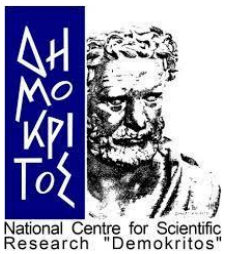

MSc in Data Science Thesis

Drug to Drug Interaction Multiclass Prediction on Biomedical Literature Knowledge Graphs

`

VOTTAS MARIOS

Supervisors:
Dr. Anastasia Krithara
Researcher (grade C)
&
Dr. Aisopos Fotis
Post-Doctoral Researcher

Athens, December 2024

Abstract

Lung cancer remains a major global health concern, requiring continued advances in therapeutic approaches. The complex landscape of lung cancer therapies introduces the potential for complex drug-drug interactions (DDIs), affecting patient outcomes and treatment efficacy. The conventional study of DDIs, based on experimental methods, is limited by time, cost and scope. Recent advances in graph databases and machine learning provide an opportunity to accelerate and enhance DDI prediction.

This MSc thesis explores the fusion of graph databases and machine learning to predict drug interactions in lung cancer. By leveraging the interconnection of graph databases, the relationships between lung cancer drugs, target proteins, pathways, and side effects are encapsulated, enabling a structured and scalable representation of complex interactions.

The central goal is to devise an accurate machine learning framework for DDI prediction among lung cancer drugs. Analyzing the structure of the graph database facilitates the extraction of key features and patterns that facilitate interaction prediction, including competitive effects. A comprehensive knowledge graph is constructed, which contains various aspects of lung cancer drugs and their interactions. Machine learning algorithms are used to build predictive models capable of understanding complex DDI patterns, and the results of this research contribute to personalized medicine and deepen the understanding of lung cancer treatments.

The following chapters present the methodology, including data acquisition, modeling techniques, and experimental results. By evaluating the effectiveness and feasibility of using graph databases and machine learning in predicting drug interactions for lung cancer, this study seeks to improve the landscape of drug interaction prediction and promote more effective treatments for lung cancer.

Table of Contents

List Of Tables

List Of Images

List of Abbreviations

CSV	Comma-Separated Values
XML	eXtensible Markup Language
DDI	Drug-Drug Interaction
API	Application Programming Interface
JSON	JavaScript Object Notation
SQL	Structured Query Language
NoSQL	Not Only SQL
KG	Knowledge Graph

Chapter 1: Introduction

Drug-Drug Interactions (DDIs) occur when the effects of one drug are affected by another. In oncology, where polypharmacy is common, the potential for DDIs is significantly heightened, as patients are often prescribed multiple drugs to manage cancer progression, symptom control, and associated comorbidities. Understanding the types of DDIs and their associated toxicities is crucial for preventing adverse outcomes in cancer treatment, especially in vulnerable populations such as lung cancer patients, who may already have compromised health statuses.

DDIs can be broadly classified into two categories: pharmacokinetic and pharmacodynamic interactions [10].

- Pharmacokinetic interactions happen when a drug affects the absorption, distribution or metabolism of another drug used at the same time. These interactions can lead to changes in drug concentrations, which may result in either sub-therapeutic levels or toxicity. For example, if Drug A inhibits the metabolism of Drug B by affecting liver enzymes such as cytochrome P450 (CYP), the concentration of Drug B may increase, leading to an increased risk of toxic side effects. Conversely, if a drug accelerates the metabolism or excretion of another, the therapeutic efficacy of the affected drug may be reduced, compromising treatment outcomes.
- Pharmacodynamic interactions occur when two drugs act on the same or related biological pathways, leading to additive, synergistic, or antagonistic effects. Synergistic interactions may be beneficial when two drugs work together to enhance therapeutic outcomes, such as in combination therapies for cancer. However, antagonistic interactions, where one drug reduces the effectiveness of another, can undermine treatment efforts. For example, combining two drugs that have opposing effects on cell signaling pathways may diminish the efficacy of the intended cancer treatment [11].

In the context of lung cancer treatment, the consequences of DDIs can be severe. Toxicities arising from DDIs are often systemic and can affect various organs and biological systems. Common DDI-induced toxicities include:

- Hematologic toxicities: Certain drug combinations can lead to bone marrow suppression, resulting in conditions such as anemia, thrombocytopenia, or neutropenia, which compromise the immune system and increase the risk of infections.
- Cardiotoxicity: Some DDIs may exacerbate cardiovascular conditions by increasing the risk of arrhythmias, hypertension, or heart failure, particularly in patients with pre-existing cardiovascular disease.

- Hepatotoxicity and nephrotoxicity: DDIs that impair liver or kidney function can lead to the accumulation of toxic drug metabolites, further exacerbating organ damage and affecting overall patient health.

Beyond the immediate toxicological effects, DDIs can significantly influence mortality rates in lung cancer patients. Inappropriate or unintended drug interactions can not only reduce the effectiveness of cancer therapies but also lead to life-threatening adverse events, such as severe infections, organ failure, or hemorrhage. For instance, a DDI that increases the risk of bleeding could be fatal in a patient who is already at risk of thrombocytopenia due to chemotherapy.

1.1 Inspiration

Managing DDIs in cancer patients requires a complex approach, integrating oncology, pharmacology, and clinical practice to mitigate risks and optimize therapeutic outcomes. With the rise of personalized medicine, there is a growing recognition of the need to predict and prevent DDIs based on individual patient profiles and cancer types. Leveraging computational tools such as graph databases and machine learning has the potential to significantly improve the prediction and management of DDIs, ultimately reducing toxicity-related mortality and enhancing patient safety.

Lung cancer remains one of the leading causes of cancer-related deaths worldwide, necessitating continued advances in therapeutic solutions. As the complexity of lung cancer treatment strategies increases, so does the potential for drug-drug interactions (DDIs), which can significantly affect patient outcomes and the overall efficiency of treatments. Understanding and predicting these interactions is vital to ensuring patient safety and optimizing therapeutic outcomes.

Traditionally, the study of DDIs has been based on experimental methods, such as in vitro studies and clinical trials. While these approaches have contributed valuable insights, they are often time-consuming, expensive, and limited in their ability to capture the full range of possible interactions. In recent years, the emergence of graph databases and machine learning techniques has provided new opportunities to accelerate and enhance DDI prediction.

1.2 Aim of this Thesis

This MSc thesis aims to explore the application of graph databases and machine learning in predicting drug interactions specifically in the context of lung cancer treatment. By exploiting the comprehensive and interconnected nature of graph databases, we can represent the relationships between lung cancer drugs and their associated factors, such as target proteins, pathways, and side effects. This approach allows us to capture the complexity of drug interactions in a structured and scalable manner.

The primary goal of this research is to develop an efficient and accurate machine learning framework that can predict DDIs between lung cancer drugs. By analyzing the underlying structure and properties of the graph database, we can extract important features and patterns that help predict interactions. This predictive model will identify potential interactions that could cause adverse effects.

By incorporating this wealth of information into a graph database, we can construct a comprehensive knowledge graph that includes various aspects of lung cancer drugs and their relationships. Leveraging this knowledge graph, we will apply machine learning algorithms, to train predictive models capable of capturing the complex patterns of DDIs.

The results of this research will contribute to the field of personalized medicine and enhance our understanding of lung cancer treatment strategies. Ultimately, this thesis aims to provide healthcare professionals with a powerful tool to identify potential drug interactions in lung cancer patients, enabling them to make informed decisions about treatment plans and improve patient safety and outcomes.

1.3 Chapter Presentation

In this MSc thesis, a comprehensive examination of Prediction Graphs is undertaken in the area of Drug to Drug Interactions. This introductory chapter guides readers to the discussions and analyzes what unfolds in the next chapters. The primary goal of this thesis is to advance the collective understanding of DDI predictions by contributing meaningfully to the ongoing research field.

In the following chapters, a structured exploration of the research journey will be presented. Central to this research is Chapter 2, where an in-depth review of the relevant literature is conducted. This chapter includes concepts, and previous research relevant to this study. This chapter is the baseline for our research, presenting the theoretical background that this thesis is based on.

Following this foundation, the next chapter, transitions into the practical aspects of this research. Here, it presents the methodology used for the experiments performed. A detailed description of the research design, data collection and storage techniques and tools used in this research is presented. The methodology is crucial to ensure the validity and reliability of the experiments that follow.

Chapter 4, represents the hands-on effort of this research. This chapter provides a thorough description of the experiments conducted. The chapter clarifies the complications of the

experimental setup, data collection, and any significant challenges encountered during the research process.

Following, the next chapter, moves from data collection and experiment presentation to result analysis. The implications of these findings are discussed, providing a perspective on their significance and relevance to the academic community.

Finally, Chapter 6 concludes this entire research. Its titled "Conclusions and Future Extensions" this chapter summarizes the key insights gained from this analysis. It draws conclusions based on the research objectives from Chapter 1, and provides a comprehensive perspective on the contributions of this study to DDI Interactions using Prediction Graphs.
Furthermore, this chapter examines possibilities for further research and identifies areas that this thesis could not tackle. It serves as an invitation to future scholars to build upon the foundation laid here.

Chapter 2: Background and Related works

This thesis draws substantial inspiration from the research presented in the paper "Drug-Drug Interaction Prediction in a Biomedical Literature Knowledge Graph" (refer DDI AIME paper). Based on the fundamental framework formulated in the aforementioned work, this thesis attempts to extend the scope of the DDI AIME document case study by moving it from a binary classification example to a more complex multi-class classification problem.

2.1 Knowledge Graphs

A Knowledge Graph (KG) is a structured representation of knowledge that connects entities (nodes) and relationships (edges) in a graph format. These entities can represent real-world concepts like people, organizations, drugs, genes, or diseases, while relationships capture the interactions or associations between them. Unlike traditional databases, a KG allows for more flexibility, semantically enriched representations of complex and interconnected data [21]. This flexibility makes them powerful tools for managing and analyzing large, diverse datasets, such as those found in the biomedical field.

Knowledge graphs use concepts from graph theory, artificial intelligence, and semantic web technologies to represent knowledge in a machine-readable way, enabling computers to better "understand" the data. The ability to model complex relationships within data makes them especially useful in fields like biomedicine, where entities such as genes, proteins, drugs, and diseases interact in complicated ways.

A knowledge graph is typically composed of:

1. **Nodes**: Represent entities or objects such as drugs and diseases in biomedical contexts.
2. **Edges**: Represent the relationships between nodes. For example, a "treats" relationship might connect a drug node to a disease node.
3. **Properties**: Metadata or attributes associated with both nodes and edges, such as the efficacy of a drug.

2.1.1 Biomedical Knowledge Graphs

In the biomedical domain, knowledge graphs are gaining traction for their ability to integrate vast amounts of heterogeneous data from multiple sources, such as scientific publications, clinical trials and drug databases. By connecting these datasets, biomedical KGs provide a structure that can help in hypothesis generation, drug discovery, and precision medicine.

Examples of Biomedical Knowledge Graphs

1. **Hetionet**: Hetionet is a knowledge graph that integrates information from various biomedical databases, representing relationships among diseases, genes, drugs, side effects, and symptoms [22]. It allows researchers to perform drug repurposing and predict new drug-disease relationships by analyzing patterns in the graph.
2. **PharmGKB**: The Pharmacogenomics Knowledge Base (PharmGKB) focuses on the relationship between human genetic variation and drug response [23]. It integrates data from genetic studies, drug-gene interactions, and disease associations into a comprehensive knowledge graph.
3. **OpenBioLink**: OpenBioLink is a publicly available biomedical knowledge graph that includes relationships between drugs, diseases, genes, and other biomedical entities [24]. It supports tasks like link prediction and network-based analysis for drug discovery and gene function prediction.

