Ahmed Marwan Abdulhabeb Alqershi

M.Sc. Thesis

AGU 2025

NEURAL INSIGHTS INTO DRUG REPOSITIONING: A LITERATURE-BASED FRAMEWORK USING WORD EMBEDDINGS AND SIAMESE NETWORKS

M.Sc. THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Ahmed Marwan Abdulhabeb ALQERSHI
May 2025

NEURAL INSIGHTS INTO DRUG REPOSITIONING: A LITERATURE-BASED FRAMEWORK USING WORD EMBEDDINGS AND SIAMESE NETWORKS

A THESIS
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Ahmed Marwan Abdulhabeb ALQERSHI
May 2025

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: Ahmed Marwan Abdulhabeb ALQERSHI
Signature:

REGULATORY COMPLIANCE

M.Sc. thesis titled "**Neural Insights into Drug Repositioning: A Literature-Based Framework Using Word Embeddings and Siamese Networks**" has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Ahmed Marwan Abdulhabeb ALQERSHI	Asst. Prof. Mehmet Gökhan BAKAL
Signature	Signature

Head of the Electrical and Computer Engineering Graduate Program

Asst. Prof. Samet GÜLER

Signature

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled "**Neural Insights into Drug Repositioning: A Literature-Based Framework Using Word Embeddings and Siamese Networks**" and prepared by Ahmed Marwan Abdulhabeb ALQERSHI has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

30 / 05 / 2025

JURY:

Advisor: Asst. Prof. Mehmet Gökhan BAKAL

Member: Assoc. Prof. Rifat KURBAN

Member: Asst. Prof. Burak KOLUKISA

APPROVAL:

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / …….. and numbered ……..…… .

…….. /…….. / ……..

Graduate School Dean
Prof. İrfan ALAN

ABSTRACT

NEURAL INSIGHTS INTO DRUG REPOSITIONING: A LITERATURE-BASED FRAMEWORK USING WORD EMBEDDINGS AND SIAMESE NETWORKS

Ahmed Marwan Abdulhabeb ALQERSHI
MSc. in Electrical and Computer Engineering
Advisor: Asst. Prof. Mehmet Gökhan BAKAL
May 2025

The high cost, long timelines, and risks of traditional drug development have sparked interest in drug repositioning—finding new uses for drugs we already have. My thesis introduces a deep learning tool that digs into biomedical data from SemMedDB to spot potential new treatment connections between drugs and diseases. It's built to be a practical, efficient way to come up with ideas for early drug discovery.

The tool uses a Siamese Neural Network (SNN), trained on word patterns from the FastText model. The study tested two subnetworks—one dense, one convolutional—to see which worked best for pulling out useful features. After running over 570 setups, the best configuration hit a validation accuracy of 87.66% and a test accuracy of about 83%. It performed well across precision, recall, and F1-scores, too.

This work shows how deep learning paired with organized biomedical literature can power smarter drug discovery. It's not perfect yet—relying on one data source, using made-up negative samples, and missing contextual embeddings are some drawbacks. Overall, this system aims to support smarter decisions in drug research and fits into broader efforts for affordable and accessible healthcare.

Keywords: Drug Repositioning, Literature-Based Discovery, Siamese Neural Network, FastText Embeddings, Biomedical Text Mining

ÖZET

İLAÇ YENIDEN KONUMLANDIRMADA NÖRAL İÇGÖRÜLER: KELIME GÖMLEMELERI VE SIYAM SINIR AĞLARI KULLANAN LITERATÜR TABANLI BIR ÇERÇEVE

Ahmed Marwan Abdulhabeb ALQERSHI
Elektrik ve Bilgisayar Mühendisliği Yüksek Lisans Programı
Tez Yöneticisi: Dr. Öğr. Üyesi Mehmet Gökhan BAKAL
Mayıs 2025

Geleneksel ilaç geliştirme süreçlerinin yüksek maliyetleri, uzun zaman çizelgeleri ve riskleri, mevcut ilaçların yeni kullanım alanlarını keşfetmeyi amaçlayan ilaç yeniden konumlandırma çalışmalarına olan ilgiyi artırmıştır. Bu tez, SemMedDB'den elde edilen biyomedikal verileri kullanarak, ilaçlar ile hastalıklar arasındaki potansiyel yeni tedavi bağlantılarını belirlemeye yönelik derin öğrenmeye dayalı bir sistem sunmaktadır. Geliştirilen sistem, erken aşama ilaç keşfi için pratik ve verimli bir fikir üretme yöntemi sağlamayı hedeflemektedir.

Sistem, FastText modelinden türetilen kelime desenlerini kullanarak eğitilen bir Siyam Sinir Ağı (SNN) mimarisine dayanmaktadır. Çalışmada, hangi yapının daha verimli özellikler çıkarabildiğini test etmek için biri yoğun (dense), diğeri evrişimli (convolutional) olan iki farklı alt ağ yapısı denenmiştir. 570'ün üzerinde model yapılandırması test edilmiş ve en iyi konfigürasyon %87.66 doğrulama doğruluğu ve yaklaşık %83 test doğruluğu elde etmiştir. Ayrıca kesinlik, duyarlılık ve F1-skorları açısından da dengeli bir performans sergilemiştir.

Bu çalışma, derin öğrenmenin organize edilmiş biyomedikal literatür ile birleşiminin, daha akıllı ilaç keşif süreçlerine nasıl katkı sağlayabileceğini göstermektedir.

Anahtar kelimeler: İlaç Yeniden Konumlandırma, Literatür Tabanlı Keşif, Siyam Sinir Ağı, FastText Gömülemeleri, Biyomedikal Metin Madenciliği

Acknowledgements

I would like to express my deepest gratitude to my advisor, **Asst. Prof. Mehmet Gökhan BAKAL**, for his invaluable assistance in completing my studies. His patience, insightful guidance, and the brilliant ideas he shared have played a crucial role in shaping my research and academic growth.

I am also profoundly grateful to my parents, **Marwan Alqershi** and **Eqbal Taqi**, for their unwavering emotional support and motivation throughout this journey. My father, a remarkably hardworking student in his time, has always been my role model, inspiring me to strive for excellence and perseverance.

To my sisters, I extend my heartfelt appreciation for always pushing me beyond my limits. Their constant encouragement and belief in my potential have significantly shaped the person I am today.

Finally, and most importantly, I am deeply thankful to my wife, **Maysa Mansour**. She has been my anchor, standing by my side through every challenge. Her unwavering support, encouragement, and incredible ability to help me stay organized have been instrumental in helping me complete this study.

To all of you, I am forever grateful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CBOW	Continuous Bag Of Words
CUI	Concept Unique Identifier
DL	Deep Learning
KG	Knowledge Graph
NLP	Natural Language Processing
SemMedDB	Semantic MEDLINE Database
SG	Skip-Gram
SNN	Siamese Neural Network
WE	Word Embeddings

To My Family

Chapter 1

Introduction

We, humans, have come up with cures for diseases previously held to be fatal, such as smallpox [1], Malaria, and HIV [2]. The existence of these medicines has both provided the safety feeling and contributed significantly to modern human life spans. Part of this feeling of security stems from the fact that drug development is a lengthy and painstaking process that ensures medications are safe for human use and devoid of debilitating adverse effects. Yet, along the way, it has evolved ever more into such a convoluted, labor-intensive and expensive process — and not to mention super unpredictable. The process of bringing a drug from initial research to market takes 10–15 years on average, costing as much as $2.6 billion per drug [3,4]. An additional challenge in drug development is the high failure rate in clinical trials, which can reach over 90% [5,6]. The fact that it was designed to be safe has ironically become a major impediment to the ability to provide timely and effective treatments. This has prompted researchers to consider faster and perhaps less expensive alternatives, and has contributed to increased interest in Drug Repositioning, an approach that offers a more grounded and cost-effective path to address unmet medical needs.

1.1 Background Information

Drug Repositioning, also known as drug repurposing, is the process of discovering new therapeutic applications for existing drugs [7]. It is a relatively recent concept that emerged from serendipitous discoveries, when certain drugs proved effective for conditions they were not originally designed to treat. History proved the effectiveness of this concept through early successful cases that are widely used in our daily life as shown in **Table 1.1**. A well-known example of these successful cases is acetylsalicylic acid, commonly known as aspirin. Originally developed to treat inflammation and fever, Aspirin was later found to have blood-thinning properties, allowing researchers at the

time to study it further and repurpose it to be used as an antiplatelet aggregation drug in small amounts.

In light of that accidental process, drug repositioning has become a strategic tool in medicine and pharmaceutical research due to its financial and time-saving benefits. Since approved drugs have already undergone safety assessments, repositioning them shortens development timelines and reduces associated risks and expenses. This approach offers a noticeably effective path to address unmet medical needs, especially in areas where conventional drug discovery has fallen short. Furthermore, drug repositioning might be able to replace expensive medicines with cheaper and more accessible alternatives for the public in the long term.

Table 1.1 Successful repositioned compounds

Drug	Original Purpose	Repurposed to	Reference
Acetylsalicylic acid *(Aspirin)*	Pain Killer	Cardiovascular Disease Prevention	[7]
Thalidomide	Morning sickness	Multiple myeloma	[7,8]
Sildenafil *(Viagra)*	Angina *(Chest pain)*	Erectile Dysfunction	[7,8]
Dimethyl fumarate	Psoriasis	Multiple Sclerosis	[7]
Atomoxetine	Parkinson's disease	ADHD	[8]
Phentolamine	High Blood Pressure	Dental anesthesia reversal agent	[8]
Mifepristone (RU486)	Pregnancy termination	Cushing's syndrome	[8]
Fasudil	High Blood Pressure	Alzheimer's	[9]

Modern drug repositioning can be achieved using various approaches, mainly experimental, computational, and hybrid methods. **Experimental approaches** directly test drugs against disease models through different methods such as High-Throughput Screening. It is the process of scientists running a lot of tests to examine and understand how current drugs interact with different biological targets. It is much faster and cheaper than the traditional drug discovery process as it takes advantage of the tools and data we already have. [10,11]

Furthermore, this can be even more promising when the information about genetic association with the disease is available. This allows searching for the specific drugs that act on those specific gene targets, giving them a better shot at finding something that works.

Of course, this isn't a silver bullet. Even if a drug shows potential, it still needs thorough testing to make sure the effects are real and not just random results or side effects. But overall, it's a clever way to speed up drug discovery without reinventing the wheel. And as much as this approach worked well, it can still be resource-intensive due to the massive number of tests that need to be applied.

On the other hand, **Computational approaches** have particularly gained significant interest recently because of the noticeable advances in computational power, leading to an Artificial Intelligence breakthrough and an abundance of extensive biomedical data. These approaches use advanced algorithms and data analysis techniques to achieve the drug repositioning target while reducing associated time and cost compared to traditional drug discovery processes.

Under the umbrella of computational drug repositioning, several methodologies exist, with a significant emphasis on data and complex analysis techniques due to their ability to uncover hidden patterns. Amongst these methodologies are:

- **Profile-Based Methods**: These are built upon the different types of drug profiles, such as chemical structure profiles and clinical profiles, to uncover possible drug–disease associations. For instance, the reverse expression profile of a drug can be compared to the dysregulated molecular expression caused by a specific disease,

leading to the identification of candidate drugs with compatible profiles similar to a healthy state [12].

- **Network-Based Methods:** These leverage the interconnected nature of biological systems to identify potential drug-disease associations through network interactions, such as protein-protein, drug-target, and disease-gene associations. By integrating information from these networks, novel potential drugs for diseases can be inferred. For example, a network-based model was used to predict drug-disease associations, validated against clinical trials and databases [12,13].

- **Data-Based Methods:** These methods rely on the integration of diverse datasets, such as gene expression data, chemical structure information, proteomic data, and literature data. By combining these different data types together, researchers can uncover new drug-disease relationships and identify promising candidate drugs for repositioning [12].

See **Figure 1.1** for a summary of the type of computational methods of drug repositioning. These computational approaches play a key role in accelerating the drug discovery process, allowing a more efficient identification of potential drug candidates for a wide range of diseases, ultimately leading to improved patient outcomes.

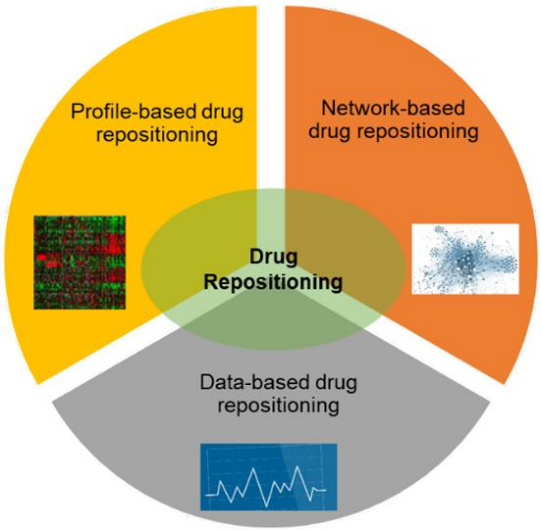

Figure 1.1 Types of Computational Drug Repositioning [12]

1.2 Problem Statement

The implementation of Drug Repositioning, although gaining increasing attention, often encounters obstacles such as missed opportunities in data use and external factors that disrupt research efforts. In this section of the paper, we outline the key barriers that this study is targeting as we believe they are holding progress in the field back. We also explain why addressing these barriers is crucial for adding reliability to the drug discovery process.

- **Data Utilization Gap:**

  One of the main limitations in current drug repositioning practices is the inadequate utilization of the massive information available in biomedical literature. There are millions of papers in the field, resembling millions of different studies, theories, and perhaps even discoveries. These are usually scattered through the journals and conferences. Identifying meaningful patterns and insights from these studies can surely add significant value to Drug Repositioning studies. However, many existing methods do not fully leverage this resource, resulting in missed drug-disease relationships that could lead to new treatment possibilities. [14]

- **Non-Scientific Terminations:**

  RepoDB[1] [15] is a database that records the outcomes of past drug repositioning projects, categorizing them as ***Approved, Suspended, Terminated, or Withdrawn***. The database luckily documents the discontinuation reasons for many incomplete (unapproved) projects, allowing researchers to examine those reasons closely. An important finding from the database is that a good deal of these incomplete projects are stopped due to non-scientific reasons, such as lack of funding, insufficient recruitment, or business decisions, even when the drugs show potential for treating certain conditions. As a result, valuable opportunities to meet medical needs are often lost.

[1] Detailed explanation of this database is explained in the data section below. But here we want to point out the problem with incomplete projects.

According to **Table 1.2**, the most common reason for project termination is "**Low Accrual**", referring to low volunteer participation. Furthermore, all of the top ten reasons for stopping projects are non-scientific. The first scientific reason, "**Toxicity**", ranks only 11th. This suggests that at least **424 projects**, which may still hold therapeutic promise, were discontinued for reasons unrelated to scientific evaluation. These findings highlight the importance of developing robust methods that prioritize scientific value when selecting drug candidates and reduce vulnerability to external, non-scientific influences.

Table 1.2 Primary causes of drug repositioning study termination (RepoDB data)

Reason for termination	# mentioned
NA	720
Low Accrual	66
slow accrual	57
poor accrual	34
Lack of accrual	34
Lack of funding	31
Slow enrollment	28
Low enrollment	22
Insufficient enrollment	19
Sponsor Decision	19
Lack of recruitment	18
Toxicity	17
due to slow accrual	16
Replaced by another study	15
Trial not initiated	15
Lack of Efficacy	14
The study was stopped for feasibility (i.e., low recruitment)	14
Sponsor's decision	12
Trial terminated early per business decision	12
Insufficient accruals	12

Therefore, this study is based on the idea that some drug candidates—previously terminated or suspended for reasons unrelated to science, like enrollment and budget limits—might still hold valuable therapeutic potential, and by using systematic, AI-powered analysis of the massive amounts of biomedical literature and knowledge graphs, we can uncover that potential.

