban, F., Cherkasov, A., & Ester, M. (2017). SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. Journal of cheminformatics, 9(1), 24.

[393] Pal, B., & Jenamani, M. (2018, September). Kernelized probabilistic matrix factorization for collaborative filtering: exploiting projected user and item graph. In Proceedings of the 12th ACM conference on recommender systems (pp. 437-440).

[394] Chu, Y., Kaushik, A. C., Wang, X., Wang, W., Zhang, Y., Shan, X., ... & Wei, D. Q. (2021). DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features. Briefings in Bioinformatics, 22(1), 451-462.

[395] Tisoncik, J. R., Korth, M. J., Simmons, C. P., Farrar, J., Martin, T. R., & Katze, M. G. (2012). Into the eye of the cytokine storm. Microbiology and molecular biology reviews, 76(1), 16-32.

[296] Malumbres, M., & Barbacid, M. (2009). Cell cycle, CDKs and cancer: a changing paradigm. Nature reviews cancer, 9(3), 153-166.

[397] Hauser, A. S., Attwood, M. M., Rask-Andersen, M., Schiöth, H. B., & Gloriam, D. E. (2017). Trends in GPCR drug discovery: new agents, targets and indications. Nature reviews Drug discovery, 16(12), 829-842.

[398] Douek, D. C., Roederer, M., & Koup, R. A. (2009). Emerging concepts in the immunopathogenesis of AIDS. Annual review of medicine, 60(1), 471-484.

[399] Hu, B., Guo, H., Zhou, P., & Shi, Z. L. (2021). Characteristics of SARS-CoV-2 and COVID-19. Nature reviews microbiology, 19(3), 141-154.

[400] Shrotriya, A., Sharma, A. K., Bairwa, A. K., & Manoj, R. (2024). Hybrid Ensemble Learning with CNN and RNN for Multimodal Cotton Plant Disease Detection. IEEE Access.

[401] Zareen, S. S., Sun, G., Kundi, M., Qadri, S. F., & Qadri, S. (2024). Enhancing Skin Cancer Diagnosis with Deep Learning: A Hybrid CNN-RNN Approach. Computers, Materials & Continua, 79(1).

[402] Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., ... & Chintala, S. (2024, April). Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (pp. 929-947).

[403] Abadi, M., & TensorFlow, A. A. B. P. (2016, November). Large-scale machine learning on heterogeneous distributed systems. In Proceedings of the 12th USENIX symposium on operating systems design and implementation (OSDI'16)(Savannah, GA, USA (pp. 265-283).

[404] dvgodoy. (2025). GitHub - dvgodoy/dl-visuals: Over 200 figures and diagrams of the most popular deep learning architectures and layers FREE TO USE in your blog posts, slides, presentations, or papers. GitHub. https://github.com/dvgodoy/dl-visuals/

[405] Rukshan Pramoditha. (2022, January 19). How to Choose the Right Activation Function for Neural Networks | Towards Data Science. Towards Data Science. https://towardsdatascience.com/how-to-choose-the-right-activation-function-for-neural-networks-3941ff0e6f9c/

[406] Weisstein, E. W. (2025). Sigmoid Function. Mathworld.wolfram.com. https://mathworld.wolfram.com/SigmoidFunction.html

[407] Bai, Y., Chen, M., Zhou, P., Zhao, T., Lee, J., Kakade, S., ... & Xiong, C. (2021, July). How important is the train-validation split in meta-learning?. In International Conference on Machine Learning (pp. 543-553). PMLR.

[408] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.

[409] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

[410] Yacouby, R., & Axman, D. (2020, November). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems (pp. 79-91).

[411] Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006, December). Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In Australasian joint conference on artificial intelligence (pp. 1015-1021). Berlin, Heidelberg: Springer Berlin Heidelberg.

[412] Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010, August). The balanced accuracy and its posterior distribution. In 2010 20th international conference on pattern recognition (pp. 3121-3124). IEEE.

[413] J. Cohen, Cohen, J. (1960). A coefficient of agreement for nomial scales. Educational and Psychological Measurement, 20(1), 37–46. doi:10.1177/001316446002000104A coefficient of agreement for nomial scales, Educ Psychol Meas. 20 (1960).

[414] S. Wu, T.W.S. Chow, Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density, Pattern Recognit. 37 (2004). https://doi.org/10.1016/S0031-3203(03)00237-1.

[415] Shin, T. (2021, February 12). Introducing Distance Correlation, a Superior Correlation Metric. | Towards Data Science. Towards Data Science. https://towardsdatascience.com/introducing-distance-correlation-a-superior-correlation-metric-d569dc8900c7/

[416] Chugani, V. (2024, September 5). Understanding Chebyshev Distance: A Comprehensive Guide. Datacamp.com; DataCamp. https://www.datacamp.com/tutorial/chebyshev-distance

[417] Banerjee, S., & Lavie, A. (2005, June). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization (pp. 65-72).

[418] Todeschini, R., & Consonni, V. (2009). Molecular descriptors for chemoinformatics: volume I: alphabetical listing/volume II: appendices, references. John Wiley & Sons.