2.1.2 Tasks in Biomedical Knowledge Graphs

Knowledge graphs allow for a wide variety of tasks that are particularly useful in biomedical research. Some of the key tasks include:

1. Error Detection

Given the large and complex nature of biomedical data, errors and inconsistencies are common [25]. These errors may arise from inaccurate data entry, outdated information, or incomplete datasets. Knowledge graphs can help detect errors through various methods:

- **Consistency Checks**: The graph structure allows for logical consistency checks, where relationships between entities can be validated. For example, if a drug is labeled as both "treats" and "causes" the same disease, the KG can flag this contradiction.
- **Outlier Detection**: By analyzing patterns across similar entities, KGs can identify anomalies or outliers. For example, if a drug shows a highly unusual interaction with a common gene, it might suggest a data entry error.
- **Inference-based Error Detection**: Many KGs are equipped with reasoning engines that can infer new relationships based on existing data. If a new relationship contradicts inferred knowledge, it might indicate an error in the dataset.

2. Link Prediction

Link prediction refers to the task of predicting missing or unknown relationships between entities in the graph [26]. This is particularly useful in biomedical knowledge graphs, where discovering new connections can lead to breakthroughs in medical research.

- **Drug Repurposing**: Link prediction can suggest new uses for existing drugs by identifying previously unknown drug-disease relationships. By analyzing patterns of

known drug-disease pairs, KGs can recommend drugs that might be effective for treating diseases not previously associated with them.

- **Gene-Disease Associations**: Link prediction can help identify new gene-disease associations, potentially uncovering the genetic basis of diseases or new therapeutic targets for drug development.
- **Drug-Drug Interactions**: Knowledge graphs can predict potential interactions between drugs, which is crucial for preventing adverse effects and improving patient safety.

3. Graph-Based Querying and Data Integration

Biomedical knowledge graphs enable advanced querying capabilities, where researchers can ask complex questions [27]. For example, one could query a KG to find all drugs that target a specific gene, which is known to be associated with a certain disease. This ability is particularly important in the biomedical field, where data often comes from fragmented sources.

- **Semantic Search**: Unlike traditional keyword-based searches, semantic searches in a KG take into account the meaning of terms and the relationships between entities, enabling more accurate and meaningful results.
- **Cross-Dataset Integration**: A KG can integrate data from multiple datasets, ensuring that a query considers information from all relevant sources. This helps eliminate biases that come from focusing on only one data source.

2.2 Drug-Drug Interaction Prediction on a Biomedical Literature Knowledge Graph

This work focuses on the use of Knowledge Graphs to extract valuable insights from the disease-specific literature, focusing specifically on the discovery of potential drug-drug interactions (DDIs) [1]. The Knowledge Graph is constructed through natural language processing and semantic indexing of biomedical publications, combining information from various open resources. The semantic paths connecting different drugs within the Graph are then extracted and clustered into feature vectors, effectively representing pairs of drugs. The chapter uses a classifier trained on known interactions derived from a manual drug database, which serves as a gold standard, to identify and discover new potential drug interactions.

The methodology is evaluated through two distinct use cases, namely Alzheimer's disease and lung cancer, demonstrating the flexibility and applicability of the approach in different medical domains. The performance of the system is evaluated by comparing it with competing graph embedding approaches. In particular, the Knowledge Graph-based approach not only outperforms these competing methods, but also demonstrates its effectiveness in identifying novel drug-drug interactions, a validation performed retrospectively.

This paper closely aligns with the theme of the thesis, as this thesis is the continuance of the paper analyzed in this chapter. While the paper focuses on binary classification tasks, whether a combination of drugs have or not adverse effects, this paper takes that task a step forward by adding the specific adverse effects on the equation.

2.3 BioDKG–DDI: predicting drug–drug interactions based on drug knowledge graph fusing biochemical information

Co-administration of drugs as an effective strategy for managing complex diseases and the critical need for accurate prediction of potential adverse drug-drug interactions (DDIs) resonate deeply with the central theme of this thesis. Given the extensive unknown interactions between drugs, computational studies, as demonstrated by the authors' work on the BioDKG–DDI [2] model, are becoming important in overcoming the limitations associated with traditional liquid lab experiments. This shared focus on computational methodologies underscores a common goal to advance predictive capabilities for DDI.

The authors of the BioDKG–DDI model identify a critical gap in existing approaches, particularly the limitations associated with neglecting multiscale features and the inability to predict interactions between new drugs. This resonates with the challenges faced by the current thesis, highlighting the necessity for models that can accommodate different characteristics and adapt to emerging associations.

Incorporating a biomedical literature knowledge graph (DKG) into the BioDKG–DDI model aligns with the primary goals of the current thesis. Similarly, the attention mechanism within a deep neural network, fusing molecular structure features, drug global association representations, and drug functional similarity features, reflects the multifaceted approach required for a comprehensive understanding of drug interactions—one that is directly relevant to the objectives of this research.

The proposed negative selection method to enhance robustness and stability in the BioDKG–DDI model serves as a notable idea with potential implications for the current thesis. Robustness and reliability in predicting drug interactions are common themes between the two works, as evidenced by the authors' emphasis on validation procedures and comparative experiments.

Moreover, the authors' claim that the BioDKG–DDI model can serve as a beneficial adjunct to experimental procedures aligns with the broader goal of computational models as complementary tools in drug interaction prediction, as pursued in this thesis.

Essentially, the BioDKG–DDI model, with its emphasis on computational prediction, integration of diverse features and attention to robustness, not only shares common ground with the current

thesis but also provides a potential avenue for inspiration, methodology validation and comparative analysis. The parallels between the two works highlight the importance of the BioDKG–DDI model in the context of advanced research in drug interaction prediction, offering valuable insights that can contribute to the theoretical framework and practical implications of this thesis.

2.4 Prediction of Drug-Drug Interaction Using an Attention-Based Graph Neural Network on Drug Molecular Graphs

This paper also studies drug-drug interactions (DDI), focusing in particular on the challenges posed by the prevalence of polypharmacy in the treatment of complex diseases [3].

The proposed methodology, GNN-DDI, introduces a new approach that uses a five-layer graph attention network. This network aims to construct low-dimensional feature representations for each drug by analyzing its chemical molecular graph. These representations are then concatenated for each drug pair and fed into a multilayer perceptron (MLP) predictor, ultimately yielding a DDI prediction score. In particular, GNN-DDI distinguishes itself by addressing limitations observed in existing computational methods. It is designed to predict potential DDIs not only between known drugs within the DDI network but also between drugs within the network and new drugs outside the DDI network.

While this thesis relies on the Random Forest classifier for multiclass prediction in drug interactions, the methodology introduced in GNN-DDI presents a complementary and innovative perspective. By leveraging graph neural network techniques, GNN-DDI provides an alternative approach to predict drug-drug interactions, addressing limitations observed in traditional computational methods. The utilization of a Random Forest classifier in the thesis is instrumental for its multiclass prediction capabilities, whereas the GNN-DDI methodology contributes additional depth by offering insights into potential interactions among known drugs and extending predictions to new drugs.

Despite the methodological differences, both the Random Forest classifier in this thesis and GNN-DDI share the common goal of enhancing polypharmacy safety. The ability of GNN-DDI to predict interactions involving new drugs aligns with the comprehensive prediction models sought in the thesis. Furthermore, the emphasis on interpretability and the exploration of drug substructures, as demonstrated by GNN-DDI, resonates with the broader objective of understanding complex interaction mechanisms within biomedical literature knowledge graphs. The combined insights from both methodologies contribute to a more holistic understanding of drug interactions, aligning with the thesis's commitment to advancing the state of the art in multiclass prediction within the biomedical domain.

2.5 On the road to explainable AI in drug-drug interactions prediction: A systematic review

This paper presents the challenge of adverse DDIs arising from instances of polypharmacy in multi disease treatment approaches [4]. Artificial intelligence (AI) has emerged as a powerful tool for DDI prediction, offering support to clinicians in making informed pharmacotherapy decisions. However, the black-box nature of AI models raises concerns about their reliability in critical medical tasks.

This paper advocates the integration of explainable artificial intelligence (XAI) mechanisms into DDI prediction models to improve their transparency. XAI serves as a key element in promoting safety and clarity by clarifying how decisions are made in AI models, particularly in tasks as critical as DDI predictions. The review provides a comprehensive overview of AI-based DDI prediction, including publicly available sources for AI-DDI studies, methodologies used in data manipulation and feature preprocessing, and the incorporation of XAI mechanisms to build confidence in AI models for DDI predictions . The discussion expands on various modeling methods used in AI-DDI studies, highlighting the need for interpretable AI in the context of polypharmacy decisions.

Furthermore, the paper critically assesses the limitations of XAI in the context of DDIs and explores possible future directions. By addressing these limitations and providing insights into the evolving landscape of XAI in DDI prediction, the review aims to contribute to the ongoing debate about the role of explanation in enhancing the reliability and applicability of AI models in clinical settings.

This paper is significantly aligned with the central theme of the thesis, but with a particular focus on incorporating explainable artificial intelligence (XAI). While the thesis focuses primarily on the use of a Random Forest classifier for multi-class prediction of drug interactions, the literature reviewed in this paper highlights the importance of transparency and interpretability in AI models, particularly in the critical area of DDI predictions.

2.6 DPDDI: A Deep Predictor for Drug-Drug Interactions

This paper presents a method for predicting drug-drug interactions (DDIs) using a graph convolution network (GCN) and a deep neural network (DNN) [14] . The authors address the growing need to predict DDIs due to the increasing use of multi-drug treatments, which can lead to adverse reactions or toxicity.

The DPDDI method bypasses the reliance on chemical or biological properties of drugs, which are often costly or unavailable, by leveraging the network structure of known drug interactions.

The GCN is used to learn low-dimensional feature representations of drugs from a DDI network, capturing the topological relationships between drugs. The DNN concatenates the features of drug pairs and uses these vectors to predict potential interactions.

The model outperforms several state-of-the-art methods in cross-validation tests and shows robust performance across datasets of varying sizes. The authors also demonstrate through case studies that DPDDI can effectively identify previously unobserved DDIs. Additionally, the method's ability to detect interactions without requiring extensive drug property data makes it scalable and efficient for real-world applications.

The approach shows promise not only for DDI prediction but also for related tasks, such as identifying unexpected side effects or guiding drug combination therapies.

2.7 Analysis of the Impact of Negative Sampling on Link Prediction in Knowledge Graphs

This paper investigates the role of different negative sampling strategies on the performance of link prediction tasks in knowledge graphs (KGs) [15]. KGs, such as Freebase and WordNet, are widely used for storing relational data but often suffer from incompleteness, necessitating link prediction techniques to infer missing connections. The paper evaluates several state-of-the-art KG embedding models, including Rescal, TransE, DistMult, and ComplEx, by using benchmark datasets FB15k and WN18.

A key component of learning these embeddings is contrasting positive and negative samples, where the negative samples are synthetic instances generated from the KG. The authors conduct a comprehensive empirical study comparing common negative sampling methods (e.g., random sampling, corrupting positive instances) and propose two approaches: nearest neighbor sampling and near-miss sampling. Their analysis reveals that the effectiveness of negative sampling methods varies depending on the dataset and model architecture. For instance, corrupting positive samples works well on WordNet, whereas near-miss sampling shows better performance on Freebase.

The study highlights the importance of carefully selecting negative sampling methods to optimize the link prediction results in different contexts, ultimately contributing to more efficient KG completion strategies.

2.8 3DGT-DDI: 3D Graph and Text-based Neural Network for Drug-Drug Interaction Prediction

This paper introduces a deep learning model for predicting drug-drug interactions (DDIs) by combining 3D molecular graphs with textual data [16]. Traditional approaches often rely on two-dimensional representations of drug molecules, which lack spatial information crucial for

accurately predicting DDIs. To address this limitation, the authors developed 3DGT-DDI, which integrates 3D molecular structures with drug description texts using a combination of 3D graph neural networks (GNNs) and a pre-trained SCIBERT text model.

The 3DGT-DDI model extracts features from both the spatial conformation of drug molecules and the textual descriptions of drug interactions. It leverages the attention mechanism for better feature extraction and introduces a hidden layer fusion technique to combine the 3D and text-based representations. Experiments conducted on the DDIExtraction-2013 dataset show that 3DGT-DDI outperforms several state-of-the-art models with an F1-macro score of 84.48%. Additionally, the model's performance and explainability are enhanced by visualizing attention weights at the atomic and text level, offering insights into which molecular substructures contribute to predicted DDIs.