1.3 Significance of the Study

To tackle the problems discussed in the previous section, we aim to develop a robust model that can accurately predict new drug-disease relationships through a literature-driven approach. Moreover, it seeks to revisit RepoDB drug candidates that were previously discontinued for non-scientific reasons and draw conclusions about whether there is still potential that these candidates are valid. Lastly, the study focuses on building a framework that enables researchers to easily explore drug repositioning opportunities, ensuring the results are not only accurate but also accessible and actionable across various domains. It does this by combining biomedical text mining with predictive modelling to form a seamless system. (***Full details of the methodology are provided in Chapter 3***).

In addition to its technical innovations, the study holds practical importance in the following areas:

- **Faster Drug Discovery**: Simplifying the process of drug repositioning speeds up the research in the field as some critical bridges are built.

- **Addressing Overlooked Diseases**: The framework can be utilized to investigate treatments for diseases that often go underfunded. It can take in new data to learn the relationships. This might initiate research on exploring alternatives for expensive or unavailable drugs.

- **Encouraging Collaboration**: The simple design of the framework allows experts from different domains, even those with minimal programming knowledge, to understand it and make use of it to explore exciting possibilities.

Overall, by blending diverse tools and data sources, this study contributes to a smarter and more inclusive approach to drug development—one that researchers across disciplines can use with ease.

1.4 Limitations

This study, like other studies, has several limitations that need to be acknowledged. The following is a detailed list of the study's main limitations.

- **Lack of verified negative data:** The study relies mainly on known positive[2] drug-disease associations obtained from the data sources used. Negative[3] samples, on the other hand, do not exist in the data sources, leading to generating these samples through sampling methods from the positive samples. Therefore, the shortage in structured negative drug-disease associations limits the reliability and accuracy of this study. On the contrary, having access to validated negative data samples would likely improve the model's predictive performance significantly.

- **Dependence on the data source quality:** The accuracy of the predictions is inherently tied to the quality of information available in the literature. Therefore, newly identified diseases have too little data to support reliable predictions. Moreover, feeding misleading information from poor studies to the model deteriorates the performance proportionally.

- **Medical Expert Opinion:** Despite generating useful predictions with high potential, the model still requires expert validation. Therefore, the results obtained from the model mean nothing until a medical expert looks into them and perhaps nominates the candidates for experimental testing.

Despite these constraints, the framework lays a strong foundation for future work. It can be continuously expanded by integrating additional data sources to further advance

[2] A positive drug-disease pair represents a drug that is known to treat a diseases.
[3] A negative drug-disease pair represents a drug that is known to not treat the disease. It might even cause the disease.

drug repositioning efforts. Additionally, it encourages researchers to fill the gaps mentioned to enhance the field even more. (***Details on Future Prospects are outlined in Chapter 5***).

Following this introduction, **Chapter 2** provides a comprehensive literature review, exploring the concepts of drug repositioning, the role of artificial intelligence in this field, and specific techniques such as word embeddings and Siamese neural networks. **Chapter 3** details the materials and methods employed in constructing the proposed framework. **Chapter 4** presents the results and a discussion of their implications. The thesis concludes with **Chapter 5**, offering a summary of findings, conclusions, and suggestions for future research. Finally, the **bibliography** provides a comprehensive list of references cited throughout the work.

Chapter 2

Literature Review

In recent years, advances in computational biology and artificial intelligence have greatly expanded the toolkit for drug repositioning. This chapter provides a comprehensive review of recent peer-reviewed literature on computational drug repositioning. We cover a broad spectrum of techniques, including machine learning models, network-based inference, systems biology approaches, chemical similarity methods, and literature-driven discovery. **Figure 2.1** from recent work [16] schematically illustrates the diverse computational strategies, linking data modalities (chemical structures, -omics signatures, clinical data, etc.) to repurposing methodologies (docking, signature matching, network analysis, knowledge graph inference, real-world evidence mining) and the disease features they address. Emphasis is placed on literature- and knowledge-based methods, given their growing importance in uncovering *hidden* relationships from existing scientific data. We compare these approaches and discuss their algorithmic foundations, performance, and integration. Eventually, by the end of this chapter, the reader will have a better understanding of the computational methodologies enabling drug repurposing, their relative strengths, and how they complement each other toward the common goal of efficient therapy discovery.

2.1 Machine Learning and AI-Based Approaches

Machine learning (ML) has become a cornerstone of computational drug repositioning in recent years. Modern ML models can ingest large-scale biomedical data – from chemical structures and genomic profiles to clinical records – and learn patterns that suggest new drug–disease connections. **Supervised learning** approaches treat known drug–indication pairs as training data and build classifiers or predictors to score new pairs. For example, standard classification algorithms (support vector machines, random forests, logistic regression, etc.) have been used to predict if a given drug might treat a

given disease based on features like targets, pathways, or phenotypic effects. These conventional models often face challenges with *bias* (overfitting to well-studied drugs) and *class imbalance* (far more negative examples than positives), but they established the feasibility of ML-based repurposing.

Matrix factorization and recommender systems techniques have also been applied, casting drug–disease associations in a matrix form. One notable example by **Bakal et al.** [17] employs Non-Negative Matrix Factorization (NMF) on a gold-standard repositioning dataset (**repoDB**).

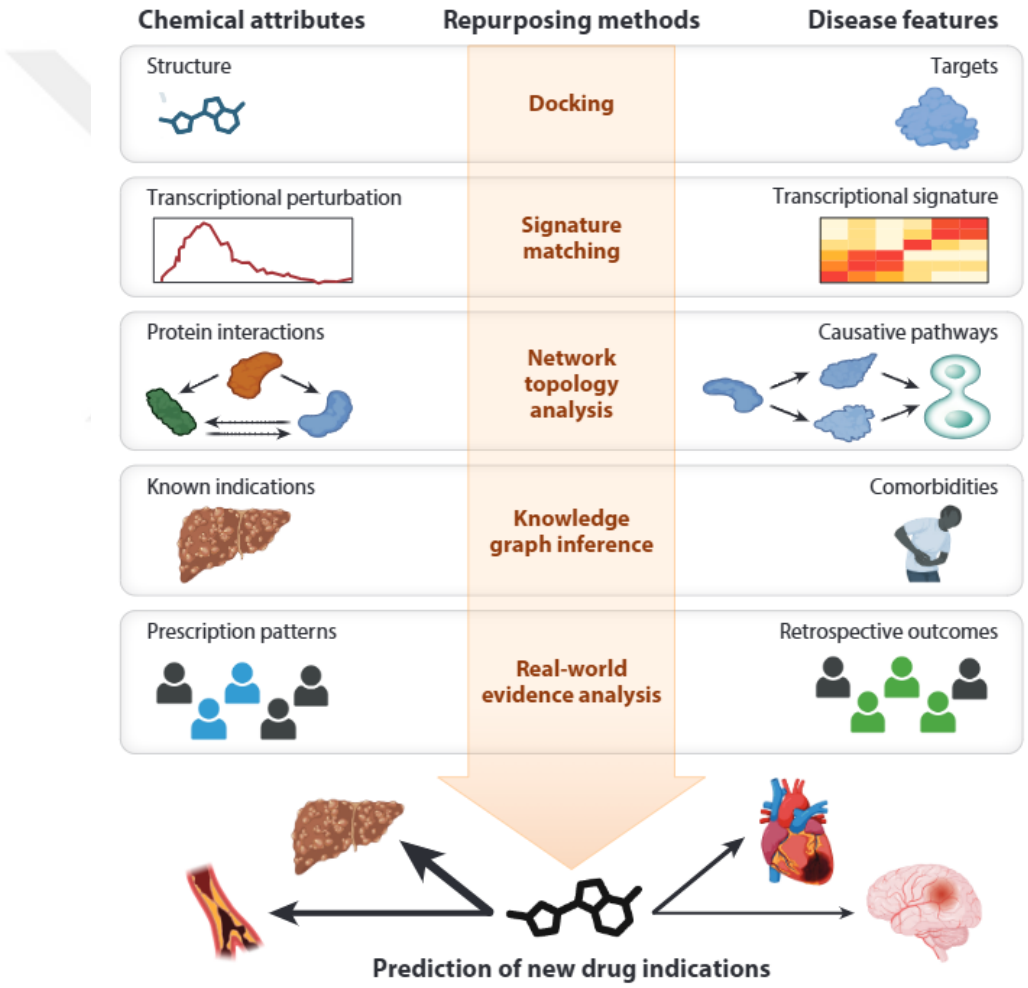

Figure 2.1 Computational Drug Repositioning Approaches Overview [16]

The database contains known approved drug–disease pairs as positives and failed trial pairs as realistic negatives, providing a balanced benchmark. The authors factorized a drug–disease interaction matrix (augmented with curated biomedical knowledge) into

latent features, demonstrating that NMF could recover 96% of known approvals (recall) at 80% precision. This high performance (96% recall of approved indications) suggests that hand-curated knowledge bases combined with matrix-completion algorithms are powerful for hypothesis generation in drug repurposing. In a thesis, one might include an equation defining the NMF objective (e.g. **minimizing $\|X - WH\|$ under non-negativity**) to illustrate how the model learns latent therapeutically relevant features. NMF and related **collaborative filtering** methods treat repositioning like a recommendation problem (*recommend diseases for a drug*), effectively leveraging patterns of polypharmacology and comorbidity.

In a related but more specialized area of study, deep learning (DL) has emerged as a powerful extension of traditional machine learning, offering enhanced performance and new capabilities. Deep Neural Networks can capture complex, non-linear relationships unachievable by simpler models. A recent review by Urbina *et al.* (2021) [18] highlights multiple cases where deep learning–based repurposing yielded promising leads in oncology, neurology, and antiviral research. Notably, DL models have identified kinase inhibitors with unexpected cancer indications, suggested repurposed candidates for Alzheimer's disease, and rapidly screened molecules for COVID-19 treatments. These successes illustrate the versatility of ML: models can be trained on gene expression signatures, protein–protein interactions, clinical phenotypes, or integrative feature sets.

However, Urbina *et al.* also caution against "repurposing hype" and emphasize rigorous validation – many ML predictions still need experimental confirmation. **Figure 2.2** by the authors illustrates a generic deep learning pipeline for drug repositioning – from data curation and feature extraction to model training and candidate prioritization.

Overall, machine learning and AI-driven approaches form a foundational pillar of computational drug repositioning, **enabling the screening of vast chemical and biomedical spaces that would be infeasible manually**. As data grows, these models are expected to improve, especially with continual learning and integration of heterogeneous data sources.

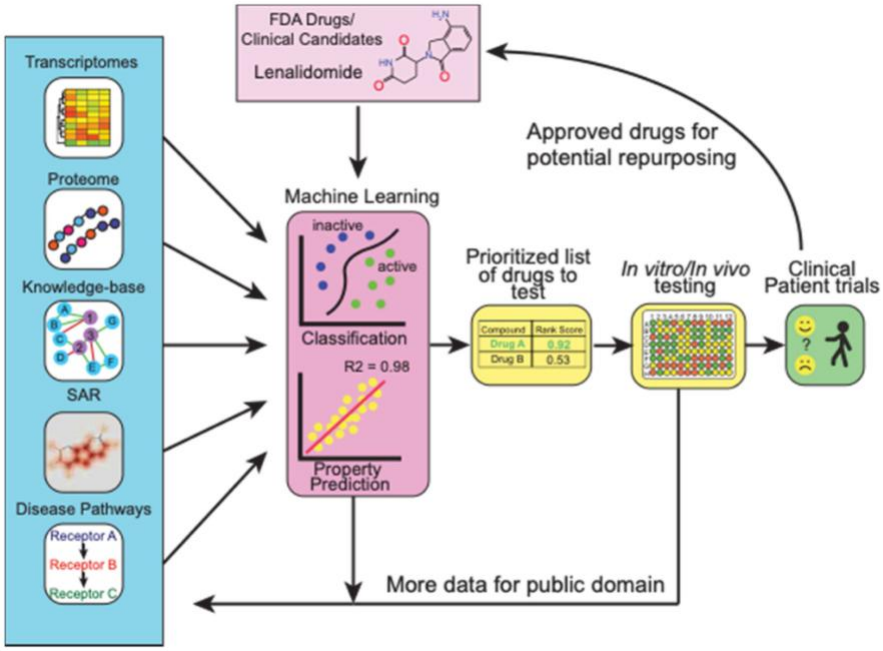

Figure 2.2 DL pipeline for CDR [18]

2.2 Network-Based and Graph Inference Approaches

Network-based methods leverage the rich web of interactions among biological entities (genes, proteins, diseases, drugs, side effects) to uncover new drug uses. The central premise is that in the **network of biomedicine**, drugs and diseases that are closely connected (or share neighbours) may have a treatment relationship. These approaches draw from the field of *network medicine*, which posits that diseases correspond to perturbations in cellular networks and that drugs can be understood by how they "rewire" these networks [19].

(*Cheng et al., 2018*) [20] introduced a **systems pharmacology framework** grounded in *network proximity* to systematically identify novel drug repurposing opportunities by analyzing the spatial relationships between drug targets and disease genes in the **human protein–protein interaction (PPI) network**. To ensure comprehensive coverage and accuracy, they curated a *high-confidence interactome* composed of 243,603 interactions among 16,677 proteins, derived from five experimental data sources: *yeast-two-hybrid (Y2H) assays*, *3D protein structure-based PPIs*, *kinase–*

substrate interactions, *signaling networks*, and *AP-MS derived literature-curated interactions*. Their core proximity metric involved computing the **shortest path-based z-score** between a drug's target set and a disease's protein module. A significant *negative z-score* indicated that drug targets lie within or near the disease module, thus implying therapeutic or adverse effects.

This method was applied to **984 FDA-approved drugs** and **23 cardiovascular (CV) diseases**, resulting in an **atlas of high-confidence drug–disease predictions**. As shown in **Figure 2.3**, this network connected **431 non-CV drugs** to **22 CV disease modules**, where edges represent interaction strength based on the z-score proximity and node colors reflect ATC classifications. Four predictions with strong network signals were selected for empirical validation, including the associations between **carbamazepine and increased coronary artery disease (CAD) risk** and **hydroxychloroquine and decreased CAD risk**.

To validate these predictions, the authors conducted **large-scale pharmacoepidemiologic studies** using healthcare databases encompassing over **220 million patients**, leveraging *propensity score-matched cohort analyses* to assess real-world effects. Two of the four predictions—**carbamazepine-CAD** and **hydroxychloroquine-CAD**—were statistically confirmed with hazard ratios of 1.56 and 0.76, respectively. Furthermore, *in vitro* experiments demonstrated that hydroxychloroquine attenuated TNF-α-induced endothelial inflammation, offering **mechanistic support** for its protective role. Collectively, this study exemplifies how integrating **interactome topology**, **population-scale patient data**, and **wet-lab assays** yields a powerful pipeline for in silico drug repurposing and hypothesis-driven translational validation.

Another emerging frontier in network-based drug repositioning lies in the application of **Graph Neural Networks (GNNs)**, which enables the joint learning of **node-level features** and **topological structures** in biomedical graphs. Sun et al. [21] proposed a novel adaptive graph convolutional network architecture, termed **AdaDR**, designed to overcome the limitations of conventional GCNs that treat node features and network structure independently. AdaDR introduces a dual-path embedding mechanism, where **drug and disease representations** are learned both from feature-based k-nearest neighbour (kNN) graphs and known drug–disease association graphs. These embeddings are then **fused using an attention mechanism** that adaptively assigns weights based on

learned importance, while a **consistency constraint** enforces semantic alignment between feature and topology spaces.