[419] S. Riniker, G.A. Landrum, Open-source platform to benchmark fingerprints for ligand-based virtual screening, J Cheminform. 5 (2013). https://doi.org/10.1186/1758-2946-5-26.

[420] Chagoyen, M., Carmona-Saez, P., Gil, C., Carazo, J. M., & Pascual-Montano, A. (2006). A literature-based similarity metric for biological processes. BMC bioinformatics, 7(1), 363.

[421] P. Willett, J.M. Barnard, G.M. Downs, Chemical Similarity Searching, J Chem Inf Comput Sci. 38 (1998) 983–996. https://doi.org/10.1021/ci9800211.

[422] Pragmatic Genomics. (2023, April 7). How BLAST E-values are calculated and what they mean. SequenceServer. https://sequenceserver.com/blog/blast-e-value-meaning/

[423] Bartocci, E., & Lió, P. (2016). Computational modeling, formal analysis, and tools for systems biology. PLoS computational biology, 12(1), e1004591.

[424] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezhuk, S. McGinnis, T.L. Madden, NCBI BLAST: a better web interface., Nucleic Acids Res. 36 (2008). https://doi.org/10.1093/nar/gkn201.

[425] G.A. Landrum, RDKit: Open-source cheminformatics. Release 2014.03.1, in: 2014. https://api.semanticscholar.org/CorpusID:61633255.

[426] C. Wei, J. Liang, D. Liu, F. Wang, Contrastive Graph Structure Learning via Information Bottleneck for Recommendation, in: Adv Neural Inf Process Syst, 2022.

[427] A. Bender, J.L. Jenkins, J. Scheiber, S.C.K. Sukuru, M. Glick, J.W. Davies, How Similar Are Similarity Searching Methods? A Principal Component Analysis of Molecular Descriptor Space, J Chem Inf Model. 49 (2009) 108–119. https://doi.org/10.1021/ci800249s.

[428] S.W. Muchmore, D.A. Debe, J.T. Metz, S.P. Brown, Y.C. Martin, P.J. Hajduk, Application of belief theory to similarity data fusion for use in analog searching and lead hopping, J Chem Inf Model. 48 (2008). https://doi.org/10.1021/ci7004498.

[429] Ohio Supercomputer Center (OSC), What is the E Value in BLAST?, (2005). http://www.biocomp.unibo.it/casadio/LMBIOTEC/evalue (accessed October 15, 2022).

[430] G. Stelzer, N. Rosen, I. Plaschkes, S. Zimmerman, M. Twik, S. Fishilevich, T.I. Stein, R. Nudel, I. Lieder, Y. Mazor, S. Kaplan, D. Dahary, D. Warshawsky, Y. Guan-Golan, A. Kohn, N. Rappaport, M. Safran, D. Lancet, The GeneCards suite: From gene data mining to disease genome sequence analyses, Curr Protoc Bioinformatics. 2016 (2016). https://doi.org/10.1002/cpbi.5.

[431] A.I.F. Blakemore, A. Cox, A.-M. Gonzalez, J.K. Maskill, M.E. Hughes, R.M. Wilson, J.D. Ward, G.W. Duff, Interleukin-1 receptor antagonist allele (ILIRN*2) associated with nephropathy in diabetes mellitus, Hum Genet. 97 (1996) 369–374. https://doi.org/10.1007/BF02185776.

[432] I. Aksentijevich, S.L. Masters, P.J. Ferguson, P. Dancey, J. Frenkel, A. van Royen-Kerkhoff, R. Laxer, U. Tedgard, E.W. Cowen, T.-H. Pham, M. Booty, J.D. Estes, N.G. Sandler, N. Plass, D.L. Stone, M.L. Turner, S. Hill, J.A. Butman, R. Schneider, P. Babyn, H.I. El-Shanti, E. Pope, K. Barron, X. Bing, A. Laurence, C.-C.R. Lee, D. Chapelle, G.I. Clarke, K. Ohson, M. Nicholson, M. Gadina, B. Yang, B.D. Korman, P.K. Gregersen, P.M. van Hagen, A.E. Hak, M. Huizing, P. Rahman, D.C. Douek, E.F. Remmers, D.L. Kastner, R. Goldbach-Mansky, An Autoinflammatory Disease with Deficiency of the Interleukin-1–Receptor Antagonist, New England Journal of Medicine. 360 (2009) 2426–2437. https://doi.org/10.1056/NEJMoa0807865.

[433] E. Sjöstedt, W. Zhong, L. Fagerberg, M. Karlsson, N. Mitsios, C. Adori, P. Oksvold, F. Edfors, A. Limiszewska, F. Hikmet, J. Huang, Y. Du, L. Lin, Z. Dong, L. Yang, X. Liu, H. Jiang, X. Xu, J. Wang, H. Yang, L. Bolund, A. Mardinoglu, C. Zhang, K. von Feilitzen, C. Lindskog, F.