This work demonstrates the advantage of incorporating 3D molecular information in DDI prediction tasks, making 3DGT-DDI a promising tool for drug safety research and drug discovery. The source code for 3DGT-DDI is made available, providing opportunities for further development and application in biomedical research.

2.9 DDI-GCN: Drug-drug interaction prediction via explainable graph convolutional networks

This paper focuses on predicting drug-drug interactions (DDIs) using an explainable Graph Convolutional Network (GCN) approach [17]. The model integrates both molecular structure data and interaction information to enhance prediction accuracy and interpretability. By utilizing explainable GCNs, the method enables the identification of critical features contributing to DDIs. The model achieves strong performance, with results showing significant improvements over traditional methods in terms of ROC and F1 scores.

Chapter 3: Methodology

This chapter serves as a comprehensive guide to the methodologies used in the study of multi-class drug to drug interaction (DDI) predictions. By providing a detailed description of the methods, techniques and tools used in this research, this chapter aims to enhance the transparency, reproducibility and reliability of the study findings.

3.1 Methodology planning

This chapter outlines the key components and methodology of this project. The following figure illustrates the approach taken.

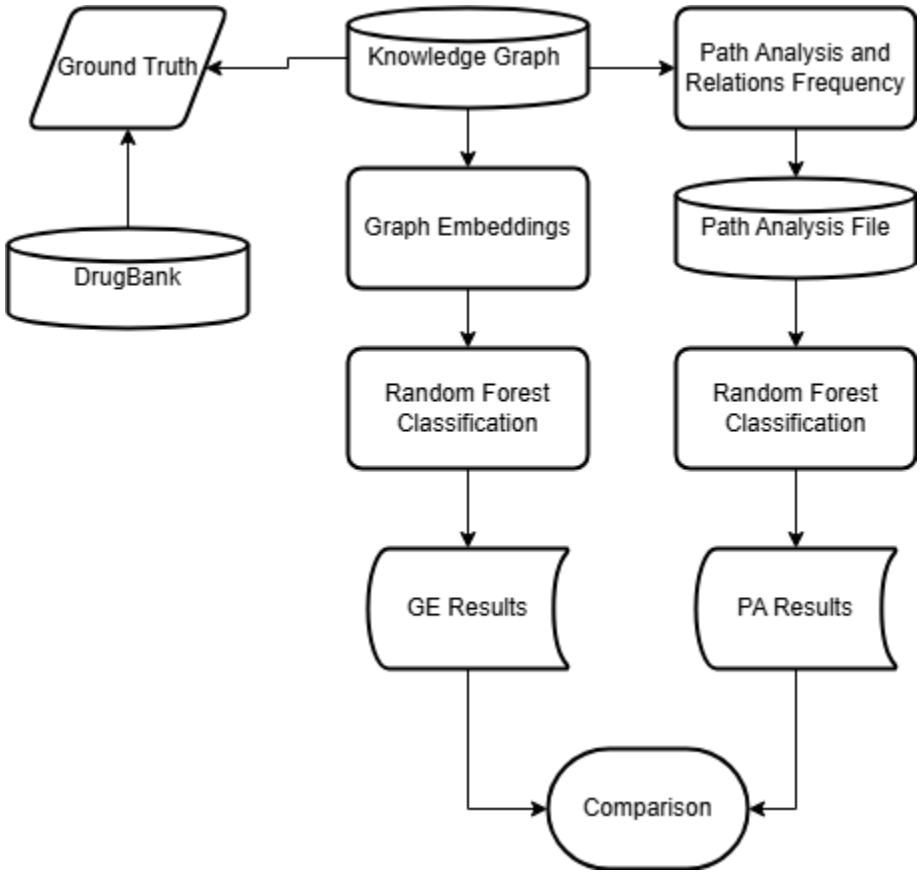

Figure 1. Thesis Methodology

The process begins with the creation of the Knowledge Graph which is formed from analyzing biomedical literature, and serves as the foundation for generating the Path Analysis file (details of this procedure are provided in a later chapter). Additionally, using DrugBank [6], the Knowledge Graph is enriched with the Ground Truth for our experiments. Next, Graph

Embeddings are applied to the Knowledge Graph to transform its data into meaningful representations for our model. Finally, both approaches, graph embeddings and path analysis, are evaluated using a random forest classifier, and their results are compared.

3.2 Data Preparation

The foundation of this thesis lies in constructing a knowledge graph, designed to support subsequent data processing and modeling tasks. This knowledge graph will serve as a structured dataset, enabling the use of graph embeddings to enhance model inputs and optimize results. Following its creation, a path analysis will be conducted on the knowledge graph to extract relevant connections and patterns, generating a comprehensive CSV file. This file will form the input for further experimental analyses, facilitating deeper insights and supporting the broader objectives of this study.

3.2.1 Knowledge Graph Creation

This method for creating a domain-specific biomedical knowledge graph draws information from extensive biomedical literature databases such as PubMed [30] and MEDLINE [31]. These databases, rich in structured biomedical data and semantic annotations, serve as the basis for targeted data mining. Specifically, queries focused on diseases like lung cancer retrieve relevant articles, which include titles, abstracts, and curated subject annotations (eg, MeSH terms). Full-text articles from PubMed Central augment this data set where accessible.

To structure the biomedical knowledge within the graph, a processing pipeline leverages the Unified Medical Language System (UMLS) [12]. UMLS standardizes terminology by mapping various expressions of the same biomedical concept into a unified vocabulary, also linking concepts through defined semantic relationships. This pipeline uses the SemRep [13] tool, a UMLS-based NLP framework, to identify semantic triples (subject-predicate-object) within biomedical text. These triples, such as a chemical that "cures" a disease or a drug that "inhibits" a gene, provide fundamental connections between concepts.

To further structure the graph, additional relationship types are introduced: MENTIONED_IN and HAS_MESH. MENTIONED_IN indicates concept co-occurrence in the same article, helping to quantify term associations that may indicate relevance between concepts. HAS_MESH associates articles with MeSH annotations, embedding expert-curated biomedical knowledge into the graph. These custom relationships support deeper insights by linking terms based on both co-occurrence and domain-specific annotations.

The graph incorporates hierarchical relationships from biomedical ontologies available in Open Biomedical Ontologies (OBO) format, including Gene Ontology and Disease Ontology. These

additional relations enrich the graph by adding is_a hierarchies and related taxonomies, expanding the analytic capabilities of the graph. The completed graph is stored in Neo4j [7], optimizing the connectivity structure for fast data retrieval and manipulation.

3.2.2 Path Analysis

The previous step produces a multi-relational, directed knowledge graph (KG), setting the foundation for identifying possible interactions among biomedical entities. Specifically, for a pair of drugs, the problem of drug-drug interaction prediction can be framed as a link prediction task within the KG. Here, the task is framed as a supervised learning problem, where data samples are derived from drug pairs within the knowledge graph, aiming to identify pairs of drugs that interact.

To generate these data samples, DrugBank is utilized as a source of known interactions (ground truth), and drugs are mapped to corresponding UMLS entities. For example, a drug is mapped under the UMLS schema to two entities. We assume that if DrugBank lists an interaction between two drugs, their respective UMLS entities are also considered to interact. This mapping process is illustrated in the following figure.

Subsequently, we aggregate all feasible paths within the KG that connect each drug pair. Let $E = \{e_1 e_2, \ldots, e_M\}$ represent all relations within the graph. Given two drug nodes d_1 and d_2, let π^l denote a path of length l linking d_1 and d_2. This path π comprises a sequence of relations beginning at d_1 and ending at d_2, represented as $d_1 e_0 e_1 e_2 \ldots e_{l-1} d_2$. In this approach, the path length is restricted to l<3 after observing that paths of greater length resulted in lower-quality interactions, likely due to the high interconnectedness of the graph. Therefore, each path between a drug pair is represented as $\pi = e_0 e_1 e_2$.

For any two drugs d_1 and d_2, let $\Pi = \{\pi^l_1, \pi^l_2, \ldots, \pi^l_{Nd_1 d_2}\}$ represent the set of all possible paths between them. Figure 2 illustrates examples of such paths identified in the KG for two drugs. After retrieving the paths, each path undergoes feature extraction to create a feature representation.

The UMLS Semantic Network provides 35 unique relation types, which were further refined by merging semantically similar relations, reducing the set from an original 55. These 35 relations are then one-hot encoded to serve as features for each possible path hop. With a maximum of l=3 hops, each feature vector contains 3×35=105 features. Each feature value is binary (0 or 1), indicating the presence of a specific relation in each hop.

For a path $\pi^{l}_{d_1 d_2}$ between drugs d_1 and d_2, the resulting feature vector is denoted as $x_i = [c_{r1}, c_{r2}, ..., c_{r105}]$, where c_{r_j} represents a binary feature as previously described. For paths with a length $m < l = 3$ the last $(l - m)\, x\, 35$ elements of the feature vector are set to zero.

Finally, the features of all paths between each drug pair are aggregated into a unified feature vector. This is achieved by summing the individual path feature vectors between d_1 and d_2. Consequently, the feature vector for a drug pair with N_{d_1,d_2} paths is represented by

$$x_{d_1 d_2} = \sum_{i=1}^{N_{d1,d2}} x_i = \sum_{i=1}^{N_{d1,d2}} [c_{r_1} c_{r_2}, ... c_{r_{105}}]_i = [\sum_i c^i_{r_1}, \sum_i c^i_{r_2}, ..., \sum_i c^i_{r_{105}}]$$

where $c^i_{r_j}$ refers to the feature c_{r_j} of the path x_i. The finalized feature representation, depicted in the bottom-right section of the figure below, is subsequently used for training a classifier to predict new, probable drug-drug interactions (DDIs) [1].

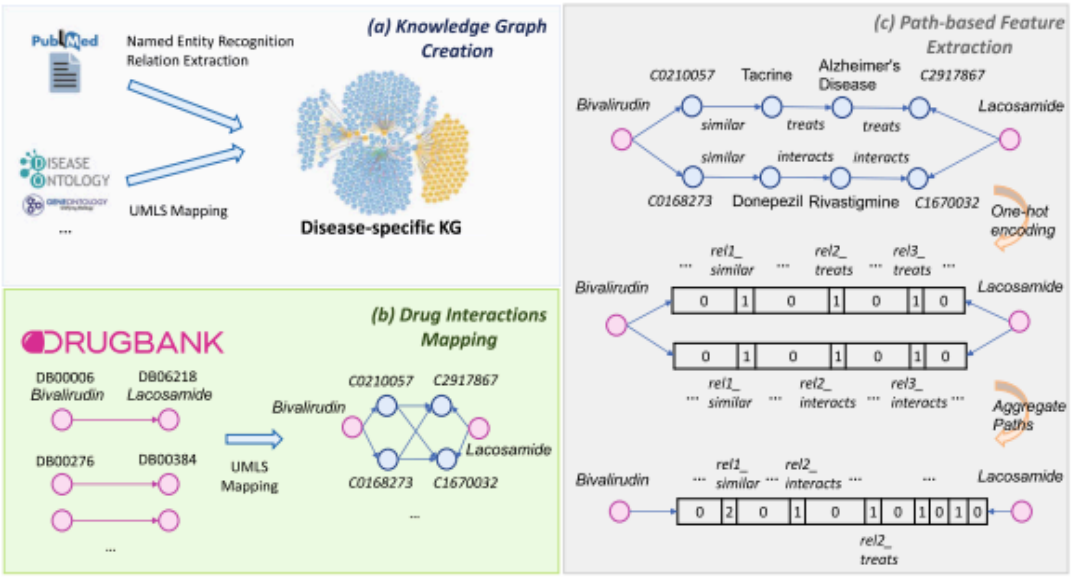

Figure 2. Process of forming the feature representation [1]

3.2.3 Drug to Drug Interactions

For predicting drug-drug interactions (DDIs), DrugBank 5.0.33 was utilized, considering only drugs related to lung cancer (LC). According to their descriptions, 68 LC-related drugs were identified. From these drugs and their interactions with other drugs in DrugBank, 4,494 interacting pairs for LC were found and we have identified DDI classes that will be presented in the Experiments chapter, Ground Truth section. Negative pairs were generated using a corrupted sampling method. For each positive pair (d1, d2 $\in$ D+), a corresponding negative pair (d1, d' $\notin$ D+) was created, ensuring that highly interacting drugs were equally represented in both positive and negative classes. Additionally, to reflect real-world scenarios where true DDIs are rare, negative pairs from the KGs were oversampled [1].