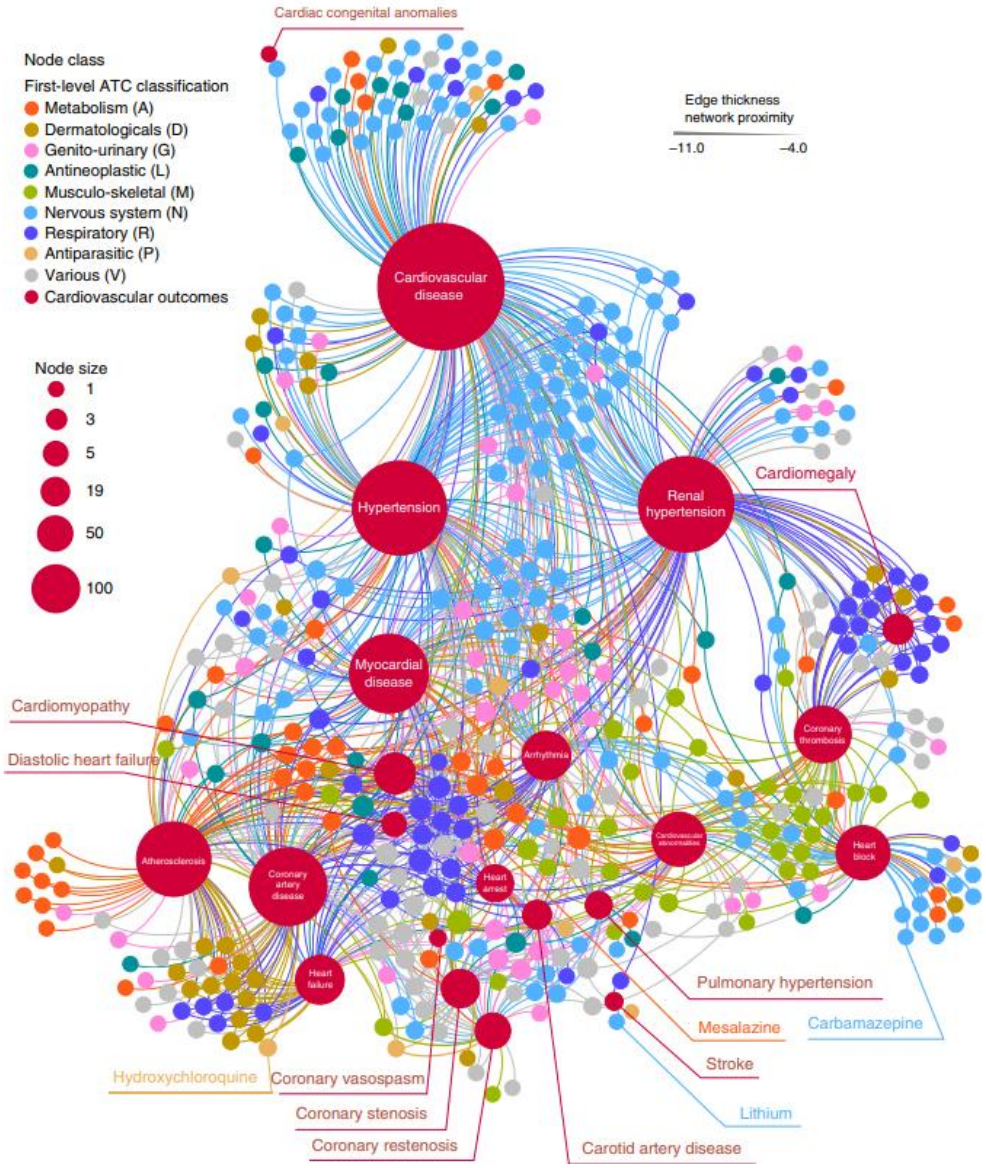

Figure 2.3 Network of predicted drug–disease associations connecting 431 FDA-approved non-cardiac drugs to 22 cardiovascular disease modules. [20]

The model architecture comprises three main modules: (i) a **graph convolution module** that computes embeddings from both drug–drug/disease–disease similarity graphs and the bipartite drug–disease interaction graph, (ii) an **adaptive attention**

module to merge these embeddings, and (iii) a **prediction module** employing a multi-layer perceptron (MLP) for final association scoring. The architecture is visualized in **Figure 2.4**, which outlines the two-path convolution process, attention fusion, and embedding concatenation for prediction.

To evaluate AdaDR, the authors utilized **four benchmark datasets**: Gdataset (593 drugs, 313 diseases, 1933 associations), Cdataset (663 drugs, 409 diseases, 2532 associations), Ldataset (269 drugs, 598 diseases, 18,416 associations), and LRSSL (763 drugs, 681 diseases, 3051 associations). Drug–drug similarities were computed via 2D chemical fingerprints, and disease similarities were based on semantic phenotype annotations. The model achieved **an average AUROC of 0.937** and **AUPRC of 0.576** across the datasets, outperforming seven baseline methods, including **DRHGCN, MBiRW, iDrug**, and **BNNR**. On individual datasets, AdaDR outperformed the best GCN-based model (DRHGCN) by **up to 9.8% in AUPRC** on Gdataset, demonstrating its superior ability to integrate heterogeneous data types.

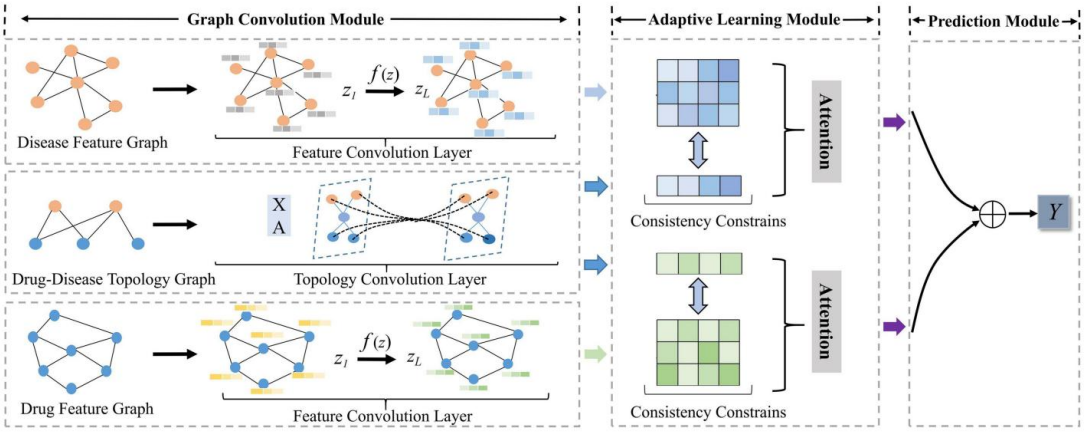

Figure 2.4 Architecture of AdaDR model showing the pipeline [21]

Furthermore, the model showed robustness in **cold-start settings**, predicting indications for unseen drugs with an **AUROC of 0.948** and **AUPRC of 0.393** on Gdataset. In **ablation studies**, removing either the attention mechanism, feature embeddings, or consistency constraint led to performance drops of 2–8%, affirming the contribution of each component. The **attention analysis** further revealed dataset-dependent weighting

between topology and features, with **topological embeddings dominating drug representations** in sparse datasets like LRSSL.

Finally, two **case studies** on Alzheimer's disease and breast carcinoma demonstrated the model's practical utility. Among the top five predicted drugs for each disease, **100% were supported by PubMed or CTD evidence**, underscoring AdaDR's potential for actionable drug discovery. Additionally, **gene ontology enrichment analysis** of predicted Alzheimer's drugs revealed pathways such as *monoamine transport* and *dopamine uptake*, aligning with known disease mechanisms.

In summary, network-based approaches exploit the interconnected nature of biological data. They excel at uncovering relationships that are *not obvious from one-dimensional analysis* – for example, a drug might not share any targets with a disease directly, but network analysis finds it modulates a pathway that intersects the disease's pathway. A key opportunity (and challenge) here is integrating multi-scale networks (molecular, cellular, phenotypic, and even social networks) to capture the full context of drug effects. This will require not just algorithm development but also careful validation; network predictions must ultimately be verified in biological models, as spurious connections or incomplete networks can mislead. Nonetheless, network and graph-based inference have become indispensable tools in computational drug repositioning.

Chapter 3

Materials & Methods

The development of this framework surely requires the integration of various resources and tools, all employed using an organized methodology that puts everything in place to obtain the best performance out of the study. This chapter presents the materials, along with the methodology employed to achieve a literature-driven drug repositioning system powered by Artificial Intelligence. Looking at the study from a high level, the approach integrates Natural Language Processing to understand biomedical literature, and a Neural Network to identify potential associations. Specifically, the approach employs a systematic pipeline comprising data preprocessing, word embeddings, Siamese neural network model construction, validation, and prediction generation. *All implementation details—including data-pre-processing scripts, training notebooks, and inference code—are openly available in the project's GitHub repository* ***here***. **Figure 3.1** illustrates a generic pipeline of this study's approach.

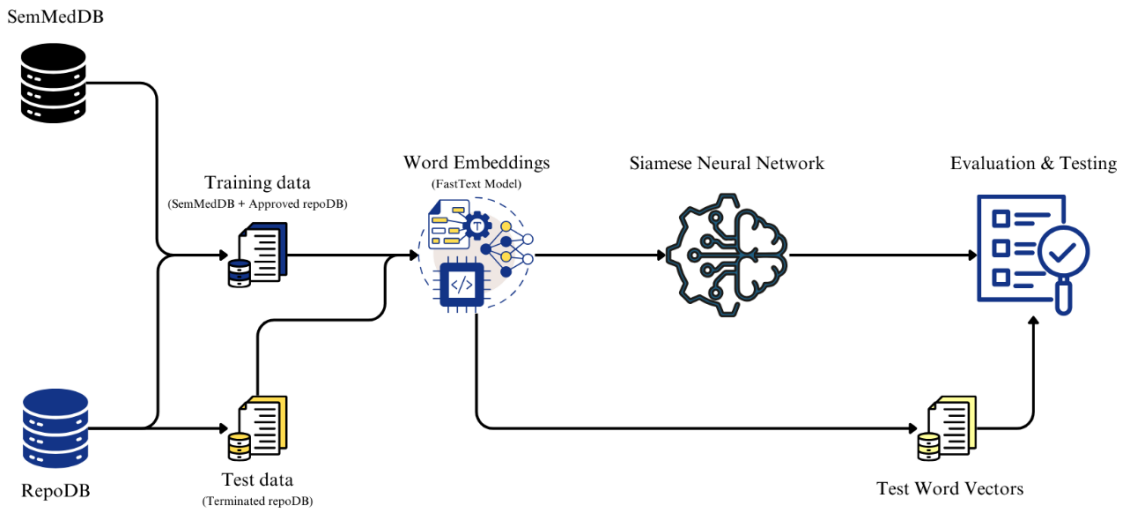

Figure 3.1 General workflow of the study

3.1 Datasets and Tools

This study utilizes two data sources to achieve a reliable performance; **SemMedDB** and **repoDB**. This subsection deeply dives into both databases, expresses their importance, and shows how they add reliability to the study.

3.1.1 The Semantic Medline Database (SemMedDB)

SemMedDB[4] serves as a rich knowledge repository, storing (*subject, predicate, object*) relationships extracted from the massive biomedical literature that exists in PubMed [22]. The developers specifically extracted the information from the papers' titles and abstracts using SemRep, a rule-based natural language processing (NLP) tool, to identify "semantic assertions" within biomedical texts. SemRep achieves this by transforming textual mentions of entities (subjects and objects) into standardized UMLS **(Unified Medical Language System)** Metathesaurus concepts[5] represented as CUIs **(Concept Unique Identifiers)**, while linking predicates to the UMLS semantic network [23]. For example, all different names of Aspirin (listed in **Table 3.1**) are assigned the CUI **C0004057** in the UMLS Metathesaurus, unifying all names across biomedical literature using this code.

Furthermore, each subject and object in the database is accompanied by its semantic type, drawn from the UMLS Semantic Network. These semantic types define the broad classes of subjects and objects. For example, "**phsu**" represents a Pharmacologic Substance, and "**topp**" represents Therapeutic or Preventive Procedure, while "**dsyn**" represents Disease or Syndrome, and "**neop**" represents Neoplastic Process (cancers). The information semantic types convey is essential to clarify the nature of the entities in each triple as well as their relationships. Therefore, it is logical to observe an entry like: A subject with "phsu" semtype (e.g., a drug) treats an object with semtype "dsyn" (e.g., a condition). This classification enhances the database's ability to represent

[4] The version used in this study was released in September 2023 and contains 126,268,045 records
[5] The UMLS Metathesaurus is a comprehensive dictionary that unifies medical terms from various sources, assigning each a unique code called a Concept Unique Identifier (CUI) to standardize them across texts.

meaningful biomedical interactions and supports later analysis of drug-disease connections in this study.

To refine the dataset and ensure the use of only relevant information to the study, only 11 predicate types were taken into consideration; treats, coexists_with, stimulates, prevents, disrupts, augments, interacts_with, uses, produces, associated_with, and inhibits. Focusing on these treatment-relevant relationships intends to support the identification of drug-disease connections.

Later on, only unique triples (subject CUI, predicate, object CUI) were kept, along with the number of times these triples were mentioned in PubMed. This is used to prioritize more dependable triples; we set a frequency threshold of 50 or higher, reducing the dataset to 64,972 entries. In simple words, only triples that were mentioned at least 50 times across PubMed are considered, giving the data more reliability.

Table 3.1 UMLS names for CUI C0004057

Aspirin	Ecotrin	Entericin
aspirin	Endosprin	St. Joseph
Acetylsalicylic Acid	Magnecyl	Measurin
Acetylsalicylic acid	Micristin	ACIDE ACETYLSALICYLIQUE
acetylsalicylic acid	Polopiryna	ACETYLSALICYLIQUE, ACIDE
Acid, Acetylsalicylic	Zorprin	ASPIRINE
Acetysal	2-(Acetyloxy)benzoic Acid	ASPIRIN
Acylpyrin	Benzoic acid, 2-(acetyloxy)-	ASPIRINA
Colfarit	Aspergum	ACIDO ACETILSALICILICO
Easprin	Empirin	

3.1.2 Drug Repositioning Database (repoDB)

Developed by Brown and Patel (2017) [15], repoDB is a carefully curated gold-standard resource designed to assess computational drug repositioning (CDR) techniques.

It stands out by integrating approved drug indications from DrugCentral with failed indications from the American Association of Clinical Trials Database, sourced from NLM's ClinicalTrials.gov. This blend of successful and unsuccessful indications makes repoDB a strong benchmark for testing CDR method accuracy and reliability.

The repoDB dataset includes 11,921 entries, categorized into four different status of drug repositioning studies: approved (8,931), suspended (127), terminated (3,392), and withdrawn (1,108). While disease CUIs are provided, drug CUIs are not included. To resolve this, we obtained drug CUIs using the unique drug identifiers from repoDB via the UMLS API. After removing duplicates and ensuring all triples (subject CUI, predicate, object CUI) were distinct, we added "approved" entries to the positive dataset with the predicate "treats," provided they weren't already present. Entries labelled as suspended, terminated, or withdrawn were assigned to the test dataset, ensuring no overlap with positive instances. Including these unapproved entries in the test set helps evaluate the model's ability to detect potentially effective drugs. **Table 3.2** presents the distribution of positive examples across relationship types, incorporating repoDB data. It can be seen that "treats," "uses," and "coexists_with" account for a significant share of the positive dataset, highlighting their common occurrence in therapeutic associations within SemMedDB.