Pontén, Y. Luo, T. Hökfelt, M. Uhlén, J. Mulder, An atlas of the protein-coding genes in the human, pig, and mouse brain, Science (1979). 367 (2020). https://doi.org/10.1126/science.aay5947.

[434] Teodoro, UniProt: the universal protein knowledgebase in 2021, Nucleic Acids Res. 49 (2021) D480–D489. https://doi.org/10.1093/nar/gkaa1100.

[435] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-December, 2016. doi:10.1109/CVPR.2016.90.

[436] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, T. Y. Liu, On layer normalization in the transformer architecture, in: 37th International Conference on Machine Learning, ICML 2020, Vol. PartF168147-14, 2020.

[437] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, L. S. Chao, Learning deep transformer models for machine translation, in: ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 2020. doi:10.18653/v1/p19-1176.

[438] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL HLT 2019 – 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, Vol. 1, 2019.

[439] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, R. Apweiler, Intact: An open source molecular interaction database, Nucleic Acids Research 32 (2004). doi:10.1093/nar/gkh052.

[440] N. M. O'Boyle, Towards a universal smiles representation - a standard method to generate canonical smiles based on the inchi, Journal of Cheminformatics 4 (2012). doi:10.1186/1758-2946-4-22.

[441] O. Harrouche, M. B. A. Yamina, Recommender system based on convolutional recurrent deep learning for protein-drug interaction prediction, Expert Systems with Applications 245 (2024). doi:10.1016/j.eswa.2023.123090.

[442] R. Barroso, L. M. Mu˜noz, S. Barrondo, B. Vega, B. L. Holgado, P. Lucas, A. Ba´ıllo, J. Sall´es, J. M. Rodr´ıguez-Frade, M. Mellado, Ebi2 regulates cxcl13-mediated responses by heterodimerization with cxcr5, FASEB Journal 26 (2012). doi:10.1096/fj.12-208876.

[443] A. Rutkowska, I. Preuss, F. Gessier, A. W. Sailer, K. K. Dev, Ebi2 regulates intracellular signaling and migration in human astrocyte, GLIA 63 (2015). doi:10.1002/glia.22757.

[444] L. X. Liu, F. Margottin, S. L. Gall, O. Schwartz, L. Selig, R. Benarous, S. Benichou, Binding of hiv-1 nef to a novel thioesterase enzyme correlates with nef- mediated cd4 down-regulation, Journal of Biological Chemistry 272 (1997). doi:10.1074/jbc.272.21.13779.

[445] M. Lenassi, G. Cagney, M. Liao, T. Vaupotiˇc, K. Bartholomeeusen, Y. Cheng, N. J. Krogan, A. Plemenitaˇs, B. M. Peterlin, Hiv nef is secreted in exosomes and triggers apoptosis in bystander cd4+ t cells, Traffic 11 (2010). doi:10.1111/j.1600-0854.2009.01006.x.

[446] K. L. Collins, B. K. Chen, S. A. Kalams, B. D. Walker, D. Baltimore, Hiv-1 nef protein protects infected primary cells against killing by cytotoxic t lymphocytes, Nature 391 (1998). doi:10.1038/34929.

[447] A. D. Blagoveshchenskaya, L. Thomas, S. F. Feliciangeli, C. H. Hung, G. Thomas, Hiv-1 nef downregulates mhc-i by a pacs-1- and pi3k-regulated arf6 endocytic pathway, Cell 111 (2002). doi:10.1016/S0092-8674(02)01162-5.

[448] J. Mao, J. Wang, A. Zeb, K. H. Cho, H. Jin, J. Kim, O. Lee, Y. Wang, K. T. No, Transformer-based molecular generative model for antiviral drug design, Journal of Chemical Information and Modeling 64 (2024). doi:10.1021/acs.jcim.3c00536.

[449] D. F. Veber, S. R. Johnson, H. Y. Cheng, B. R. Smith, K. W. Ward, K. D. Kopple, Molecular properties that influence the oral bioavailability of drug candidates, Journal of Medicinal Chemistry 45 (2002). doi:10.1021/jm020017n.

[450] F. Lovering, J. Bikker, C. Humblet, Escape from flatland: Increasing saturation as an approach to improving clinical success, Journal of Medicinal Chemistry 52 (2009). doi:10.1021/jm901241e.

[451] T. J. Ritchie, S. J. Macdonald, Physicochemical descriptors of aromatic character and their use in drug discovery (2014). doi:10.1021/jm500515d.

[452] M. D. Shultz, Two decades under the influence of the rule of five and the changing properties of approved oral drugs (2019). doi:10.1021/acs.jmedchem.8b00686.

# Scientific Contributions

Harrouche, O., & Yamina, M. B. A. (2024). Recommender system based on Convolutional Recurrent Deep Learning for protein-drug interaction prediction. Expert Systems with Applications, 245, 123090.

**Under review:** Harrouche, O., & Yamina, M. B. A. (n.d.). Sequential and structural features leveraging attention-based recommender system and translator approaches for advanced drug discovery