3.3 Graph Embeddings

Graph analysis facilitates the understanding of data, thus providing valuable insights for a variety of practical applications, such as node classification, node recommendation, link prediction, and many more. However, a significant challenge that exists in many graph analysis methods is related to their computational complexity and significant memory requirements.

Graph embedding is emerging as a powerful and resource-efficient solution to address these computational challenges. This technique systematically transforms graph data into a lower dimensional space, optimizing the preservation of vital structural information and graph properties. The result is a representation that not only mitigates the computational and storage inefficiencies associated with graph analysis but also leverages the preservation of important graph features, thereby enabling more efficient and effective analysis and modeling [5].

In these experiments, four key knowledge graph embedding methods were utilized : **TransE**, **HoLE**, **DistMult and RESCAL**, each offering distinct mechanisms for modeling relational data.

- **TransE (Translating Embeddings)** [18] is a translation-based method that represents relationships as translations between head and tail entities in a vector space. It assumes that, for a triple (h, r, t), the relationship r acts as a vector translation, such that $h + r \simeq t$. TransE is computationally efficient and works well for one-to-one relationships but struggles with more complex, multi-relational data.
- **HoLE (Holographic Embeddings)** [19] is a tensor-based method that uses circular correlation to model pairwise interactions between entities and relations. By leveraging a compositional structure, HoLE compresses high-dimensional tensor products into lower-dimensional representations, allowing it to efficiently capture multi-relational patterns while being scalable. Its main strength lies in modeling complex interactions but it sometimes faces difficulties with highly imbalanced or sparse datasets.

- **DistMult** [20] models relationships as bilinear interactions, where the relationship type r is represented by a diagonal matrix. This makes it computationally simpler and effective at capturing symmetric relationships, but its simplicity leads to limitations when dealing with asymmetric or complex relations (e.g., hierarchical or directed edges).
- **RESCAL** [28] is another tensor-based embedding method that represents entities and relations as matrices, allowing it to model multi-relational data more flexibly. In RESCAL, each relation is associated with a matrix that transforms the representation of the head entity into the tail entity. This method captures both symmetric and asymmetric relations effectively but can be computationally intensive, especially with a large number of entities and relations.

Differences:

- **TransE** uses additive translation to model relations, while HoLE and DistMult rely on multiplicative interactions (tensor and bilinear models, respectively).
- **HoLE** is more expressive than TransE and DistMult, capturing more complex relationship patterns.
- **RESCAL** provides greater flexibility for multi-relational data compared to TransE, while maintaining the ability to model asymmetric relationships.
- **DistMult** is more capable of modeling multi-relational data than TransE.

Usefulness:

- **TransE** is fast and effective for simple, large-scale knowledge graphs, making it ideal for datasets with mostly 1-to-1 relationships.
- **HoLE** excels in balancing expressiveness and efficiency, suitable for more complex interaction modeling without overwhelming computational resources.
- **DistMult** is especially useful for symmetric relations but can be less expressive for diverse relationship types, though its simplicity leads to fast computation.
- **RESCAL** excels in capturing the intricacies of multi-relational data and can effectively model both symmetric and asymmetric relationships, although it may require more computational resources.

By incorporating these methods, your experiments likely benefited from a mix of efficiency and the ability to model varying relational complexities, enhancing both scalability and accuracy.

3.4 Random Forest Classifier

In the field of drug-drug interactions (DDIs) in knowledge graphs, decoding the complex web of relationships requires the expert use of powerful machine learning algorithms. Among these algorithms, the Random Forest Classifier (RFC) is a machine learning algorithm, known for its flexibility, predictive ability and reduced risk of overfitting [29]. This section begins a

comprehensive exploration of the RFC, clarifying its key features, functions, and deep significance in the context of DDI knowledge graphs.

The Random Forest Classifier, an ensemble learning technique, derives its name from merging multiple decision trees to form a powerful and flexible classifier. Its establishment comes from the need to enhance predictive accuracy, mitigate overfitting, and provide a reliable framework for classification tasks.

RFC works on the principle of ensemble learning, where multiple decision trees are constructed, each capturing different aspects of the data. These decision trees cast their vote on case classification and the final prediction is determined by majority vote, making RFC robust to individual tree idiosyncrasies. RFC is suitable for both classification and regression tasks. In the context of DDI knowledge graphs, its classification capabilities enable the prediction of drug interactions, facilitating decision support in drug development and clinical practice. Aggregating multiple decision trees within RFC mitigates the risk of overfitting by ensuring that the model generalizes well to unseen data. This is an advantage when dealing with noisy or complex DDI datasets. In addition, it provides a mechanism for evaluating attribute importance by specifying the contribution of variables (attributes) to the classification process. In DDI research, this feature analysis can help identify key factors for drug interactions.

The utility of RFC in the context of DDI knowledge graphs is underscored by several key advantages:

- Predictive Modeling: RFC excels at predictive modeling, making it suitable for DDI prediction tasks. Its ability to handle high-dimensional data, consider feature importance, and mitigate overfitting enhances the accuracy of drug interaction predictions.

- Handling unbalanced data: Imbalanced data sets, a common challenge in DDI research, are effectively handled by RFC. Its ensemble approach ensures robust performance even when some DDI classes have limited occurrences.

- Interpretability: The RFC offers a level of interpretability through feature importance analysis, allowing researchers to identify the critical factors influencing drug interactions within knowledge graphs.

- Robustness: RFC's robustness to noise and outliers in DDI data ensures that it can handle real-world complexities, providing reliable information on drug interactions.

In conclusion, the Random Forest Classifier is a powerful machine learning algorithm capable of making meaningful predictions of drug-drug interactions within knowledge graphs. Its ensemble learning model, flexibility, robustness to overfitting, and feature importance analysis make it a powerful tool for DDI prediction and analysis.

Chapter 4: Experiments

This chapter presents the experiments conducted and details the implementation of each stage of this thesis. It covers the creation of the ground truth, the key technologies employed, and the specific modeling techniques used to achieve optimal results.

4.1 Ground Truth

In the field of biomedical and pharmaceutical research, the acquisition of comprehensive information about drugs is of very high importance. DrugBank, an important tool in this field, is an invaluable resource, offering a repository of drug data and related knowledge. This section begins a comprehensive exploration of DrugBank, its fundamental characteristics, underlying goals, and central role in advancing research in the field of drug-drug interactions and beyond.

DrugBank is a comprehensive pharmacology repository designed to centralize and organize drug-related knowledge. It integrates detailed information on drug properties, interactions, and therapeutic uses, serving as a vital resource for researchers, clinicians, and the pharmaceutical industry.

DrugBank's core function include but are not limited to:

1. **Drug Data Aggregation**: Consolidates data on drug structures, pharmacological properties, mechanisms, and clinical uses.
2. **Drug Interaction Profiles**: Catalogs interactions between drugs, as well as with food and diseases, aiding in adverse effect evaluation and therapy optimization.
3. **Drug Classification and Annotation**: Systematically categorizes drugs by therapeutic and chemical classes, enhancing data retrieval and supporting medical research.

Within the extensive collection of data and functionality offered by DrugBank, the aspect that assumes importance for the pursuit of Drug-Drug Interaction (DDI) research is the Drug Interaction Profiling functionality. This particular feature has emerged as an invaluable resource that has not only facilitated the efforts of researchers, but has also played a key role in shaping the landscape of scientific literature, giving rise to a significant volume of scientific publications.

The drug interaction profile feature in DrugBank serves as a foundation for DDI research, providing researchers with a structured and comprehensive platform to explore, analyze and annotate drug interactions. Its central role in defining drug interactions and decoding their implications in the field of pharmacology underscores its profound impact on the advancement of scientific knowledge. As the following sections of this thesis make clear, the drug interaction profiling function within DrugBank assumes a central role within DDI research, contributing substantially to the body of research in this critical area of pharmacological research.

In summary, DrugBank is an amazing tool for drug research. The curation of drug data such as the description of drug interactions and their therapeutic applications are important for the discovery of adverse effects. Some drug-drug interactions will be presented later in this section.

Given the large number of potential adverse reactions and the comparatively limited number of drug pairs, it becomes apparent that numerous adverse reactions lack a sufficiently meaningful representation to facilitate effective training of a machine learning model. Furthermore, recognizing the similarity between some adverse effects, merging specific initial classes was deemed imperative to increase the volume in each class, thereby enhancing the effectiveness of model training. Consequently, the final classification scheme included six distinct categories.

In the absence of access to a domain expert, whose input would have been invaluable for ensuring the quality of the dataset and the selection of classes, the dataset was based on the limited expertise available at the time. From the DrugBank dataset, which includes over 30 distinct interaction classes, five classes were selected for analysis. Despite the reduction in the number of classes, these five account for approximately 92% of the dataset's total representation, underscoring their relevance. A closer examination of the chosen classes reveals their significance and the reasoning behind their selection.

The table below provides an overview of the names of the positive categories along with their respective representation within the original dataset.

The risk or severity of bleeding can be increased	Class 1	4.78%
The risk or severity of adverse effects can be increased	Class 2	15.92%
The therapeutic efficacy of X can be decreased	Class 3	5.93%
The metabolism of X can be increased/decreased	Class 4	40.66%
Increases serum level/concentration	Class 5	25.3%

Table 1. Dataset class percentages

While Class 0 was not in the table it represents cases with no observed negative effects, serving as the negative or control category. Notably, this class comprises the majority of the dataset, exceeding the combined instances of all other classes. The remaining five classes are defined as positive categories, each corresponding to a specific type of adverse drug reaction: Class 1, Class 2, Class 3, Class 4, and Class 5.

Classes 1 through 4 are native, but Class 5, labeled "Increases serum level/concentration," represents a merged class derived from two closely related interactions: "May decrease the

excretion rate of Y, which could result in a higher serum level," and "The serum concentration of X can be increased when combined with Y.".

4.2 Experiment Presentation

This section serves as a comprehensive exploration of the experimental methodology undertaken in this research, including the two main experiments that are central to this research. These experiments include predictions derived from DrugBank data via a CSV file and predictions derived from knowledge graph integrations. This section navigates the complex process, providing insight into the code segments, decision-making, and multifaceted actions that led to the desired research results.

At the core of this section are the two main experiments, each of which represents an important aspect of this research. The former focuses on predictions derived from DrugBank data stored in a CSV file, while the latter focuses on predictive models generated by knowledge graph integrations. These primary experiments serve as the conduits through which this research is conducted on drug-drug interactions.

Within the general framework of main experiments, a structured approach that involves breaking each main experiment into smaller, discrete experiments is used . These individual experiments are meticulously presented, offering an in-depth understanding of the decision-making processes that guided this research. This aims to present the progression that supports this experimentation.

In conclusion, this chapter forms the core of the experimental approach used in this research. It systematically addresses the main objectives by organizing the experiments into clear and manageable parts. By explaining the code and the reasoning behind key decisions, it provides an overview of the research process. The upcoming sections will focus on the details of each experiment, emphasizing how structured experimentation contributes to better understanding drug-drug interactions in knowledge graphs.

4.2.1 File Preprocessing

In this research, data processing takes on a central role as it serves to distill the data set to include only the relevant information from the original data. Additionally, generating the final CSV file derived from the graph database and DrugBank, the online database, required careful structuring to facilitate the correct relationships that were included in both the databases. Additionally, further preprocessing was performed to achieve a fine-grained understanding of the available classes without adding noise, given the multiclass nature of the problem.

Creating the final CSV file required a series of complex mappings. Primarily, the project involved correlating each drug's name with its corresponding DrugBank identification (ID), a daunting task due to the imperative to locate each drug in the comprehensive DrugBank database. The database, in this context, was obtained in the form of an XML file, thus requiring the application of a sophisticated script capable of traversing the entire database to distinguish the relevant drugs.