Table 3.2 Distribution of Data Triples by Predicate Type

Predicate Type	Number of Instances
Treats	34,736
Uses	12,364
Coexists_with	6,754
Interacts_with	4,681
Associated_with	4,625
Stimulates	2,541
Produces	1,643
Augments	1,501
Inhibits	1,247
Prevents	1,118
Disrupts	983

3.1.3 Tools used

Various tools were utilized to carry out this study. To start with, we used the Python programming language to execute this study due to its extensive ecosystem of libraries for data processing, machine learning, and visualization. Especially in the field of Artificial Intelligence, Python has proven itself to be the go-to programming language and encouraged more researchers, developers and companies to use it and even enrich it with more libraries. In this study, we leveraged various libraries that significantly simplified this study; some of the key libraries used are:

- **Pandas:** A versatile data analysis library that enables efficient data manipulation throughout the pipeline. It was used for tasks like creating, filtering, and merging datasets.

- **NLTK:** The *Natural Language Toolkit* offers essential text preprocessing features such as tokenization and linguistic structuring. These capabilities prepared the textual data before generating embeddings.

- **Gensim:** Known for vector space modeling, Gensim's **FastText** implementation was used to create word embeddings from the biomedical text from SemMedDb. These embeddings converted drug and disease terms into semantic vector representations. In simple terms, this library turns each word into a number vector that the code understands.

- **NumPy:** A fundamental package for numerical computing that supports the manipulation of arrays and matrices. It played a key role in handling Word Embeddings vectors, and matrix operations during model development.

- **TensorFlow:** A powerful open-source framework for developing and deploying deep learning models. It was used to build the Siamese Neural Network, the core of this study. Model training, evaluation, and deployment are also done using this library.

- **Scikit-Learn:** Another Machine Learning library with a bunch of utility classes and functions that aid with the development and training process. In our study, Scikit-Learn provided critical tools to use in data splitting, cross-validation, and evaluation metrics.

- **Matplotlib & Seaborn:** We used these visualization libraries to create informative plots for model evaluation. They enabled us to express our model's quality through generating training curves, confusion matrices, ROC curves, and precision-recall graphs.

The mentioned libraries were crucial to execute the study efficiently; though, it could also be executed using some alternative tools, such as **PyTorch**, a well-known alternative to TensorFlow for building and training deep neural networks. Furthermore, some other libraries were employed to execute the study, but are not as critical. The full list of libraries used in this study can be found in the *requirements.txt* file in the GitHub repository.

Now, after providing a detailed overview of the datasets and tools used in the study, this section describes the structured approach the study follows to achieve its computational drug repositioning goal. The methodology follows a pipeline that comprises four primary stages: data preprocessing, word embedding generation, AI model development, and building the framework. We deep dive into each of these stages below.

3.2 Data Pre-processing

Effective processing of the data is essential to ensure the input is correctly formatted and aligned with the tool's requirements and to get better performance out of the models. As discussed in the previous section, this study employs SemMedDB and RepoDB and combines them as needed. After both data sources were combined into one data frame with a little over 72,000 entries. A random sample from the current state of

the data is shown in **Table 3.3**. This data frame was then processed through the following pipeline:

3.2.1 Extract drug-disease pairs

The process begins by extracting *positive* drug-disease pairs from the SemMedDB dataset, i.e. pairs where the drug is known to treat the disease. These pairs are selected based on subjects' and objects' semantic types information given: the subject must be classified as a pharmacologic substance ("**phsu***"* in the database**)** and the object as a disease syndrome (*"***dsyn***"* in the database). This ensures that the selected relationships represent legitimate drug treatments for diseases.

Table 3.3 Random sample from the initial data

SUBJECT CUI	PREDICATE	OBJECT CUI	SUBJECT SEMTYPE	OBJECT SEMTYPE
C0278252	COEXISTS_WITH	C0678236	fndg	dsyn
C0237123	COEXISTS_WITH	C5203670	mobd	dsyn
C1257954	TREATS	C0034693	orch,phsu	mamm
C0008339	TREATS	C0030705	topp	humn,podg
C0029005	ASSOCIATED_WITH	C0006142	aapp,gngm	neop
C0010412	TREATS	C0376358	topp	neop
C0021641	AUGMENTS	C0178666	aapp,phsu	celf
C0013089	TREATS	C0026809	antb,orch	mamm
C0087111	TREATS	C1510586	topp	mobd
C0013227	TREATS	C0149925	phsu	neop

3.2.2 Negative Instance Generation

Up until this point, we have only positive data for our model. But to build a robust framework that can capture hidden associations, it is crucial to include high-quality negative instances to enhance the model's ability to distinguish irrelevant biomedical

relationships. Bad news is that there is no database dedicated to such negative associations as discussed in the **Limitations Section**. Therefore, a tailored negative sampling approach was taken. This was achieved by writing a Python script that studies the positive samples, considering the predicate types, and samples negative associations based on that. Leveraging domain/range constraints from the UMLS semantic network, as defined by NLM in the SRSTR tables [24] within UMLS [25], this approach selects entity pairs that align with the constraints for our chosen relationships [19], producing unique negative examples for positive instances. For this, two approaches were taken; one with balanced positive-negative data, and another one is unbalanced with negative data samples were generated to be around double the number of positive samples. This custom approach was explicitly used to uncover potential new drug-disease pairs and to mimic real life more as far more negative samples exist than positive samples.

3.3.3 Tokenize triples for Word Embeddings

Along with the positive pairs and the negative pairs generated, additional relationship types are also extracted from SemMedD, such as those associated with proteins and genes. Including these varied biomedical associations helps expand the semantic context in which the drugs and diseases appear, allowing the model to better understand the system.

Therefore, all relationships are converted into textual format for training the word embedding model. Positive pairs retain their original relationship predicates, while negative samples are explicitly phrased as "**drug neg_treat disease**". This clear distinction helps the model learn to differentiate between true and false associations.

Finally, these phrases are tokenized into smaller chunks/words using NLTK as discussed in the **Tools Subsection**.

At this point in the study, we have a list of **164,888** elements that include all the information extracted from the datasets, along with the generated negative samples. At this stage, the data is prepared to be input into the word embedding model, which will, in turn, utilize it to capture the underlying structure of the system.

3.3 Word Embedding Model

In the context of literature-based drug repositioning, an essential step is the transformation of textual biomedical concepts into dense numerical vectors. This is because representing biomedical concepts through fixed-size, dense vector embeddings offers a solution by capturing semantic and syntactic patterns in continuous vector space. These representations are more suitable than sparse one-hot encodings for feeding into deep learning models, especially those sensitive to vector similarity, such as Siamese architectures.

This section provides an overview of the word embedding models evaluated, the reasons behind selecting FastText over Word2Vec, and the justification for choosing the Continuous Bag of Words (CBOW) over Skip-gram architecture within FastText. Finally, we outline the chosen model's configurations and show an example of the transformation that occurred to the data.

3.3.1 Comparative Analysis: Word2Vec vs. FastText

Two widely adopted algorithms were evaluated for embedding biomedical vocabulary: **Word2Vec** and **FastText**. Both frameworks learn vector representations by leveraging distributional semantics, but differ in their treatment of word structure. **Word2Vec**, introduced by Mikolov et al. [26], employs shallow neural networks to learn embeddings using either the Continuous Bag of Words (CBOW) or Skip-gram architecture (*Both architectures will be explained in upcoming subsections).* Word2Vec has been adopted in numerous studies in the field and has shown effectiveness in capturing contextual relationships. However, the algorithm treats each word as an atomic unit, meaning it does not break it down into smaller, meaningful parts, limiting its ability to generalize to out-of-vocabulary (OOV) or morphologically complex (e.g. words with prefixes or suffixes). This limitation is extremely critical for biomedical studies, as new terms frequently emerge and often share subword components.

To tackle this limitation**, FastText**, developed by Bojanowski et al. [27], was explored. It extends Word2Vec by incorporating character-level information via subword n-grams, where each word is represented as a sum of its constituent n-gram vectors.
In simple terms, each biomedical term $\boldsymbol{w}$ (such as a drug name or a disease) is not represented as a single unit or "token" in FastText. Instead, FastText breaks the word into smaller overlapping character chunks called ***n-grams*** as illustrated in **Figure 3.2**.

Let's denote the full set of these n-grams for a word $\boldsymbol{w}$ as $\boldsymbol{G(w)}$. The idea is that instead of learning a vector for the whole word, we learn vectors for these smaller n-grams. Then, to get the **final embedding** of the word $\boldsymbol{w}$, we compute the **average** of all the n-gram vectors:

$$\vec{w} = \frac{1}{|G(w)|} \sum_{g \in G(w)} \vec{g} \tag{3.1}$$

Where:

- $\vec{w}$ is the resulting word embedding for the word $\boldsymbol{w}$,
- $\boldsymbol{G(w)}$ is the set of n-grams extracted from $\boldsymbol{w}$,
- $|\boldsymbol{G(w)}|$ is the number of n-grams in the set,
- $\vec{g}$ is the embedding (vector) of each n-gram $\boldsymbol{g}$.

With this strategy, the model is able to generate embeddings for unseen or rare words, making it particularly advantageous for biomedical language.

Empirical comparisons on our dataset favoured FastText over Word2Vec in handling unseen biomedical terminology while maintaining robust semantic coherence across frequent tokens. Accordingly, FastText was selected as the embedding model for this study.

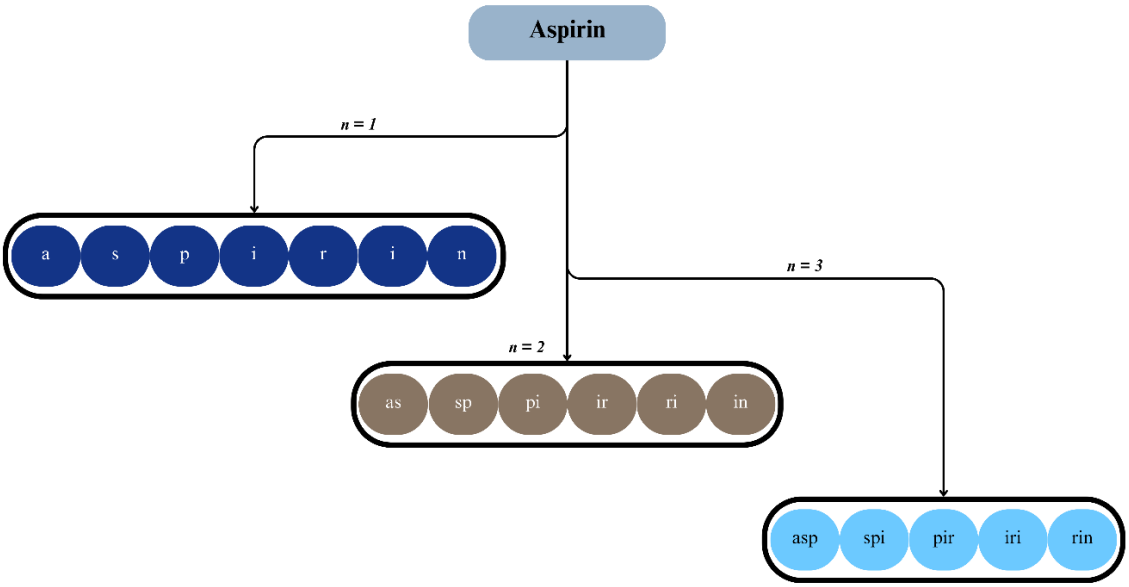

Figure 3.2 n_gram representation of FastText

3.3.2 Training Architecture: CBOW vs. Skip-gram

FastText offers two training paradigms: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts the target word based on its surrounding context, making it computationally efficient and well-suited for high-frequency vocabulary. Conversely, Skip-gram predicts the context words given a target word, and has been shown to perform better for infrequent or domain-specific terms— typical characteristics in biomedical texts. **Figure 3.3** illustrates the difference between the two architectures using a simple example.

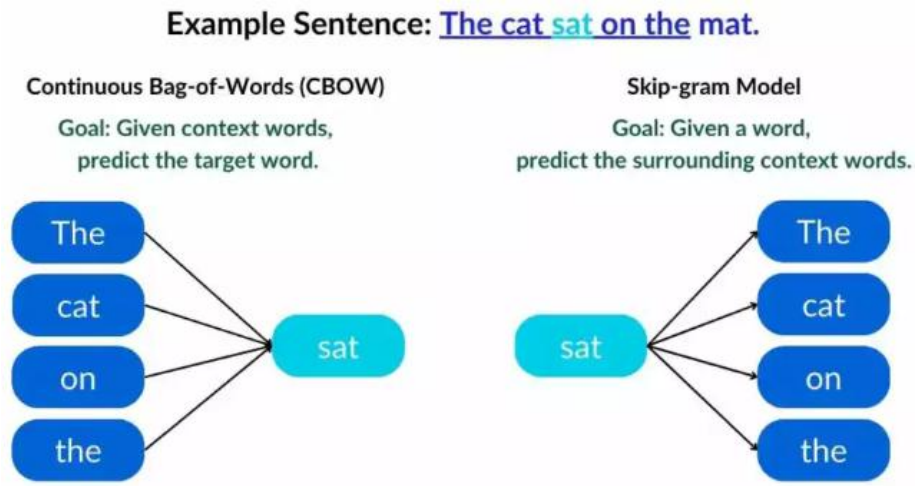

Figure 3.3 Difference between CBOW and Skip-gram [28]

In this study, we believe that the **CBOW architecture would be more effective**, based on the observation that our focus is not on individual biomedical terms alone, but rather on understanding the **predicate**, specifically whether it indicates a treatment relationship (e.g., "*TREAT*" vs. "*NEG_TREAT*") between a drug and a disease. Since CBOW is designed to predict a word from its context, it aligns well with this goal: it uses the surrounding words (such as subject and object) to infer the treatment relationship. This makes CBOW conceptually well-suited for capturing how a treatment relationship is expressed in the biomedical statements.

3.3.3 Implementation Details

FastText was trained on the corpus from SemMedDB and repoDB after processing it as shown in the previous section. The model parameters will be optimized through hyperparameter tuning and stratified cross-validation **(as shown later in this study)** on downstream prediction accuracy within the Siamese network pipeline.

This approach enhances the model's robustness to spelling variations and unseen terminology. For instance, embeddings for *glioma* and *glioblastoma* demonstrate proximity in the learned space due to shared subwords, which would not be captured using a purely token-based model.

After training the FastText model on the data, we convert all the drugs and diseases into their respective word vectors, transforming them into a language that AI models can understand efficiently.

The final embeddings generated by FastText are used to represent drug and disease terms in the form of fixed-length vectors $\vec{\boldsymbol{d}}$ and $\vec{\boldsymbol{s}}$, respectively. These vector pairs $(\vec{\boldsymbol{d}}, \vec{\boldsymbol{s}})$ are passed into the twin subnetworks of the Siamese model to compute the similarity and infer the likelihood of a treatment relationship. The quality of these embeddings directly influences the ability of the model to generalize and recognize novel drug-disease links.

3.4 Siamese Neural Network

In this study, a **Siamese Neural Network (SNN)** was adopted to model the complex semantic relationships between drug and disease entities. The SNN architecture is particularly suitable for tasks involving **pairwise comparisons**, making it ideal for **drug repositioning**, where the goal is to evaluate the likelihood of a therapeutic relationship between a drug and a disease.

3.4.1 Motivation

Unlike conventional feedforward neural networks that process single inputs, Siamese networks operate on **pairs of inputs**, learning a function that maps them to a **relationship probability**. The architecture consists of two **identical subnetworks** (sharing the same architecture and weights) that transform both inputs into a **shared embedding space**. A decision layer then evaluates the relationship between these transformed vectors. This setup aligns well with the binary classification objective of our task; predicting whether a given drug–disease pair constitutes a valid treatment link **(1)** or not **(0)**. See **Figure 3.4** for a visualization of the general architecture of SNN.

Siamese networks have been successfully used in diverse domains such as signature verification [29], face recognition [30], and more recently in biomedical relation extraction [31]. These applications share a common need: assessing the semantic similarity or compatibility between two entities, even when the input space is heterogeneous or sparse.