After successful mapping of drug names, an exhaustive extraction of all possible combinations of adverse effects associated with these drugs was performed. In addition, a thorough cleaning process followed, as the original CSV file, containing the path analysis of the drug pairs, was found to be outdated when compared to the DrugBank database. The latter, being more recent, led to inconsistencies, necessitating the deletion of many drug pairs to align with current and accurate data.

4.2.2 Path Analysis Experiments

This chapter describes the main category of experiments conducted in the context of this thesis. As shown earlier, the project is basically oriented to the comparative analysis of two distinct methodologies for multi-class predictions. The focus of this chapter is to explain the classification approach, where our models were trained using the Path Analysis CSV file specified. Several experiments were meticulously performed, aiming not only to improve the overall results but also to systematically distinguish the incremental improvements observed in model performance across the various methodologies used.

4.2.2.1 Baseline

Establishing the base model serves the purpose of measuring the performance of the selected algorithm within the default data set. This implies refraining from every detail of the algorithm or dealing with data preprocessing. A tenfold cross-validation methodology was used to train and test the model on the raw data. This approach ensures comprehensive training without exposing the model to the test data before the evaluation phase. To perform this baseline experiment the algorithm chosen was Random Forest Classifier (RFC), known for its robust results, and for its fine tuning capabilities.

Evaluation of results was performed based on precision, recall, and F1 score metrics derived from the classification report. In addition, the support metric was considered to verify the exposure of the model to the correct number of examples. In addition, an examination of false positives and false negatives was performed through the confusion matrix. While the confusion matrix provides a visual method for model evaluation, its utility is somewhat limited in this

context due to the multiclass taxonomic nature of the problem, which requires significant time for comprehensive visual evaluation.

4.2.2.2 Grid Search with balanced weights

After establishing the basic model and a comprehensive analysis of the data set, insights into the limitations of the model become apparent. Identifying these limitations is crucial to determining the appropriate preprocessing steps that can improve overall performance. In the realm of machine learning, it is often argued that the quality of the input data directly affects the quality of the output, embodied in the saying: "garbage in, garbage out". For the particular data set under consideration, given its previous cleaning to create the archive, basic pre-processing has been completed. Attention then turns to fitting the data for optimal fit to the model. After scrutinizing the results of the basic model, one obvious observation emerged: classes 1 and 3 were underrepresented, while class 0 was disproportionately dominant. To correct this imbalance, two main interventions were implemented.
First, subsampling of the majority class (class 0) was performed. This involved deliberately hiding some examples from the class with the highest instances to ensure that, while remaining dominant, the model did not focus solely on learning that particular class.

Second, the minority classes (classes 1 and 3) were sampled. Given the underrepresentation of these classes, the synthetic minority oversampling technique (SMOTE) was used [8], leveraging the K Nearest Neighbors (KNN) algorithm. In this implementation the five, if the upsampling is 200% its two, nearest points are chosen and one sample is generated in the direction of each. Synthetic samples are generated using this method: First, calculate the difference between the feature vector and its nearest neighbor. Then, scale this difference by a random value between 0 and 1 and add the result to the original feature vector. This process selects a random point along the line segment connecting the two feature vectors, effectively broadening the decision boundary for the minority class. This technique, known as SMOTE, promotes a more generalized decision region for the minority class [8]. This method strategically expands the decision-making area of the minority class, promoting a more generalized model. Importantly, SMOTE ensures that the data being synthesized unambiguously belongs to the target class.

Both the downsampling and upsampling methods required user-defined specifications about the desired number of examples for each class. To optimize the results, a variety of possible combinations for both upsampling and downsampling were systematically explored.

In addition, to improve model performance, hyperparameter optimization is imperative. The chosen Random Forest Classifier (RFC) exhibits several parameters that, when properly tuned, can significantly improve the results. The main parameter that requires optimization is 'n_estimators', which dictates the number of decision trees that make up the forest. In addition,

the 'max_depth' parameter specifies the depth of each tree, with the option of expanding nodes until the leaves are clean or contain fewer than the specified 'min_samples_split' samples if 'None' is selected. Finally, 'min_samples_leaf' represents the minimum number of samples required at a leaf node.

Given the number of possible combinations between these parameters, specific values for each parameter were initialized to explore a diverse set of combinations without exhausting computational resources. To systematically evaluate the performance of the model under various parameter configurations, the Grid Search function was used. This function takes as input the model and the allowed values for each parameter, training the model with all possible combinations. To ensure the robustness of the model, a five-fold cross-validation strategy was implemented under Grid Search, facilitating a comprehensive evaluation of the model's effectiveness with different parameter configurations.

With a full understanding of the previous considerations, the experimental architecture is delineated as follows:

1. Decimal External Cross Validation: The dataset undergoes a tenfold external cross-validation, splitting it into training and test sets. Specifically, the model is trained exclusively on the processed training set, characterized by the optimized number of examples per class, while the evaluation results on the original class distribution.

2. Nested For Loop for Upsampling and Downsampling: Within the outer cross-validation, a for loop iterates through all possible combinations of upsampling and downsampling values. This makes it easier to train models on different data sets, exploring a range of possible configurations.

3. Nested cross-validation with grid search: Within the for loop, another cross-validation loop is implemented, using the grid search function. This function systematically trains and tests the model with various configurations of hyperparameters, optimizing the choice of parameters.

4. Extract the best model: After completing outer cross-validation and nested loops, the best performing model is extracted. This model represents the optimal combination of hyperparameters and data configurations.

5. Metric Evaluation: The experiment is completed by extracting relevant measurements to evaluate the performance of the model. Precision, recall, and F1 score, among other metrics, provide a comprehensive assessment of model performance.

This meticulous experimental design ensures that the model is trained and evaluated over a range of hyperparameter configurations, upsampling and downsampling values. Consequently, it makes it easier to identify the most effective combination to achieve optimal performance.

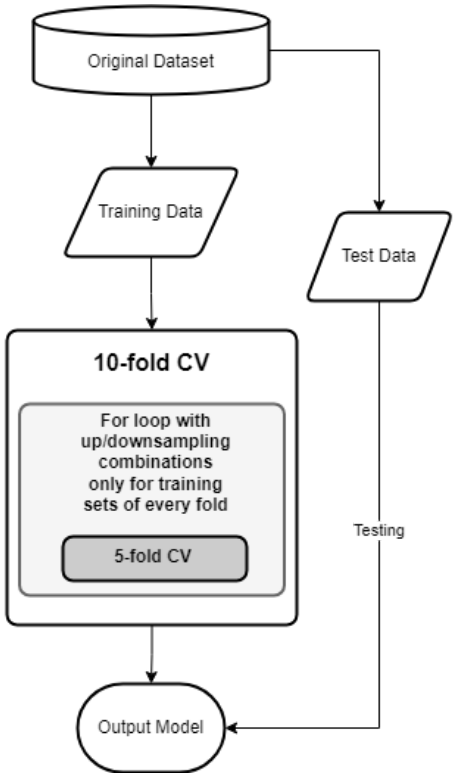

Figure 3. Architecture of Path Analysis Experiments

In addition to the previously discussed methods to improve model performance, another critical technique used in this experiment involves incorporating class weights into the Random Forest Classifier (RFC). The "class_weight" parameter, an integral part of the RFC, plays a key role in adjusting the importance of each class during model training [9].

By default, if the 'class_weight' parameter is left undefined, all classes are assigned a weight of 1, indicating equal importance. However, given the significant class imbalance inherent in this work, assigning weighted categories becomes imperative. This strategic adjustment allows the model to place greater emphasis on underrepresented classes, resulting in facilitating improved learning and improved performance.

By assigning appropriate weights to each class, the model is encouraged to allocate more resources to learning from minority classes, thereby mitigating the negative effects of class imbalance and promoting fairer classification outcomes. This approach complements the previously described methods, offering a comprehensive strategy for addressing the challenges posed by class imbalance and optimizing model performance.

In this experiment, the class_weight parameter was set to 'balanced' mode. When set to "balanced" mode, the RFC adjusts class weights automatically based on the input data. Specifically, the weights are inversely proportional to the class frequencies, calculated as *n_samples / (n_classes * np.bincount(y))*.

Using the 'balanced' function, RFC dynamically adjusts the weights of each class to mitigate the effects of class imbalance within the dataset. Consequently, the model pays more attention to the minority classes while softening the influence of the majority class. This mechanism ensures a fairer treatment of all classes during model training, enhancing the model's ability to discern patterns and make accurate predictions across the range of class distributions.

4.2.2.3 Grid Search with custom weights

In pursuit of further performance optimization, an extension of the experiment described in section 4.2.2.2 was performed. Depending on the fine-tuning of the model parameters to identify the most effective configurations, attention was turned to improving the ‘class_weights’ parameter to better match the characteristics of each class. In contrast to the balanced class weights approach, adjusted weights were assigned to individual classes based on empirical knowledge gained from previous runs and experiential observations.

Essentially, if a particular category performed suboptimal, or if its performance significantly affected the performance of other categories, targeted adjustments were made to improve the category weights. This iterative process entailed meticulous refinement with the goal of striking a balance between maximizing the performance of each category and maintaining the overall efficiency of the model.

By adjusting class weights based on empirical observations and past performance, the experiment sought to optimize the model's performance and achieve superior classification results tailored to the unique features and challenges each class poses in the dataset.

4.3.3 Graph Embedding Experiments

This chapter describes the second methodology used for multi-class predictions in this thesis. This experiment’s data is derived directly from the Neo4j database, transformed into graph embeddings used to train and test the model.

4.3.3.2 Baseline Experiment

In alignment with the Path Analysis experiment, it was essential to establish baseline models to generate initial metrics and provide a benchmark for subsequent comparisons. Given the nature

of the study, four distinct baseline models were developed, corresponding to each of the graph embeddings utilized throughout the experimental series (TransE, HoLE, DistMult and RESCAL). This was crucial for evaluating the performance of each embedding in relation to experiments that incorporated hyperparameter optimization and data manipulation, as well as for comparing the embeddings against one another.

Specifically, the baseline experiments employed the knowledge graph to extract semantic triples, which were subsequently subjected to graph embedding. The resulting data was then input into a basic Random Forest Classifier, which was trained using 10-fold cross-validation to ensure robustness and reliability in the performance evaluation.

4.3.3.3 Experiments with Balanced and Custom Weights

To enhance model performance, several established methods were implemented. Initially, the hyperparameters for the Random Forest Classifier were directly obtained from the preliminary experiments. Subsequently, data manipulation techniques were applied through a trial-and-error approach. Given the overrepresentation of the negative class, undersampling was performed, while classes 1 and 3, which were underrepresented, were addressed using the Synthetic Minority Over-sampling Technique (SMOTE) as in the first experiment.

Furthermore, class weights were adjusted to improve model accuracy. The results of the experiments will be presented, focusing on three primary experiments using the Graph Embedding method: the baseline, the Balanced Weights Experiment, and the Custom Weights Experiment. While the first experiment is the Baseline which has no data manipulation, the other two experiments share the same hyperparameter tuning and data manipulation techniques, with variations in class weights. The second experiment employs balanced class weights, calculated based on class representation in the dataset, giving greater weight to underrepresented classes. The third experiment utilizes custom weights determined through extensive trial and error, with only the optimal custom weight set reported in this thesis.

Chapter 5: Results

This chapter presents the findings derived from the methodologies described in the previous sections. The results include a detailed analysis of the performance metrics of the proposed models in addressing the problem of drug to drug interaction prediction. Each subsection provides a thorough examination of the outcomes highlighting the strengths and limitations observed. This discussion aims to underscore the practical implications, and potential for future applications of the proposed approaches.

5.1 Result Discussion

5.1.1 Path Analysis Results

The Path Analysis experiments consist of a series of complex tasks utilizing data directly extracted from the Knowledge Graph. To fully leverage the triple paths created within the dataset, it was necessary to implement several robust methodologies. Establishing a baseline remains essential to assess the model's performance effectively and provide a benchmark for improvement.

Baseline	precision	recall	f1-score	support
Class 0	0.79	0.98	0.88	21102
Class 1	0.50	0.05	0.09	190
Class 2	0.66	0.26	0.37	1506
Class 3	0.55	0.12	0.19	224
Class 4	0.53	0.21	0.30	3426
Class 5	0.43	0.07	0.11	1726
accuracy			0.77	28174
macro avg	0.58	0.28	0.32	28174
weighted avg	0.73	0.77	0.72	28174

Table 2. Path Analysis Baseline Results

The baseline model demonstrates an accuracy of 77%. However, given that this task involves multiclass classification, accuracy alone does not provide a comprehensive assessment of model

performance. In this project series, the primary evaluation metrics are the macro-average F1-score and the individual F1-scores for each class. Despite the overall accuracy of 77%, classes 1 through 5 have a maximum F1-score of only 37%, with a total support of 7,072 instances, while the negative class achieves an F1-score of 88% with a support of 21,102 instances. This disparity indicates that the high accuracy is largely due to the strong performance on the negative class, whereas our main focus lies on the positive classes, where F1-scores range from a maximum of 37% to a minimum as low as 9%.

The following experiment presents the results achieved after conducting hyperparameter tuning using grid search. Also, it involved determining the optimal level of downsampling for the majority class and the ideal upsampling for the two minority classes. Additionally, balanced class weights were applied to the random forest classifier, assigning higher weights to classes with fewer instances to address class imbalance effectively.

Balanced weights	precision	recall	f1-score	support
Class 0	0.87	0.82	0.85	21102
Class 1	0.22	0.48	0.30	190
Class 2	0.46	0.48	0.47	1506
Class 3	0.27	0.40	0.33	224
Class 4	0.43	0.56	0.48	3426
Class 5	0.33	0.31	0.32	1726
accuracy			0.76	28174
macro avg	0.43	0.51	0.46	28174
weighted avg	0.74	0.76	0.73	28174

Table 3. Path Analysis Balanced Weights Results

The results indicate that the application of balanced weights, optimal downsampling of the majority class, and upsampling of the minority classes has yielded mixed improvements when compared to the baseline.

For class 0, the precision remains high at 0.87, though slightly lower than the baseline precision of 0.79. However, the recall decreases from 0.98 in the baseline to 0.82, resulting in a slight

decrease in the F1-score from 0.88 to 0.85. Despite this decrease, the model's high performance in this majority class remains largely stable, suggesting effective retention of accuracy.

For minority classes (1, 2, 3, and 4), notable improvements are observed in recall. Class 1 recall, for example, increases substantially from 0.05 in the baseline to 0.48 in the balanced model, with a corresponding increase in F1-score from 0.09 to 0.30. Similarly, classes 2, 3, and 4 show increases in recall (from 0.26 to 0.48 for class 2 and from 0.12 to 0.40 for class 3), indicating improved identification of these underrepresented classes. However, this improvement in recall often comes at the cost of precision, which is lower for most minority classes compared to the baseline. As a result, F1-scores in these classes remain moderate, signaling a trade-off between precision and recall.

Class 5 also shows minor improvements in F1-score (from 0.11 to 0.32) and recall (from 0.07 to 0.31), although the gains are not as significant as in other minority classes. This suggests that while the model has benefited from balanced class weights, further tuning may be necessary to optimize performance in this class.

The overall accuracy of the model is slightly reduced from 0.77 in the baseline to 0.76 with balanced weights. However, the macro-average F1-score improves from 0.32 to 0.46, reflecting enhanced performance across minority classes.

To further improve the model's performance, a series of iterative experiments were conducted, refining the class weights through trial and error. This process led to a set of final weights assigned to each class. The results presented below utilize the same experimental setup as the previous trial, with the only modification being the adjusted class weights applied to the random forest classifier.

Custom weights	precision	recall	f1-score	support
Class 0	0.86	0.88	0.87	21102
Class 1	0.35	0.35	0.35	190
Class 2	0.62	0.26	0.36	1506
Class 3	0.45	0.47	0.46	224
Class 4	0.44	0.53	0.48	3426
Class 5	0.33	0.23	0.27	1726
accuracy			0.80	28174
macro avg	0.51	0.46	0.47	28174
weighted avg	0.77	0.80	0.78	28174

Table 4. Path Analysis Custom Weights Results

The introduction of custom class weights shows meaningful improvements in certain metrics when compared to both the baseline and the balanced weights configurations.

For class 0, the F1-score slightly increases to 0.87 from 0.85 in the balanced weights setup, with precision and recall values now more closely aligned at 0.86 and 0.88, respectively. This result suggests that the custom weights have allowed the model to maintain high performance in the majority class while achieving better balance between precision and recall.

In the minority classes, custom weights produce mixed but generally positive outcomes. Class 1, for example, shows a further improvement in F1-score from 0.30 (balanced weights) to 0.35, with both precision and recall reaching 0.35. While this improvement is modest, it highlights a more balanced identification of class 1 instances compared to the previous setups.

Class 2, however, experiences a decline in recall (from 0.48 with balanced weights to 0.26 with custom weights) and a minor reduction in F1-score (from 0.47 to 0.36). This indicates that custom weights may have impacted the model's ability to detect instances in this class effectively.

Class 3 shows a marked improvement, with both precision and recall at 0.45 and 0.47, respectively, resulting in an F1-score of 0.46. Compared to balanced weights, which yielded an F1-score of 0.33, this enhancement indicates a more robust classification of this class.

Similarly, class 4 demonstrates stability with a slightly higher F1-score of 0.48, up from 0.48 in the balanced setup, while class 5 experiences a slight decline in performance, with its F1-score dropping from 0.32 to 0.27.

The overall accuracy of the model improves from 0.76 with balanced weights to 0.80 with custom weights, reflecting a more accurate overall classification. The macro-average F1-score remains stable at approximately 0.47, while the weighted average F1-score shows a slight increase from 0.73 to 0.78. This outcome suggests that the custom weights contribute positively to the model's overall performance by fine-tuning the balance across both majority and minority classes.

5.1.2 Graph Embedding Results

In this series of experiments, four distinct graph embeddings are employed to generate the input features for our model. The experimental setup mirrors that of the Path Analysis experiments, encompassing baseline results, hyperparameters tuned with balanced weights, and the application of custom weights. The key difference in this set of experiments lies in the utilization of four unique embeddings while maintaining consistent hyperparameters from the Path Analysis configuration. This approach allows us to systematically compare performance outcomes across various embedding techniques within a controlled parameter setting.

The first set of results provides baseline scores for our models, establishing a benchmark against which subsequent configurations will be evaluated.

TransE	precision	recall	f1-score	support
Class 0	0.75	1	0.86	7363
Class 1	0	0	0	55
Class 2	0	0	0	603
Class 3	0	0	0	72
Class 4	0	0	0	1169
Class 5	0	0	0	545
accuracy			0.75	9807
macro avg	0.13	0.17	0.14	9807
weighted avg	0.56	0.75	0.64	9807

Table 5. TransE Baseline Results

HoLE	precision	recall	f1-score	support
Class 0	0.75	1	0.86	7363
Class 1	0	0	0	55
Class 2	0	0	0	603
Class 3	0	0	0	72
Class 4	0	0	0	1169
Class 5	0	0	0	545
accuracy			0.75	9807
macro avg	0.13	0.17	0.14	9807
weighted avg	0.56	0.75	0.64	9807

Table 6. HoLE Baseline Results

DistMult	precision	recall	f1-score	support
Class 0	0.75	1	0.86	7363
Class 1	0	0	0	55
Class 2	0	0	0	603
Class 3	0	0	0	72
Class 4	0	0	0	1169
Class 5	0	0	0	545
accuracy			0.75	9807
macro avg	0.13	0.17	0.14	9807
weighted avg	0.56	0.75	0.64	9807

Table 7. DistMult Baseline Results

RESCAL	precision	recall	f1-score	support
Class 0	0.75	1	0.86	7363
Class 1	0	0	0	55
Class 2	0	0	0	603
Class 3	0	0	0	72
Class 4	0	0	0	1169
Class 5	0	0	0	545
accuracy			0.75	9807
macro avg	0.13	0.17	0.14	9807
weighted avg	0.56	0.75	0.64	9807

Table 8. RESCAL Baseline Results

The baseline results for all four embeddings (TransE, HoLE, DistMult, and RESCAL) show identical performance patterns, with strong classification for the majority class (class 0) and no successful classifications for minority classes (classes 1–5).

For class 0, all models achieve a perfect recall of 1.0 and a precision of 0.75, resulting in an F1-score of 0.86. This indicates that the models are effectively identifying class 0 instances, which likely represent the majority of the data, as seen in the support count of 7,363 out of a total of 9,807 instances. However, for the minority classes, precision, recall, and F1-scores are all 0, showing that the models fail to identify any instances in these classes. This issue suggests that the baseline models are highly biased toward the majority class, likely due to severe class imbalance.

The overall accuracy of 0.75 across all embeddings primarily reflects the high classification rate of class 0, rather than a balanced performance across classes. The macro-average F1-score, a more balanced metric, is notably low at 0.14, underscoring the models' poor performance on minority classes. Similarly, the weighted average F1-score is 0.64, which remains moderately high due to the dominance of class 0 but does not reflect effective classification across the dataset.

These results highlight the need for more sophisticated strategies to handle class imbalance. Moving forward, tuning parameters and adjusting class weights will be essential to improve recall and precision for minority classes, thereby achieving a more balanced model performance across all classes.

The next set of experiments examines the performance of each embedding after applying several techniques to address class imbalance and optimize the model. These include downsampling the majority class, using SMOTE to upsample minority classes 1 and 3, and applying the optimal hyperparameters identified in the path analysis experiments for the random forest classifier. Additionally, balanced class weights were implemented to further enhance model performance across classes.

TransE	precision	recall	f1-score	support
Class 0	0.8	0.87	0.84	7363
Class 1	0	0	0	55
Class 2	0.67	0	0.01	603
Class 3	0.27	0.04	0.07	72
Class 4	0.31	0.34	0.32	1169
Class 5	0.26	0.25	0.25	545
accuracy			0.71	9807
macro avg	0.38	0.25	0.25	9807
weighted avg	0.7	0.71	0.68	9807

Table 9. TransE Balanced Weights Results

HoLE	precision	recall	f1-score	support
Class 0	0.8	0.87	0.84	7363
Class 1	0	0	0	55
Class 2	0.22	0.03	0.06	603
Class 3	0.27	0.04	0.07	72
Class 4	0.33	0.27	0.3	1169
Class 5	0.22	0.28	0.24	545
accuracy			0.71	9807
macro avg	0.31	0.25	0.25	9807
weighted avg	0.67	0.71	0.68	9807

Table 10. HoLE Balanced Weights Results

DistMult	precision	recall	f1-score	support
Class 0	0.77	0.94	0.85	7363
Class 1	0	0	0	55
Class 2	0	0	0	603
Class 3	0	0	0	72
Class 4	0.35	0.03	0.05	1169
Class 5	0.22	0.29	0.25	545
accuracy			0.73	9807
macro avg	0.22	0.21	0.19	9807
weighted avg	0.63	0.73	0.66	9807

Table 11. DistMult Balanced Weights Results

RESCAL	precision	recall	f1-score	support
Class 0	0.82	0.62	0.71	7363
Class 1	0	0	0	55
Class 2	0.24	0.01	0.01	603
Class 3	0.21	0.19	0.2	72
Class 4	0.45	0.04	0.07	1169
Class 5	0.09	0.71	0.17	545
accuracy			0.19	9807
macro avg	0.3	0.26	0.19	9807
weighted avg	0.69	0.51	0.55	9807

Table 12. RESCAL Balanced Weights Results

For the **TransE**, the model shows strong performance for class 0 with a precision of 0.8, recall of 0.87, and an F1-score of 0.84. However, the minority classes continue to face challenges. Class 2 achieves minimal improvement, with an F1-score of only 0.01, and classes 1 and 3 have F1-scores of 0, indicating difficulty in capturing these classes even with upsampling and balanced weights. Class 4 shows moderate performance with an F1-score of 0.32, while class 5 achieves an F1-score of 0.25. Overall, the weighted average F1-score is 0.68, but the macro-average F1-score remains low at 0.25, highlighting an ongoing imbalance.