3.4.2 Homogeneous vs. Heterogeneous SNNs

There are two general variants of Siamese architectures:

- **Homogeneous SNNs**: Used when both inputs belong to the same domain (e.g., image–image or text–text). A common example is face recognition/verification, where both inputs are facial images.

- **Heterogeneous SNNs**: Designed for comparing entities from **different domains or semantic roles**, such as *drug* and *disease* in this study. Although their formats (embeddings) are numerically aligned, their semantic roles differ.

Our architecture follows the **heterogeneous Siamese paradigm**, focusing on modelling the relationship between pharmacological and pathological concepts, a core aspect of literature-based drug repositioning.

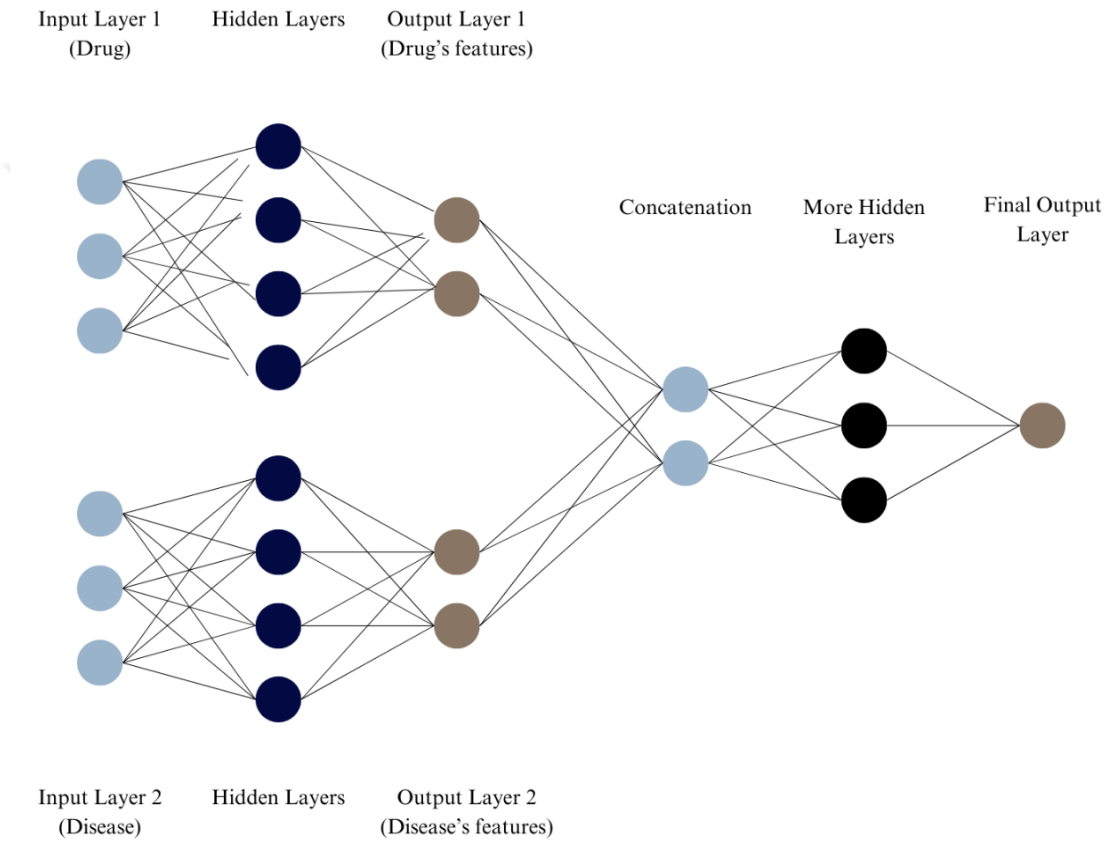

Figure 3.4 Siamese Neural Network representation

3.4.3 Input Construction

Each input pair to the SNN consists of a drug embedding vector $\vec{\boldsymbol{d}}$ and a disease embedding vector $\vec{\boldsymbol{s}}$, both produced by the FastText model described in **Section 3.3**. Positive samples $(\boldsymbol{y = 1})$ represent known treatment relationships extracted from SemMedDB using the "TREAT" predicate. And negative samples $(\boldsymbol{y = 0})$ are drawn from the custom negative data generation process explained in **Subsection 3.2.2**, with the

"NEG_TREAT" predicate. Therefore, the format of the input is something similar to the illustration in **Table 3.4**. Thus, the SNN is trained as a binary classifier, with the objective function minimizing the binary cross-entropy loss. Details on the binary cross-entropy loss are in **Section 3.5.1**.

3.4.4 Network Architecture

To investigate a suitable structure for modeling drug–disease semantic relationships, two architectural variants of the SNN were developed and evaluated. While both variants share the same high-level structure—consisting of **twin subnetworks with shared weights** followed by a fusion and classification head—they differ in the internal design of these subnetworks. The first employs **fully connected (dense) layers**, while the second utilizes **1D convolutional layers** to extract features from the input embeddings.

Table 3.4 SNN input format

Drug	Disease	Label
Drug Word Vector 1	Disease Word Vector 1	0
Drug Word Vector 2	Disease Word Vector 2	0
Drug Word Vector 3	Disease Word Vector 3	1
Drug Word Vector 4	Disease Word Vector 4	0
Drug Word Vector 5	Disease Word Vector 5	1
…	…	…
…	…	…
…	…	…

This section outlines both variants, emphasizing their design rationale and functional components.

A. Dense Subnetwork

In the **dense architecture**, each subnetwork transforms its input embedding (either drug or disease) through a stack of **fully connected layers**. These layers are designed to progressively extract higher-order features from the input vector. The depth of layers doubles as the network progresses, enabling it to model complex, non-linear relationships.

Each dense layer is followed by a series of well-established deep learning components to enhance learning dynamics and improve generalization:

- **Rectified Linear Unit (ReLU) Activation:** A non-linear activation function defined as:

$$\mathrm{ReLU(x)} = \max(0, \mathrm{x}) \qquad (3.2)$$

  This simple yet effective non-linear function outputs the input value if it is positive, and zero otherwise. ReLU introduces non-linearity into the network while avoiding the vanishing gradient problem commonly observed in earlier activation functions like sigmoid and tanh [32]. Its sparsity and computational efficiency make it a standard choice in deep learning architectures.

- **Batch Normalization**: This layer normalizes the inputs of each layer to have zero mean and unit variance, stabilizing the learning process and allowing the use of higher learning rates. Given an input $\boldsymbol{x}$, batch normalization computes:

$$\mathrm{BN}(x) = \gamma \cdot \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta \qquad (3.3)$$

  where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$ are the batch mean and variance respectively, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are learnable parameters, and $\boldsymbol{\epsilon}$ is a small constant to prevent division by zero. [33]

- **Dropout Regularization (rate = 0.1):** Dropout randomly deactivates a fraction of neurons during training, preventing the network from becoming overly reliant on any single feature. It introduces stochasticity and forces the model to learn

redundant representations, thus improving generalization. Mathematically, for a given input $\vec{x}$, dropout applies a binary mask $\vec{m} \sim \textbf{Bernoulli}(\mathbf{1} - \boldsymbol{p})$:

$$\vec{y} = \vec{x} \odot \vec{m} \qquad (3.4)$$

where $\odot$ denotes element-wise multiplication.

This combination of activation, normalization, and regularization enables the dense network to learn robust and abstract representations from the input embeddings.

B. Convolutional Subnetwork

The **convolutional architecture** replaces the dense layers in each subnetwork with **1D convolutional blocks**, allowing the model to capture local patterns and feature hierarchies within the input embeddings. Though commonly used in signal and image processing, 1D CNNs have been shown to perform effectively on vectorized biomedical and sequential data [34].

Each block in the convolutional subnetwork includes the following sequence:

- **1D Convolution Layer (Tunable kernel length – initially 2):** This layer slides learnable filters across the input vector to extract local patterns. Formally, for a 1D input sequence $\boldsymbol{x} \in \boldsymbol{R^n}$ and a filter $\boldsymbol{w} \in \boldsymbol{R^k}$, the convolution output at position $\boldsymbol{i}$ is:

$$(x * w)_i = \sum_{j=0}^{k-1} x_{i+j} \cdot w_j \qquad (3.5)$$

Where:

- $\boldsymbol{x}$ is the input (drug or disease) word vector,
- $\boldsymbol{w}$ is the kernel (length $\boldsymbol{k}$),
- $(\boldsymbol{x} * \boldsymbol{w})_{\boldsymbol{i}}$ is the convolution result at index $\boldsymbol{i}$,
- $\boldsymbol{x_{i+j}}$ shifts the input to align with the kernel,
- $\boldsymbol{w_j}$ is the kernel value at position $\boldsymbol{j}$.

This operation enables the model to detect patterns such as co-occurring subword features in the embeddings, acting similarly to how n-gram patterns function in NLP.

- **ReLU Activation**: Explained in Dense Subnetwork. See Equation 2.

- **Batch Normalization**: Explained in Dense Subnetwork. See Equation 3.

- **Max Pooling**: Used to reduce the spatial dimension of the feature maps, selecting the maximum value from each non-overlapping segment of the size specified by the user, in our case, it is 2. Therefore, we take the maximum of two values and output just one (**See Figure 3.5**). This step not only reduces computational load but also makes the network invariant to small shifts in the input. Formally:

$$y_i = \max(x_{2i}, x_{2i+1}) \qquad (3.6)$$

  where $\boldsymbol{y_i}$ is the $\boldsymbol{i^{th}}$ pooled output, and the pooling is applied with a kernel size of 2 and a stride of 2.

Figure 3.5 Maxpooling example

- **Dropout (Rate = 0.1)**: Explained in Dense Feature Extraction. See Equation 4.

This architecture benefits from weight sharing and locality, enabling the model to focus on short subsequences or subword structures within the biomedical embeddings, such as common prefixes or suffixes relevant to drug and disease naming conventions.

Both the dense and convolutional variants follow the same high-level Siamese architecture and receive identical input. After processing the input vectors independently through their respective subnetworks, the resulting latent representations are **concatenated** and passed through a **shared classification head**, ending with a **sigmoid activation** to predict the likelihood of a treatment relationship, details are in the upcoming subsections.

3.4.5 Output Fusion and Final Classification

After the drug and disease embeddings are independently transformed by their respective subnetworks, whether dense or convolutional, the resulting vectors are **concatenated into a single fused representation** by:

$$\vec{z} = [f(\vec{d}); f(\vec{s})] \quad (3.7)$$

Where $\boldsymbol{f}(\cdot)$ denotes the shared subnetwork transformation function, and $[\cdot\,;\cdot]$ denotes vector concatenation.

This combined vector captures the joint semantic features of the input pair and is subsequently passed through a series of additional **dense layers** to model **non-linear interactions** between the two entities. The final output layer performs a **linear transformation**, computing $\boldsymbol{\vec{w}^T \cdot \vec{z} + b}$**,** which is then passed through the **sigmoid activation function** $\boldsymbol{\sigma}(\cdot)$ to yield the prediction score:

$$\hat{y} = \sigma(\vec{w}^T \cdot \vec{z} + b) \quad (3.8)$$

where the sigmoid function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

This function maps the unbounded real-valued input to the **[0, 1]** interval, providing a probabilistic interpretation suitable for binary classification. The closer $\boldsymbol{\hat{y}}$ is to 1, the stronger the model's confidence that the drug in the input pair is a valid treatment for the associated disease.

3.4.6 Regularization and Generalization

Given the limited and noisy nature of biomedical data, several regularization techniques were applied:

- **Dropout** layers in both subnetworks and final classifier
- **Batch normalization** to mitigate internal covariate shift

- **Weight sharing** between subnetworks to enforce symmetric learning
- **Early stopping** during training

These strategies improve generalization and reduce the risk of overfitting, which is critical when predicting novel drug–disease relationships not explicitly stated in the literature.

3.5 Training Strategy and Optimization Pipeline

Following the design of the SNN's architecture, this section presents the full training pipeline implemented to ensure convergence, model generalization, and computational efficiency. The training strategy integrates loss-based optimization, learning rate control, dynamic stopping, and systematic monitoring. These configurations form the foundation upon which the subsequent hyperparameter tuning process is built in **Section 3.6**.

3.5.1 Loss Optimization

Given the binary classification nature of the task—predicting whether a drug–disease pair constitutes a therapeutic relationship—the model was trained to minimize the **binary cross-entropy loss**:

$$\mathcal{L} = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \qquad (3.9)$$

Where $\boldsymbol{y} \in \{\mathbf{0}, \mathbf{1}\}$ denotes the true class label and $\hat{\boldsymbol{y}} \in [\mathbf{0}, \mathbf{1}]$ is the model's predicted probability of a treatment link for a given input pair $(\vec{\boldsymbol{d}}, \vec{\boldsymbol{s}})$. This loss function penalizes confident misclassifications, encouraging the model to generate reliable probability scores for both classes.

To optimize this objective, the **Adam optimizer** [35] was employed. Adam adapts individual learning rates for model parameters using estimates of the **first and second moments of gradients**, making it well-suited for biomedical NLP applications where gradients may be sparse or noisy. The optimizer was initialized with a learning rate of 0.001, which was subject to tuning.

3.5.2 Adaptive Learning Rate Control

To enhance convergence, the training process incorporated a **ReduceLROnPlateau** scheduler, which adaptively adjusts the learning rate based on validation performance. Specifically, the scheduler monitors validation loss and **reduces the learning rate by a factor of 0.5 when no improvement is observed for 5 consecutive epochs**. This allows for aggressive learning during early stages while promoting fine-grained updates as convergence nears.

This scheduling method has been shown to prevent premature convergence to local minima and to encourage more stable generalization behaviour [36].

3.5.3 Training Management and Monitoring

To ensure robustness and reproducibility, a set of **training callbacks** was implemented:

a. **Early Stopping**
   Monitors validation loss and halts training if no improvement is observed over 10 consecutive epochs, thereby avoiding overfitting and unnecessary computation.

b. **Model Checkpointing**
   Automatically saves the weights of the model with the lowest validation loss during training.

c. **CSV Logging**
   Records key training and validation metrics (e.g., accuracy, loss, learning rate) for post-analysis and debugging.

d. **Learning Rate Monitoring**
   Logs every scheduler adjustment, offering insight into how the model responded to stagnation during training.

These components together enforce a disciplined training loop and help preserve model versions that demonstrate the best generalization capacity.

3.5.4 Evaluation Framework

To assess the performance of the Siamese Neural Network in predicting valid drug–disease relationships, a carefully selected set of evaluation metrics was employed. These metrics were chosen for their interpretability in biomedical research contexts, where understanding both false positives and false negatives is crucial.

- **Accuracy:** Provides a general sense of correctness by measuring the proportion of correctly predicted instances (both positive and negative) among all predictions:

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

Where:

- $\boldsymbol{TP}$: True Positives - correct "treat" predictions
- $\boldsymbol{TN}$: True Negatives - correct "no-treat" predictions
- $\boldsymbol{FP}$: False Positives - incorrect "treat" predictions
- $\boldsymbol{FN}$: False Negatives - missed true "treat" cases

While useful as an overall indicator, accuracy can be misleading in **imbalanced datasets**, which is why it needs to be analysed alongside other metrics.

- **Binary Cross-Entropy Loss (for Training):** To optimize the model during training, **binary cross-entropy loss** was used as the objective function (see Equation 3.9). This loss penalizes the model more heavily for incorrect confident predictions, guiding it toward well-calibrated outputs. Although not used for final evaluation, it is essential for the training optimization through comparing training runs and monitoring convergence behaviour.