HoLE performs similarly to TransE for the majority class (F1-score of 0.84) but shows slightly lower precision and recall for minority classes. While class 4 has a modest improvement (F1-score of 0.3), other classes, especially 1 and 2, show little to no gain. The macro-average F1-score remains at 0.25, and the weighted average F1-score is 0.68, similar to TransE, suggesting limited improvement in class balance.

The **DistMult** displays a higher recall for class 0 (0.94), resulting in an F1-score of 0.85. However, performance for minority classes is largely unchanged or worsened. Classes 1, 2, and 3 have no recorded F1-scores, and class 4 shows an F1-score of only 0.05. Class 5's F1-score improves slightly to 0.25, but the overall macro-average F1-score drops to 0.19, with a weighted F1-score of 0.66, suggesting a reliance on class 0 for performance metrics.

The **RESCAL** shows unique but uneven results. Class 0's recall drops significantly to 0.62, which impacts its F1-score, reducing it to 0.71. Interestingly, class 5 exhibits the highest recall among minority classes at 0.71, though its precision remains low (0.09), resulting in an F1-score of 0.17. Classes 1 and 2 have no measurable performance, and class 4 achieves only an F1-score of 0.07. The overall accuracy of 0.19 and weighted F1-score of 0.55 reflect RESCAL's poor overall classification performance, largely due to lower recall for class 0.

In the next set of experiments, custom weights are applied to further address the model's limitations in classifying minority classes effectively. By adjusting the class weights beyond balanced weights, these experiments aim to enhance the model's sensitivity to underrepresented classes, striving for improved recall and precision across all categories while reducing the bias toward the majority class. This approach seeks to build on the previous modifications, focusing on achieving a more balanced performance across the entire dataset.

TransE	precision	recall	f1-score	support
Class 0	0.86	0.69	0.7	7363
Class 1	0.4	0.04	0.07	55
Class 2	0.24	0.04	0.07	603
Class 3	0.09	0.01	0.02	72
Class 4	0.28	0.73	0.4	1169
Class 5	0.24	0.27	0.25	545
accuracy			0.63	9807
macro avg	0.35	0.3	0.26	9807
weighted avg	0.71	0.63	0.64	9807

Table 13. TransE Custom Weights Results

HoLE	precision	recall	f1-score	support
Class 0	0.88	0.66	0.75	7363
Class 1	0.18	0.07	0.1	55
Class 2	0.18	0.03	0.05	603
Class 3	0.12	0.24	0.16	72
Class 4	0.29	0.83	0.43	1169
Class 5	0.23	0.3	0.26	545
accuracy			0.62	9807
macro avg	0.31	0.35	0.29	9807
weighted avg	0.72	0.62	0.64	9807

Table 14. HoLE Custom Weights Results

DistMult	precision	recall	f1-score	support
Class 0	0.84	0.73	0.78	7363
Class 1	0.1	0.02	0.03	55
Class 2	0.38	0.03	0.05	603
Class 3	0.18	0.24	0.2	72
Class 4	0.28	0.63	0.39	1169
Class 5	0.22	0.25	0.23	545
accuracy			0.64	9807
macro avg	0.33	0.32	0.28	9807
weighted avg	0.7	0.64	0.65	9807

Table 15. DistMult Custom Weights Results

RESCAL	precision	recall	f1-score	support
Class 0	0.84	0.62	0.72	7363
Class 1	0	0	0	55
Class 2	0.34	0.15	0.21	603
Class 3	0.2	0.22	0.21	72
Class 4	0.41	0.08	0.14	1169
Class 5	0.11	0.77	0.19	545
accuracy			0.53	9807
macro avg	0.32	0.31	0.24	9807
weighted avg	0.71	0.53	0.58	9807

Table 16. RESCAL Custom Weights Results

For **TransE**, the model demonstrates improved precision for class 0 (0.86) but a decline in recall (0.69), resulting in an F1-score of 0.77. This indicates that while the model maintains a good precision for the majority class, it is less effective at capturing all relevant instances. The performance for the minority classes remains limited, with class 1 achieving a precision of 0.4 and a recall of 0.04 (F1-score of 0.07), and classes 2 and 3 displaying similar shortcomings. However, class 4 shows a notable increase in recall to 0.73, leading to a higher F1-score of 0.4. Overall, the accuracy of 0.63 reflects the model's reliance on class 0, and the macro-average F1-score of 0.26 indicates a persistent imbalance in classification performance across all classes.

HoLE shows slightly higher precision for class 0 (0.88) compared to TransE, but its recall drops to 0.66, resulting in an F1-score of 0.75. Similar to TransE, the performance for the minority classes is subpar, with classes 1 and 2 receiving low scores (F1-scores of 0.1 and 0.05, respectively). Class 4 exhibits improved recall at 0.83, which contributes to a better F1-score of 0.43. The accuracy remains at 0.62, with a macro-average F1-score of 0.29, suggesting that the model continues to struggle with effective minority class identification.

In the case of **DistMult**, the precision for class 0 is at 0.84, with a recall of 0.73, leading to an F1-score of 0.78. However, the model again shows limited success for minority classes, with class 1 achieving low metrics (F1-score of 0.03). Class 4's recall is improved at 0.63, leading to an F1-score of 0.39, but overall, the macro-average F1-score drops to 0.28, indicating that custom weights have not sufficiently addressed the imbalance.

RESCAL presents an overall accuracy of 0.53, the lowest among the embeddings. While class 0 retains a decent precision of 0.84, its recall is considerably lower at 0.62, resulting in an F1-score of 0.72. The performance on minority classes is notably poor, particularly for class 1, which remains unclassified. Class 5, however, achieves a high recall of 0.77, but it's precision is low at 0.11, resulting in a weak F1-score of 0.19. The macro-average F1-score is particularly low at 0.24, reflecting significant challenges in classifying minority instances effectively.

5.2 Result Comparison

In previous chapters, the results of the conducted experiments were presented. In this chapter, we will undertake an in-depth comparison of the results from each individual experiment. Subsequently, a comparative analysis and discussion will be conducted to evaluate the outcomes of both experiments, aiming to determine which method demonstrates greater effectiveness and reliability.

5.2.1 Path Analysis Result Comparison

In Figure 4, the improvements in macro-average score across the experiments are displayed. The macro-average is the preferred metric for this project due to the multiclass classification nature

of the task, as it offers a comprehensive view of each class's performance within the overall results.

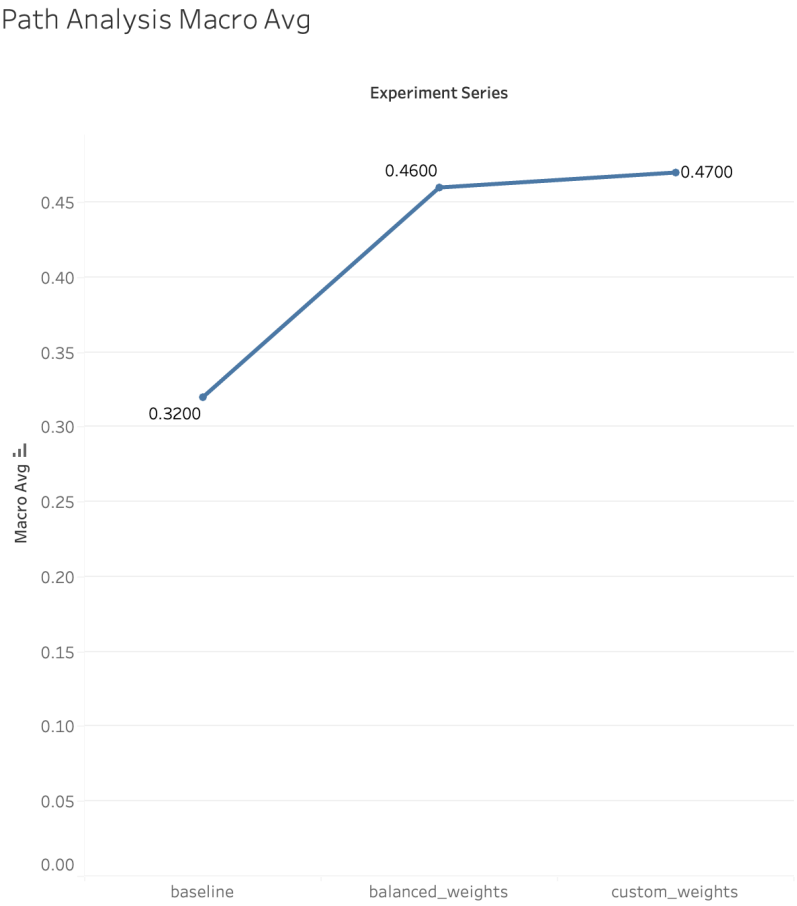

Figure 4. Path Analysis F1 Macro Average Improvement

The baseline model achieves a F1 macro-average score of 32%, establishing a benchmark to surpass. The subsequent experiment, which incorporates data processing techniques (including downsampling of the majority class and upsampling of minority classes), fine-tuning (hyperparameter tuning via grid search), and balanced class weights, demonstrates a marked improvement over the baseline. With a score of 46%, this experiment achieves a 44% improvement over the baseline.

Building on this approach, the next experiment retains the data processing and tuning steps but applies custom class weights determined through a trial-and-error process. This weight configuration produces the best-performing model, with a 47% macro-average score. A more detailed discussion of the results is provided in the following figures.

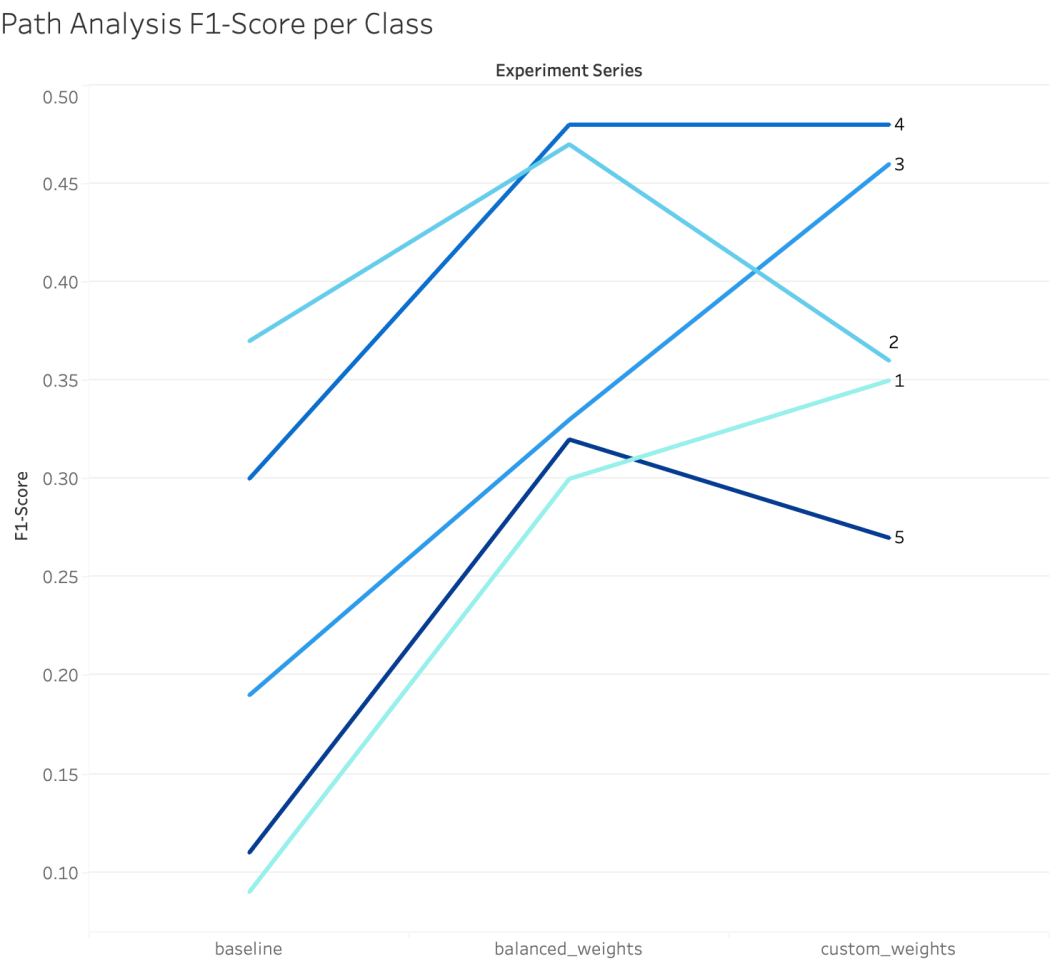

Figure 5. Path Analysis F1-Score Improvements per Class

The goal of this experiment is to predict interactions between drugs, making the positive classes, cases where drug interactions are present, our primary focus. Thus, greater attention should be given to the model's performance in identifying positive classes (actual drug interactions), rather than its classification of negative classes (instances where no interaction is present).