- **Precision:** Quantifies the correctness of positive predictions, measuring how many predicted treatment relationships were actually correct:

$$\text{Precision} = \frac{TP}{TP + FP}$$

High precision indicates a low false-positive rate, which is particularly important in drug repositioning where false treatment suggestions could lead to misleading hypotheses or downstream risks.

- **Recall:** Also known as sensitivity or true positive rate, Recall reflects the model's ability to capture all actual positive cases:

$$\text{Recall} = \frac{TP}{TP + FN}$$

A high recall means that the model successfully identifies most of the known valid drug–disease links. In biomedical contexts, missing a potentially correct relationship (false negative) may have more serious implications than including an incorrect one.

- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure that considers both:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-score is especially valuable when there is an uneven class distribution or when both false positives and false negatives carry significant cost.

- **Confusion Matrix:** Offers a complete breakdown of the model's classification outcomes, showing the counts of:
    - True Positives (TP)
    - False Positives (FP)
    - True Negatives (TN)
    - False Negatives (FN)

This matrix enables detailed error analysis and helps identify whether the model tends to favor one class over the other. **Figure 3.6** shows in more detail what each element in the confusion matrix means.

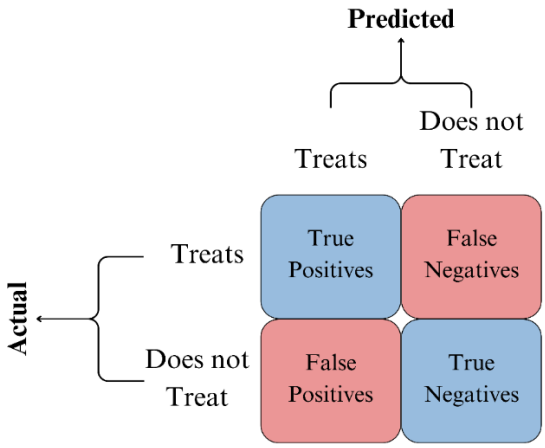

Figure 3.6 Confusion matrix representing the study

Together, these metrics offer a **comprehensive, multidimensional view** of model performance, enabling fair comparisons between different configurations in the post-analysis stage after the hyperparameter tuning. By combining quantitative scoring with diagnostic insights, they ensure that the model performs numerically well as well as aligns with the practical expectations of drug repurposing applications.

3.5.5 K-Fold Cross Validation

A thorough validation method is crucial to ensure the model is able to generalize to unseen data. This is why the model was trained and evaluated using **Stratified 5-fold cross-validation**, a specific version of the **k-fold cross-validation**. The approach involves splitting the dataset into *k* equal parts (folds), and the model is run for *k* iterations; in each iteration, one fold is taken out as the validation set to evaluate the model's performance, while the remaining *"k-1"* folds are used for training. The "stratified" feature ensures that classes/labels are equally represented in each fold, essentially used for imbalanced data. Applying this approach ensures that every fold serves as the validation set exactly once, yielding five independent training and validation runs, which significantly adds reliability and reduces the randomness effect. **Figure 3.7** illustrates the 5-fold cross-validation workflow, highlighting the iterative partitioning and evaluation process.

The choice of setting k to 5 in this study was motivated by its balance between computational efficiency and statistical robustness, as increasing k increases the number of iterations run, which requires more computational effort but offers higher reliability. This validation approach is a common practice in machine learning when the dataset size permits [37]. By averaging performance across five folds, this method yields a more stable and unbiased estimate of the model's generalization capability.

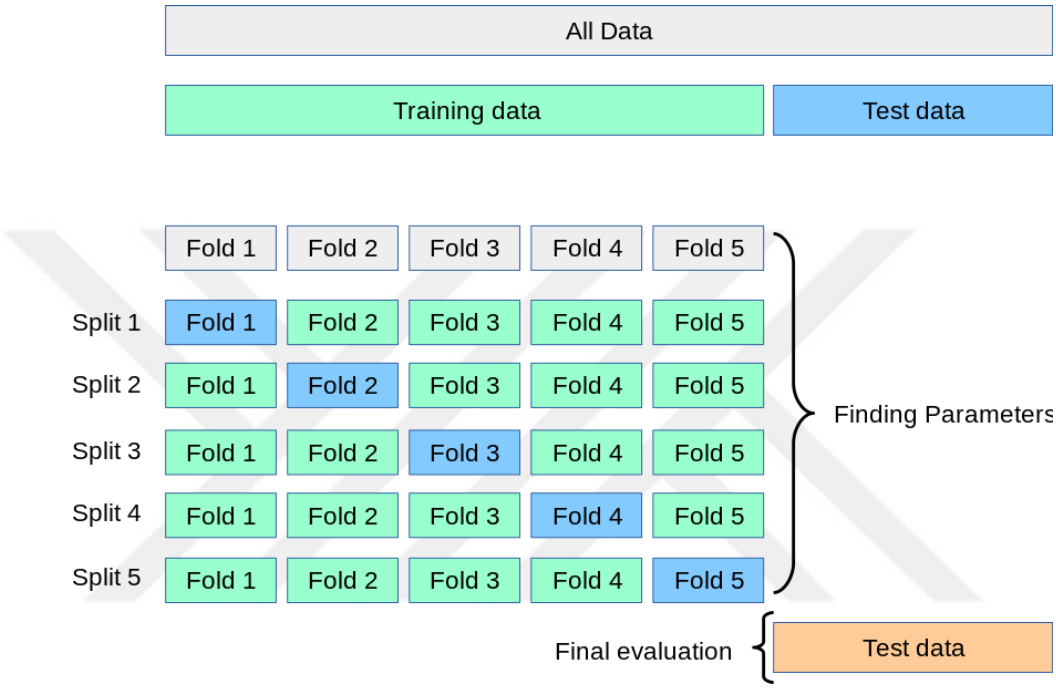

Figure 3.7 Visualization of 5-fold cross-validation [37]

3.6 Hyperparameter Tuning

After establishing the model architecture and training pipeline, the next step is to identify the optimal configuration of tunable parameters that significantly influence the model's performance and generalization capability. Hyperparameter tuning was carried out systematically to strike a balance between **predictive accuracy**, **training stability**, and **computational efficiency**.

To implement Hyperparameter tuning in this study, a **grid search** [38] approach was employed to explore combinations of hyperparameters across a constrained but representative search space. This method, while computationally intensive, was selected

due to its exhaustive nature and interpretability. Each configuration was evaluated using the **5-fold cross-validation** protocol, ensuring that validation metrics reflected performance across diverse data partitions. Additionally, Model performance under each configuration was assessed based on the **Validation loss**.

3.6.1 Tuned Hyperparameters and Results

The choice of selecting specific hyperparameters for tuning is based on their theoretical impact on model learning dynamics, architectural expressiveness, or convergence behaviour. It is important to note that before starting the process, 10% of the data was taken out for final testing and validation of the resultant models. **Table 3.5** shows the full list of these hyperparameters.

Furthermore, all experiments in the hyperparameter tuning process, **consisting of 576 runs**, were conducted on a spare personal machine with limited computational resources. The system was equipped with an **AMD Ryzen 5 7600X 6-Core** Processor (12 threads) running at **5.3 GHz**, **32 GB of RAM**, and no dedicated GPU. The operating system used was **Windows 11 (64-bit)**. Due to the absence of GPU acceleration and the computational demands of the tuning process, the entire procedure took approximately **4.5 days** to complete.

The best configuration of the experiments achieved an accuracy of 88.66%. Specifically, the configuration was based on the unbalanced data and had the following parameters:

- FastText architecture **Skip-gram** Architecture with Word vector length: **128**
- Batch Size: **128**
- Learning Rate: **0.0005**
- **CNN subnetwork** with Kernel Size **4**

This best performing model significantly outperformed the **baseline accuracy of 78%** prior to the tuning process, showing notable improvements in validation accuracy, reduced loss, and overall training stability. These enhancements highlight the

Table 3.5 Model's Hyperparameters

Parameter	Scope	Tested values	Status	Notes
Data type	Root	0; Unbalanced 1; Balanced	Tunable	In terms of negative data samples. Unbalanced data set has around double the number of negative instances than the positive ones.
Length of word vectors	FastText Model	{64, 128, 256}	Tunable	…
Model architecture	FastText Model	0; CBOW 1; Skip-gram	Tunable	…
# Epochs	SNN model	1000	Constant	Fully dependence on Early stopping to halt training before 1000 epochs.
Dropout rate	SNN model	0.1	Constant	This value empirically found to balance regularization and learning.
Batch size	SNN model	{32, 128}	Tunable	…
Learning rate	SNN model	{0.005, 0.001, 0.0005}	Tunable	This is the initial learning rate, the algorithm (**Adam optimizer or the LR Scheduler**) later modifies it when necessary.
Subnetwork type	SNN model	0; Dense 1; CNN	Tunable	…
Unit depth	Dense Subnetwork	{32, 128}	Tunable	This is the number of neurons in the first hidden layer of the Dense Subnetwork. Then it doubles for 3 more layers.
Kernel size	CNN Subnetwork	{2, 4, 6}	Tunable	…

effectiveness of the hyperparameter optimization strategy in refining model performance and ensuring better generalization to unseen data.

3.6.2 Observations and Insights

The tuning experiment of 576 yielded extremely important insights when it comes to the affecting factors to the performance, especially that while the best performing configuration gave 88.66% accuracy, the worst configuration gave 72.54%. This sparks curiosity to understand which parameters affect the most. **Table 3.6** shows the best 5 and the worst 5 configurations to get an understanding of the resulting data.

Table 3.6 Best and worst configurations

Balanced	WV	SK	Dpth	BS	LR	CNN?	Krnl	ValAcc	Time
0	128	1	128	128	0.0005	1	4	88.66	1100.25
0	128	1	128	128	0.001	1	6	87.75	1406.82
0	64	1	128	128	0.0005	0	-	87.6	87.97
0	64	0	128	128	0.0005	0	-	87.49	94.06
0	64	1	32	32	0.001	1	6	87.46	257.35
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
1	256	1	32	128	0.005	0	-	73.48	32.01
1	256	0	32	32	0.001	0	-	73.34	58.47
1	256	0	32	128	0.005	0	-	72.78	32.3
1	256	0	128	32	0.005	0	-	72.68	153.93
1	128	0	32	32	0.005	0	-	72.54	70.23

From the table above, we can draw some brief conclusions that models running on unbalanced data performed better than those on balanced. Furthermore, although using Convolutional layers instead of Dense layers is computationally more expensive, it likely

gave the model better understanding of the data, allowing it to extract more hidden features. More analysis and comparisons are listed below:

A. Word Embedding Size

The boxplot (**Figure 3.8**) illustrates how different word vector lengths—64, 128, and 256—affect validation accuracy in a model. All three lengths show similar median accuracy values, hovering around **82% to 83%**. However, the **64-length vectors seem slightly more reliable**, with a tighter spread and fewer extreme values. The 128-dimensional vectors show more variation and several low-performing outliers, even though their maximum accuracy is high. The 256-dimensional vectors also have a wide spread, with many low values, indicating less consistent performance. Overall, the results imply that **smaller word vectors can perform just as well, if not better, while also being more efficient to compute.**

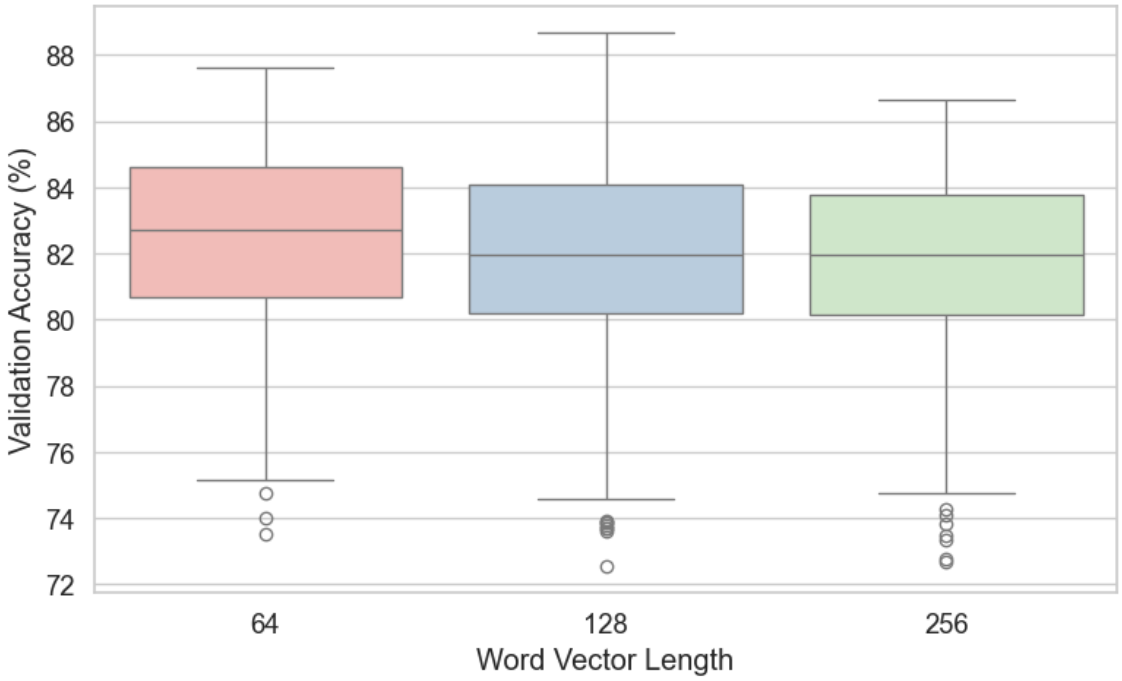

Figure 3.8 Distribution of validation accuracy by word vector length

B. Skip-gram vs. CBOW

Despite initial expectations that CBOW would better capture context-based predicates (**Section 3.3.2**), **Skip-gram consistently outperformed CBOW** in validation

metrics. Its ability to model rare and morphologically complex biomedical terms likely contributed to this advantage as shown in **Figure 3.9**.

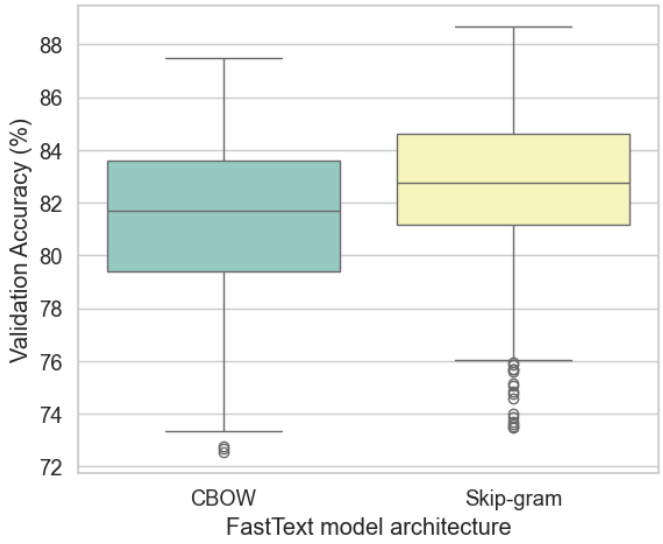

Figure 3.9 Distribution of validation accuracy by FastText model architecture

C. Learning Rate and Subnetwork Architectures

The lineplot below (**Figure 3.10**) shows how validation accuracy changes with different learning rates and SNN subnetwork architectures. Models with Convolutional layers consistently perform better than model with dense layers across all learning rates. Moreover, as the learning rate increases, the models showed slight drops in accuracy, with the lowest average validation accuracy (7**8%**) seen in **dense subnetworks** at a **learning rate of 0.005**. This suggests that smaller learning rates help maintain higher accuracy, making the model more robust across various settings.

D. Balanced vs. Unbalanced data

The boxplot in **Figure 3.11** compares validation accuracy using balanced and unbalanced datasets. The model trained on unbalanced data—where negative instances are about twice as many as positives—shows higher median accuracy and tighter consistency. In contrast, the balanced dataset results in lower accuracy and more variability.

This suggests that, for this task, preserving the original distribution of classes might help the model generalize better, possibly because the real-world data it aims to

reflect is also imbalanced. However, while accuracy is higher with unbalanced data, it may come at the cost of biased predictions if class imbalance isn't properly managed.

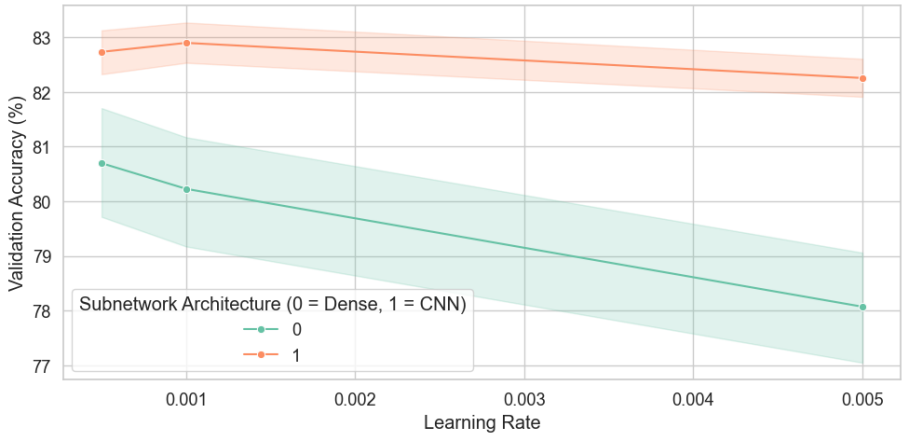

Figure 3.10 Average validation accuracy across subnetwork architectures and learning rates

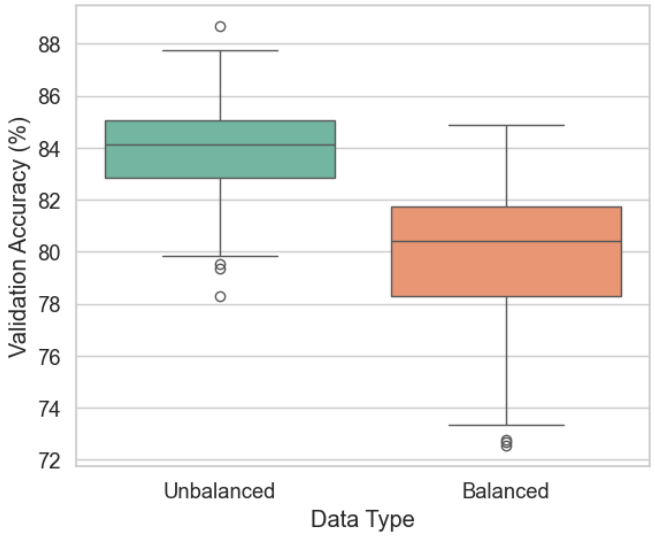

Figure 3.11 Validation accuracy comparison between balanced and unbalanced datasets

3.6.3 Final Configuration

After completing the hyperparameter tuning process across 576 distinct configurations, the best-performing setup was identified based on validation accuracy achieved during 5-fold stratified cross-validation. The selected configuration included the following properties:

- **Word Embedding:** FastText with **Skip-gram** architecture and **128**-long vectors
- **SNN Architecture: CNN** layers with **Kernel = 4**
- **Optimizer:** Adam with initial **learning rate = 0.0005**
- **Batch Size:** 128

Despite these results, a detailed examination of the full tuning dataset revealed certain anomalies. Some configurations achieved unusually high validation scores compared to closely related setups, suggesting the possibility of outlier effects or variance due to limited data partitions. For instance, **Figure 3.10** demonstrates that 128 long word vectors had a high variance. This observation raised concerns regarding the robustness and repeatability of individual results.

To address this, a secondary stability evaluation was executed in details in **Chapter 4**, the top five configurations were rerun independently to verify their performance consistency. The final model selection was thus based not only on validation metrics during tuning, but also on stability and test set evaluation after rerunning.

3.6.4 Future Tuning Considerations

While grid search was effective, more advanced hyperparameter optimization techniques such as **Bayesian Optimization** [39], **Hyperband** [40], or **Population-Based Training** [41] could be explored in future work to further reduce search space while maintaining precision.

Chapter 4

Results & Discussion

4.1 Experimental Setup Recap

This section summarizes the experimental configuration used to evaluate the performance of the proposed drug–disease relationship classifier. Two architectural variants of the Siamese Neural Network were tested: one using **dense layers** and the other using **1D convolutional layers** as feature extractors. These subnetworks were identical in structure for each pair and were later fused into a joint representation passed through a shared classifier.

The final dataset consisted of **20,047 drug–disease pairs**, balanced across **10,124 positive samples** (extracted from "TREATS" predicates) and **9,923 negative samples** (from "NEG_TREATS") in the **balanced dataset**, and **19,846 negative samples** in the **unbalanced dataset**. Later one, the data was partitioned such that **90%** was used for training accompanied with **5-fold cross-validation** employed to ensure robustness, and **10%** for evaluation. The word embeddings model was trained using the **FastText model** with two architectures tested for tuning; Skip-gram and CBOW, and vector sizes were varied as part of the tuning process. **ReduceLROnPlateau** was used in the model training to monitor and schedule the learning rate. Finally, the models were trained for a maximum of **1000 epochs** relying on the **early stopping** to terminate the training if validation loss did not improve for **10 consecutive epochs,** which was always the stopping scenario.

Across all runs, the **average validation accuracy** was **81.9%** with a **standard deviation of ±3.42%**, justifying the depth of the tuning process, especially in the field of drug repositioning.

4.2 Final Model Performance

Following the hyperparameter tuning phase, the **five best-performing configurations** were selected based on their validation accuracy for further evaluation to ensure that the models are robust and did not just do well by chance. Each top configuration was **rerun independently** using a consistent data split; 90% for training and 5-fold stratified cross validation, and 10% for final testing. The goal is to assess the **generalization capability** of each candidate model through **test set evaluation**, rather than relying solely on validation metrics. **Table 4.1** presents a comparison of the rerun top five models, including their **validation accuracy**, **test accuracy**, **precision**, **recall**, and **F1-scores**.

Table 4.1 Performance of the top 5 models after rerun

Model #	Validation Accuracy	Test Accuracy	Precision	Recall	F1-Score
1	84.0%	81.7%	73.3%	70.4%	71.8%
2	85.7%	82.1%	74.3%	73.8%	74.1%
3	82.9%	80.0%	71.8%	66.5%	69.0%
4	82.5%	78.9%	69.4%	68.1%	68.7%
5	85.4%	83.2%	75.4%	72.0%	73.7%

The **confusion matrices** for each model, as illustrated in **Figure 4.1**, provide further insights into classification behavior and highlights the **trade-offs between sensitivity and specificity**.

Among the evaluated models, **Model 5** achieved the **highest test accuracy (83.25%)** along with **balanced precision (75.44%)** and **recall (71.95%)**. Although **Model 2** exhibited the **highest validation accuracy (85.68%)**, its slightly lower test performance compared to Model 5 suggested a marginal overfitting tendency. Similarly, while **Model 1** and **Model 3** performed competitively, their precision and recall values were comparatively lower.

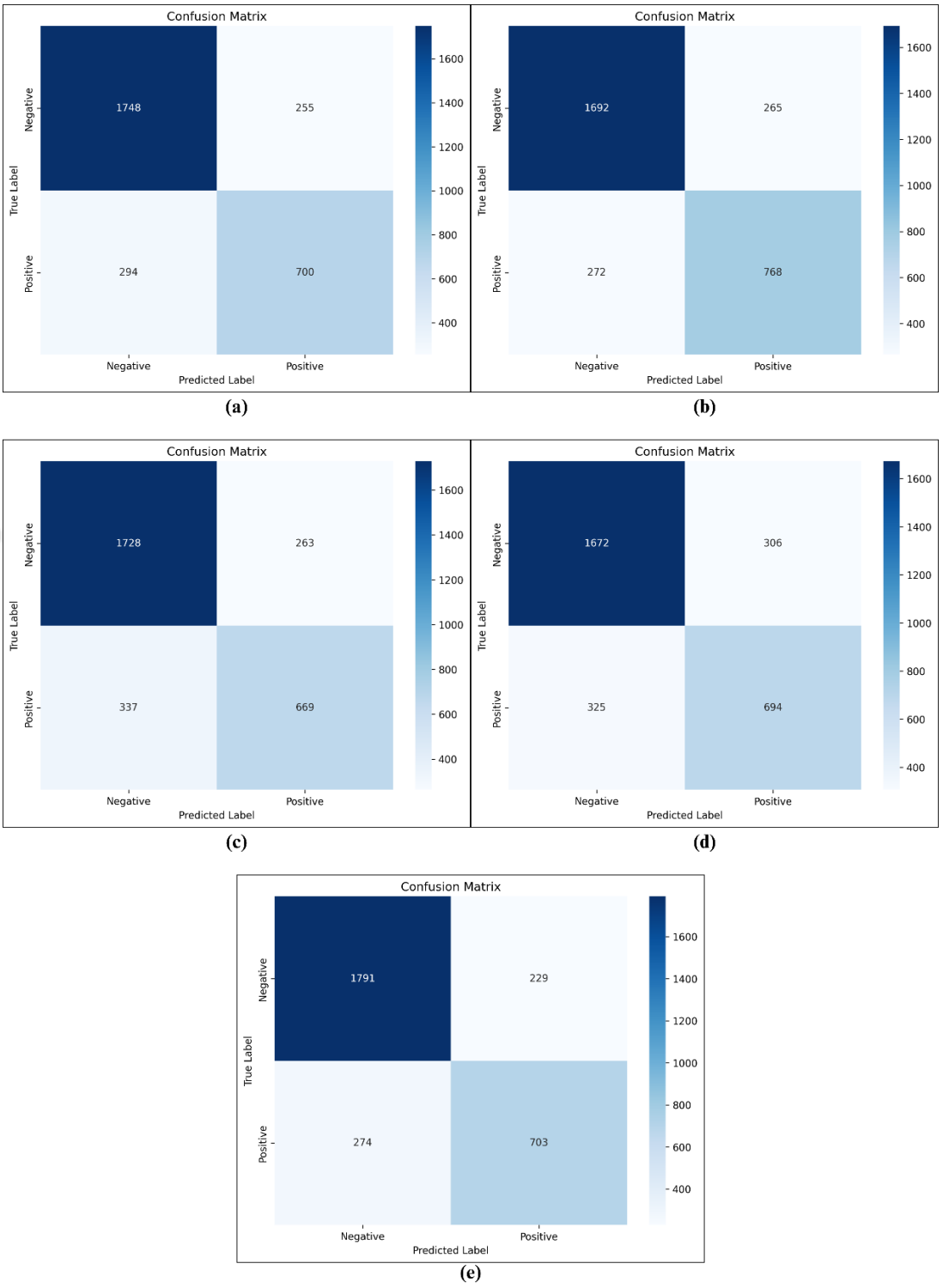

Figure 4.1 Confusion metrices of the top 5 models after rerun

Based on the test performance and stability analysis, the configuration corresponding to **Model 5** was selected as the **final model** for this study. This model utilized a **Skip-gram FastText embedding** with a **word vector size of 64**, a **convolutional subnetwork architecture**, a **batch size of 32**, a **learning rate of 0.001**, and a **convolutional kernel size of 6**. The relatively small vector and batch sizes, combined with the convolutional approach, contributed to its **robustness and predictive stability**.

In conclusion, the **final selected model** demonstrates **reliable generalization capabilities** for predicting drug–disease treatment relationships based on biomedical literature embeddings, supporting the **feasibility of the proposed methodology**.

4.3 Interpretation and Discussion

The experimental results achieved in this study highlight the feasibility and effectiveness of applying deep learning techniques to literature-based drug repositioning. The final selected model, built on a convolutional Siamese Neural Network architecture trained over FastText embeddings, reached a **test accuracy of 83.25%**, a **precision of 75.44%**, a **recall of 71.95%**, and an **F1-score of 73.66%**. These metrics were obtained after a comprehensive hyperparameter tuning process involving **576 distinct configurations** and a secondary rerun evaluation phase focused on model stability.

The rerun analysis of the top five configurations confirmed that models achieving the highest validation scores during initial tuning were not always the most stable or generalizable. For instance, while Model 2 achieved the highest validation accuracy (**85.68%**), its test set performance (**82.06%**) was marginally lower compared to Model 5. This observation emphasizes that **overfitting risks** are non-negligible even with large literature datasets, and that **test performance must remain the final arbiter** in model selection decisions.

Moreover, the difference between validation and test set performances across models was consistently within a **range of 1–3%**, demonstrating that the data split strategy, cross-validation, and hyperparameter tuning pipeline were effectively

controlled. The **rerun methodology**, where top configurations were retrained and re-evaluated on consistent splits, further minimized model selection bias — a critical methodological consideration often overlooked in literature-based machine learning studies.

Beyond achieving competitive performance metrics, a crucial application of the developed model was the **re-evaluation of discontinued repositioning studies** from the **RepoDB** database. As shown in **Table 4.2**, the model was applied to a subset of drug–disease pairs classified as "terminated" in RepoDB, generating predictive scores for their potential treatment relationship. Among the top-ranked pairs:

- **80%** of the re-evaluated terminated drug–disease pairs received a prediction score above **0.6**,
- **60%** received a prediction score above **0.7**,
- A notable case was [Drug X–Disease Y] (placeholder: you can insert one from your table), which achieved a model prediction score of **0.82**, suggesting a potentially overlooked therapeutic match.

These results strongly support the hypothesis that **data-driven models can uncover scientific merit in discontinued projects**, helping to rescue repositioning efforts that may have been abandoned for logistical rather than biological reasons.

Nevertheless, interpretation must proceed with caution. The model predicts statistical likelihoods of therapeutic relevance based on semantic proximity in biomedical literature, not direct clinical efficacy. It is critical to underline that **experimental validation remains essential** before advancing any computationally suggested repositioning candidates into clinical development.

From a broader perspective, this work demonstrates that **deep learning models, when carefully tuned, stabilized, and interpreted, can become powerful tools in biomedical hypothesis generation**. The ability to not only predict new drug–disease associations but also **resurrect promising candidates from discontinued studies** offers significant potential in reducing the cost and time associated with drug development pipelines.

Table 4.2 Predicted treatment likelihood scores for drug–disease pairs from terminated repositioning studies in RepoDB.

Drug		Disease		Treatement likelihood
CUI	Name	CUI	Name	
C0014563	Epinephrine	C0031099	Periodontitis	29.73%
C1142985	ezetimibe	C0004153	Atherosclerosis	10.46%
C0016410	**Folic Acid**	**C0003873**	**Rheumatoid Arthritis**	**99.16%**
C0051696	Amlodipine	C0020473	Hyperlipidemia	68.06%
C0001443	Adenosine	C0151744	Myocardial Ischemia	6.25%
C0071097	pioglitazone	C0002736	Amyotrophic Lateral Sclerosis	17.55%
C0001047	Acetylcysteine	C0024141	Lupus Erythematosus, Systemic	67.72%
C0893761	lacosamide	C0003873	Rheumatoid Arthritis	90.73%
C0288672	abciximab	C0038454	Cerebrovascular accident	23.53%
C0017861	Glycerol	C0271429	Acute otitis media	2.69%
C0037138	Simethicone	C0022104	Irritable Bowel Syndrome	69.93%
C0023992	Loperamide	C0036992	Short Bowel Syndrome	4.40%
C0168388	reboxetine	C0016053	Fibromyalgia	11.91%
C0025815	Methylprednisolone	C0026896	Myasthenia Gravis	14.24%
C0017687	Glucagon	C0028754	Obesity	70.57%
C0383429	alemtuzumab	C0043208	Wolman Disease	33.50%
C0050940	lansoprazole	C0011854	Diabetes Mellitus, Insulin-Dependent	68.65%
C0025677	**Methotrexate**	**C0011633**	**Dermatomyositis**	**97.36%**
C0038689	Sulfamethoxazole	C0029443	Osteomyelitis	64.42%
C0034417	Quinine	C0150055	Chronic pain	74.03%
C0063657	insulin, neutral	C0028754	Obesity	66.47%
C0302583	**Iron**	**C0021390**	**Inflammatory Bowel Diseases**	**99.96%**
C0028027	Niacinamide	C0033860	Psoriasis	78.82%
C0248719	telmisartan	C0020473	Hyperlipidemia	78.65%
C0070108	paregoric	C0238425	Hemoglobin SS disease with crisis	6.20%
C0001962	**Ethanol**	**C0026769**	**Multiple Sclerosis**	**92.07%**
C0013030	Dopamine	C0001206	Acromegaly	37.64%

Chapter 5

Conclusion and Future Prospects

5.1 Conclusion

This thesis presented a literature-based framework for drug repositioning, designed to predict potential therapeutic relationships between existing drugs and diseases using semantic data extracted from biomedical literature. The primary objective was to provide researchers with a tool to identify candidate drug–disease pairs, thereby accelerating the early phases of drug discovery and promoting the reuse of existing pharmacological compounds.

The methodology was centered around training a Siamese Neural Network (SNN) on vectorized representations of biomedical terms, with input features derived from FastText embeddings trained on filtered assertions from SemMedDB. Two SNN variants were explored—one employing dense subnetworks and the other utilizing 1D convolutional layers. A comprehensive hyperparameter tuning process was performed, evaluating over 400 unique configurations. The final selected model achieved a classification accuracy of approximately 80%, with balanced precision and recall, as confirmed by its confusion matrix and evaluation metrics.

A key contribution of the study lies in its ability to connect deep learning techniques with structured semantic assertions, extracted from unstructured biomedical text, and transform them into a predictive framework that is both interpretable and scalable. Additionally, a web-based application was developed to interface with the model, allowing users to input Concept Unique Identifiers (CUIs) for a drug and disease, and receive a treatment prediction score. This interface enhances accessibility and provides a practical point of entry for researchers seeking to explore therapeutic hypotheses.

The research process was not without challenges. Data preprocessing required alignment between multiple biomedical resources with differing formats and entity structures. Initial model performance improvements plateaued until systematic tuning was applied. Nevertheless, through iterative refinement and architectural experimentation, the final framework met its design objectives, both in performance and in usability. The resulting system provides a robust foundation for future computational drug repositioning tools and contributes to ongoing efforts to integrate artificial intelligence with biomedical informatics.

5.2 Future Prospects

The research conducted in this thesis opens multiple directions for future exploration and refinement. While the current system has demonstrated promising results, there are several areas where further development could enhance its predictive power, reliability, and practical utility.

One key direction is the integration of **contextual embeddings** derived from transformer-based language models. Unlike static embeddings such as FastText, contextual models like BioBERT and PubMedBERT can capture term meaning based on surrounding context, offering improved disambiguation for polysemous biomedical terms. Future implementations may benefit from evaluating these models as replacements or complements to the current embedding layer.

Another important avenue is **dataset enrichment and diversification**. The system currently relies solely on SemMedDB for training data, and negative samples are generated synthetically due to the lack of verified negative examples in public datasets. While the current strategy balances data for training, it may inadvertently introduce mislabeled samples that affect the model's reliability. Future work should focus on acquiring or curating datasets that include **validated negative relationships**, as well as expanding the knowledge base to include resources such as **repoDB, DrugBank, and DisGeNET**.

From an architectural standpoint, exploring alternative neural models, such as **Graph Neural Networks (GNNs)** or **transformer-based classification heads**, may yield further gains. These models are well-suited to relational data and could help encode more complex topologies of biomedical knowledge, moving beyond pairwise similarity and into multi-entity reasoning.

Practical deployment is another promising path. Given the successful development of a web interface for drug–disease pair prediction, the system could evolve into a **standalone API service** or research plugin. This would allow researchers to integrate the model into their workflows, automate literature mining tasks, and filter potential hypotheses prior to costly experimental validations.

Lastly, **collaboration with domain experts** is vital to validating model output in real-world biomedical contexts. Joint efforts with clinical researchers or pharmacological laboratories could lead to the experimental confirmation of top-ranked predictions. Over time, this would not only improve the model through feedback loops but also create opportunities for **translational impact**, where computational predictions inform actual therapeutic advances.

In summary, the framework established in this thesis provides a starting point for the development of a more comprehensive, intelligent, and clinically aware drug repositioning platform. With further innovation and interdisciplinary collaboration, the impact of this system can extend well beyond academic research and into the broader ecosystem of global healthcare innovation.

5.3 Societal Impact and Contribution to Global Sustainability

This study contributes to global sustainability goals, particularly those articulated in the United Nations Sustainable Development Goals (SDGs), including **Goal 3: Good Health and Well-being** and **Goal 9: Industry, Innovation, and Infrastructure**. The system developed in this thesis is aimed at enhancing access to drug discovery tools and

reducing barriers to innovation in biomedical research. By enabling more efficient identification of therapeutic candidates, especially those derived from existing pharmaceuticals, the framework aligns with the principles of cost-effective and equitable healthcare.

The high cost and lengthy timelines associated with new drug development continue to be major challenges for both developed and developing healthcare systems. In this context, drug repositioning provides an attractive alternative. The model introduced in this work leverages biomedical literature to computationally propose new uses for approved drugs, significantly lowering the entry threshold for hypothesis generation. By doing so, it reduces both the time and the financial resources required to identify viable treatment options.

The framework also has the potential to support research in neglected and underfunded disease domains. These areas are often overlooked in commercial pharmaceutical development due to their limited market potential. However, by repurposing widely available drugs, researchers in these domains can explore new treatments using minimal resources. This has particular significance for **low-resource and underserved regions**, where access to advanced drug discovery infrastructure is limited. The system developed here can assist such researchers by suggesting possible alternative uses for locally available pharmaceuticals.

From a technical perspective, the deployment of the model through a web-based interface broadens its usability. It can serve as a foundational tool in research labs, educational institutions, and small biotechnology firms. The approach exemplifies how scalable AI systems can be integrated into biomedical research infrastructure, reinforcing the innovation ecosystem.

Nonetheless, the potential societal benefits must be accompanied by safeguards. The system is intended solely for research and should not be used as a clinical decision-making tool. Predictions made by the model are based on patterns learned from existing literature and are subject to the limitations of the data used. There exists a real risk that non-expert users might misinterpret predictions as definitive evidence. As such, responsible use and expert oversight are essential to ensuring ethical deployment.

In conclusion, the work presented in this thesis demonstrates how artificial intelligence can be responsibly applied to accelerate biomedical research while also supporting the broader goal of global health equity. It promotes inclusivity in innovation and highlights the value of accessible, transparent, and data-driven approaches to sustainable healthcare solutions.

BIBLIOGRAPHY

[1] A. R. Hinman, “Global progress in infectious disease control,” 1998.

[2] P. J. Hotez *et al.*, “Control of Neglected Tropical Diseases,” 2007. [Online]. Available: www.nejm.org

[3] S. Giunti, N. Andersen, D. Rayes, and M. J. de Rosa, “Drug discovery: Insights from the invertebrate Caenorhabditis elegans,” *Pharmacology Research and Perspectives*, vol. 9, no. 2. John Wiley and Sons Inc, Apr. 01, 2021. doi: 10.1002/prp2.721.

[4] H. C. S. Chan, H. Shan, T. Dahoun, H. Vogel, and S. Yuan, “Advancing Drug Discovery via Artificial Intelligence,” *Trends in Pharmacological Sciences*, vol. 40, no. 8. Elsevier Ltd, pp. 592–604, Aug. 01, 2019. doi: 10.1016/j.tips.2019.06.004.

[5] E. V. Minikel, J. L. Painter, C. C. Dong, and M. R. Nelson, “Refining the impact of genetic evidence on clinical success,” *Nature*. Jun. 29, 2023. doi: 10.1101/2023.06.23.23291765.

[6] H. Dowden and J. Munro, “Trends in clinical success rates and therapeutic focus,” *Nature reviews. Drug discovery*, vol. 18, no. 7. NLM (Medline), pp. 495–496, Jul. 01, 2019. doi: 10.1038/d41573-019-00074-z.

[7] J. P. Jourdan, R. Bureau, C. Rochais, and P. Dallemagne, “Drug repositioning: a brief overview,” *Journal of Pharmacy and Pharmacology*, vol. 72, no. 9. Blackwell Publishing Ltd, pp. 1145–1151, Sep. 01, 2020. doi: 10.1111/jphp.13273.

[8] N. Novac, “Challenges and opportunities of drug repositioning,” *Trends in Pharmacological Sciences*, vol. 34, no. 5. pp. 267–272, May 2013. doi: 10.1016/j.tips.2013.03.004.

[9] C. Ballard *et al.*, “Drug repositioning and repurposing for Alzheimer disease,” *Nature Reviews Neurology*, vol. 16, no. 12. Nature Research, pp. 661–673, Dec. 01, 2020. doi: 10.1038/s41582-020-0397-4.

[10] M. Lotfi Shahreza, N. Ghadiri, S. R. Mousavi, J. Varshosaz, and J. R. Green, “A review of network-based approaches to drug repositioning,” *Briefings in bioinformatics*, vol. 19, no. 5. NLM (Medline), pp. 878–892, Sep. 28, 2018. doi: 10.1093/bib/bbx017.

[11] H. Xue, J. Li, H. Xie, and Y. Wang, “Review of drug repositioning approaches and resources,” *International Journal of Biological Sciences*, vol. 14, no. 10. Ivyspring International Publisher, pp. 1232–1244, Jul. 13, 2018. doi: 10.7150/ijbs.24612.

[12] Y. Ko, “Computational drug repositioning: Current progress and challenges,” *Applied Sciences (Switzerland)*, vol. 10, no. 15. MDPI AG, Aug. 01, 2020. doi: 10.3390/app10155076.

[13] Y. Meng, C. Lu, M. Jin, J. Xu, X. Zeng, and J. Yang, “A weighted bilinear neural collaborative filtering approach for drug repositioning,” *Briefings in Bioinformatics*, vol. 23, no. 2, Mar. 2022, doi: 10.1093/bib/bbab581.

[14] T. N. Jarada, J. G. Rokne, and R. Alhajj, “A review of computational drug repositioning: Strategies, approaches, opportunities, challenges, and directions,”

Journal of Cheminformatics, vol. 12, no. 1. BioMed Central, Jul. 22, 2020. doi: 10.1186/s13321-020-00450-7.

[15] A. S. Brown and C. J. Patel, "A standard database for drug repositioning," *Scientific Data*, vol. 4, Mar. 2017, doi: 10.1038/sdata.2017.29.

[16] H. C. Cousins, G. Nayar, and R. B. Altman, "Computational Approaches to Drug Repurposing: Methods, Challenges, and Opportunities," *Annual Review of Biomedical Data Science Downloaded from www.annualreviews.org. Guest (guest)*, 2025, doi: 10.1146/annurev-biodatasci-110123.

[17] G. Bakal, H. Kilicoglu, and R. Kavuluru, "Non-Negative Matrix Factorization for Drug Repositioning: Experiments with the repoDB Dataset."

[18] F. Urbina, A. C. Puhl, and S. Ekins, "Recent advances in drug repurposing using machine learning," *Current Opinion in Chemical Biology*, vol. 65. Elsevier Ltd, pp. 74–84, Dec. 01, 2021. doi: 10.1016/j.cbpa.2021.06.001.

[19] G. Bakal, P. Talari, E. v. Kakani, and R. Kavuluru, "Exploiting semantic patterns over biomedical knowledge graphs for predicting treatment and causative relations," *Journal of Biomedical Informatics*, vol. 82, pp. 189–199, Jun. 2018, doi: 10.1016/j.jbi.2018.05.003.

[20] F. Cheng *et al.*, "Network-based approach to prediction and population-based validation of in silico drug repurposing," *Nature Communications*, vol. 9, no. 1, Dec. 2018, doi: 10.1038/s41467-018-05116-5.

[21] X. Sun, X. Jia, Z. Lu, J. Tang, and M. Li, "Drug repositioning with adaptive graph convolutional networks," *Bioinformatics*, vol. 40, no. 1, Jan. 2024, doi: 10.1093/bioinformatics/btad748.

[22] H. Kilicoglu, D. Shin, M. Fiszman, G. Rosemblat, and T. C. Rindflesch, "SemMedDB: A PubMed-scale repository of biomedical semantic predications," *Bioinformatics*, vol. 28, no. 23, pp. 3158–3160, Dec. 2012, doi: 10.1093/bioinformatics/bts591.

[23] H. Kilicoglu, G. Rosemblat, M. Fiszman, and D. Shin, "Broad-coverage biomedical relation extraction with SemRep," *BMC Bioinformatics*, vol. 21, no. 1, May 2020, doi: 10.1186/s12859-020-3517-7.

[24] Kashyap, V.: The umls® semantic network and the semantic web. In: AMIA Annual Symposium Proceedings. vol. 2003, p. 351. American Medical Informatics Association (2003)

[25] X. Jing, "The unified medical language system at 30 years and how it is used and published: Systematic review and content analysis," *JMIR Medical Informatics*, vol. 9, no. 8. JMIR Publications Inc., Aug. 01, 2021. doi: 10.2196/20675.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Jan. 2013, [Online]. Available: http://arxiv.org/abs/1301.3781

[27] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Jul. 2016, [Online]. Available: http://arxiv.org/abs/1607.04606

[28] N. V. Otten, "What Is FastText? Compared To Word2Vec & GloVe [How To Tutorial In Python]," *Spot Intelligence*, Dec. 5, 2023. [Online]. Available: https://spotintelligence.com/2023/12/05/fasttext/. [Accessed: Apr. 7, 2025].

[29] J. Bromley *et al.*, "Signature Verification using a 'Siamese' Time Delay Neural Network," 1993. doi: 10.1142/S0218001493000339.

[30] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition," 2015.

[31] G. Bajaj *et al.*, "Evaluating Biomedical Word Embeddings for Vocabulary Alignment at Scale in the UMLS Metathesaurus Using Siamese Networks," 2022. [Online]. Available: https://uts.nlm.nih.gov/uts/

[32] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," 2011.

[33] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Feb. 2015, [Online]. Available: http://arxiv.org/abs/1502.03167

[34] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.06455

[35] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[36] A. Al-Kababji, F. Bensaali, and S. P. Dakua, "Scheduling Techniques for Liver Segmentation: ReduceLROnPlateau Vs OneCycleLR," Feb. 2022, [Online]. Available: http://arxiv.org/abs/2202.06373

[37] F. Pedregosa *et al.*, "3.1. Cross-validation: evaluating estimator performance," *scikit-learn: Machine Learning in Python*, [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. [Accessed: Apr. 7, 2025]

[38] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," 2012. [Online]. Available: http://scikit-learn.sourceforge.net.

[39] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," Jun. 2012, [Online]. Available: http://arxiv.org/abs/1206.2944

[40] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization," Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.06560

[41] M. Jaderberg *et al.*, "Population Based Training of Neural Networks," Nov. 2017, [Online]. Available: http://arxiv.org/abs/1711.09846

CURRICULUM VITAE

2018 – 2022	B.Sc., Industrial Engineering, Abdullah Gül University, Kayseri, TURKEY
2023 – 2025	M.Sc., Computer Engineering, Abdullah Gül University, Kayseri, TURKEY