With this focus, only the positive classes and their improvements are highlighted, as each experiment aims to boost these scores. In Figure 5, it is evident that all results improve from the baseline; however, to determine whether balanced weights or custom weights yield the superior model, a closer examination of the results is required.

Using custom weights notably improves the F1-scores of minority classes 1 and 3. Specifically, class 1 increases from approximately 0.30 to 0.35, marking a 0.05 improvement, while class 3 rises from 0.33 to 0.45, an increase of 0.12—resulting in a combined improvement of 0.17. Conversely, classes 2 and 5 experience a decline, with class 2 dropping from 0.47 to 0.36 and class 5 from 0.32 to 0.27, totaling a decrease of 0.16, which nearly offsets the gains in the

minority classes. Class 4 remains largely unchanged, showing only a slight and negligible increase.

Considering these findings, although both the balanced and custom weights approaches yield comparable overall results, custom weights are identified as the better-performing model. This is due to the more substantial increase in total F1-score, largely driven by improvements in the minority classes and the minor gain in class 4.
To substantiate the previous claims, Figure 6 presents a detailed analysis of each class's precision and recall, which are the metrics used to calculate the F1-score.

It is observed that the baseline model exhibits higher precision compared to the balanced weights experiment, but the recall is substantially increased. This suggests that, while the model successfully captures more relevant positive instances, it also introduces a higher number of false positives. The custom weights model addresses this issue by attempting to restore the baseline precision levels while minimizing the recall loss. This model achieves a balance between precision and recall that is not attained by the other two models. For these reasons, the custom weights model is selected as the overall best model for the Path Analysis experiment.

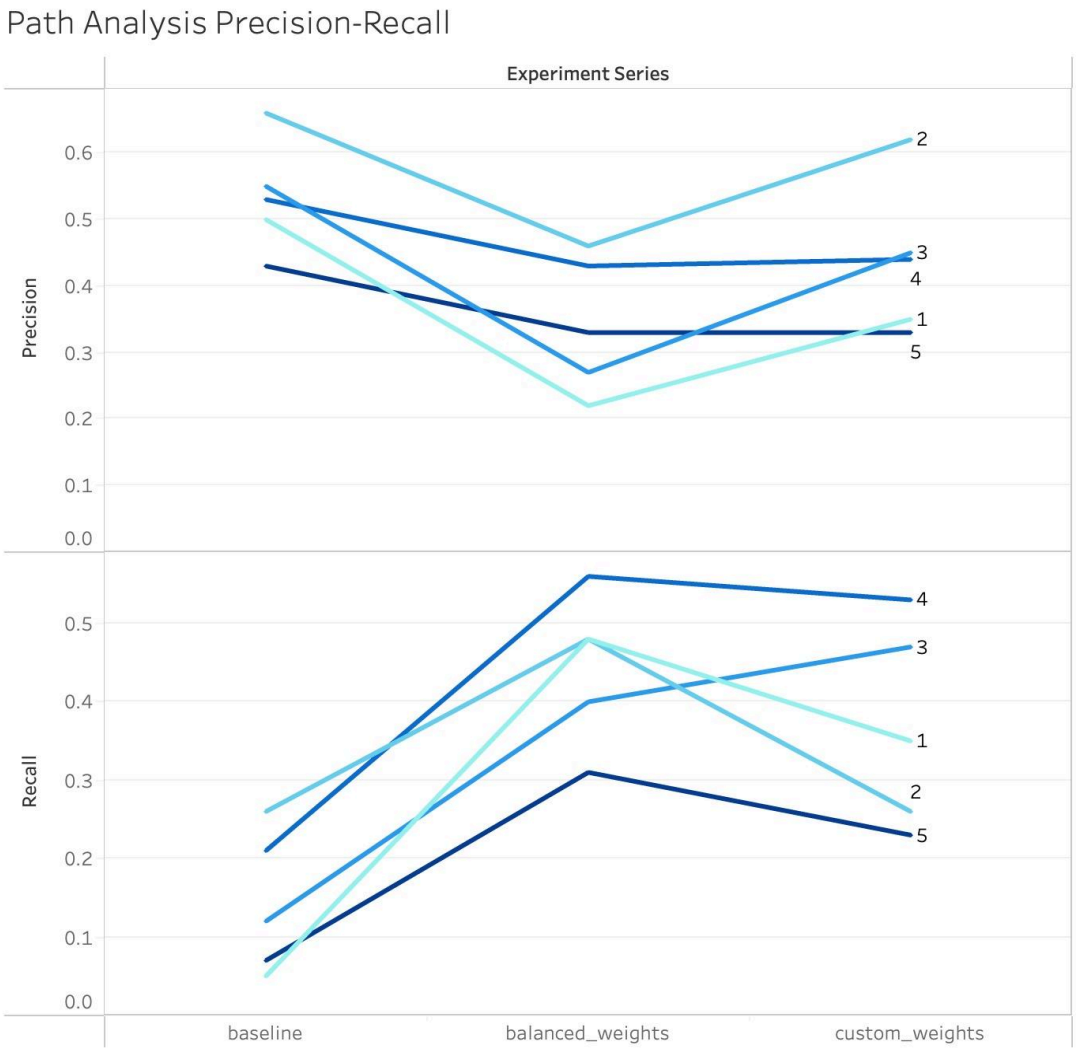

Figure 6. Path Analysis Precision-Recall Improvements per Class

5.2.2 Graph Embeddings Results Comparison

As in the previous chapter, this section will present an in-depth discussion and comparison of the results obtained through the Graph Embedding methodology. The following figures will offer the necessary context and details to facilitate a comprehensive understanding of the outcomes of this analysis. A comparative evaluation of each embedding approach will be presented, including performance metrics for each class. Finally, based on both the macro-average metric and the performance across individual classes, the most effective model will be identified.

Figure 7 presents the macro average metric for each model across all embeddings. This means that each embedding model has undergone three stages: the baseline model, the balanced weights model, and the custom weights model. The graph illustrates the performance differences for each model at each stage.

While the baseline model exhibits consistent performance across all embeddings, the performance gap in the subsequent experiments is quite noticeable. With the balanced weights model, the performance of the HoLE and TransE embeddings nearly doubles, whereas RESCAL and DistMult show a more modest increase of 35%, rising from 0.14 to 0.19. When custom weights, derived through a trial-and-error process, are applied, all models show an improvement over their balanced weights counterparts. Specifically, TransE's macro average increases by just 0.1, RESCAL rises from 0.19 to 0.24, still lagging behind TransE with balanced weights, and DistMult experiences a significant improvement from 0.19 to 0.28, surpassing TransE in performance. Finally, HoLE shows a performance boost from 0.25 to 0.29, yielding the overall best results.

While DistMult and HoLE clearly outperform the other models in terms of macro average, a more detailed analysis of each embedding performance will be provided in the Figure 8.

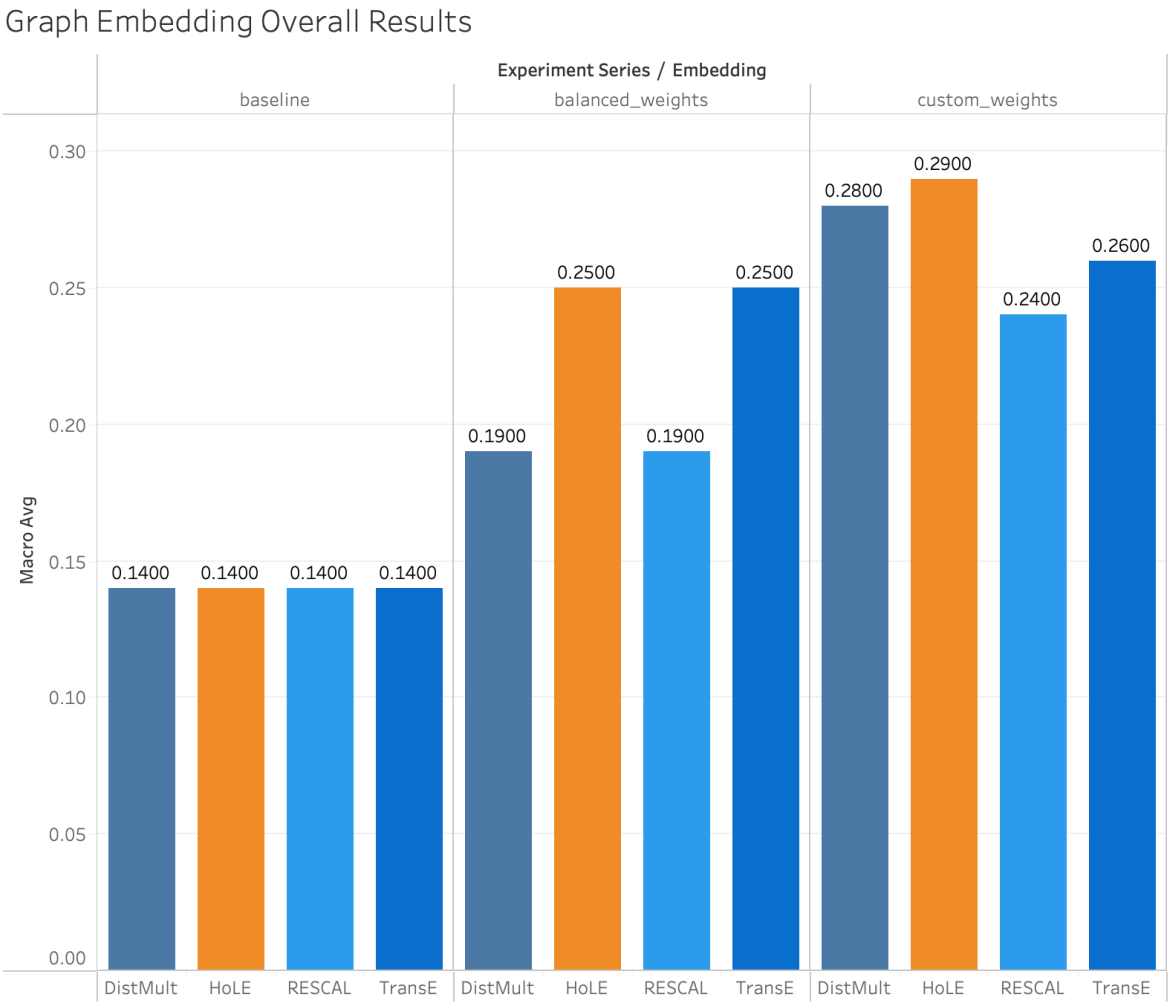

Figure 7. Graph Embedding F1 Macro Average per Embedding

As previously mentioned, Figure 8 provides a more detailed breakdown of the metrics for all custom weights models. Specifically, it displays the F1-score for each individual positive class, allowing for a comparison between embeddings.

Upon initial inspection of the plot, it is evident that DistMult and HoLE outperform the other embeddings, with RESCAL being the lowest performing model. Despite RESCAL outperforming DistMult in both classes 2 and 3, its complete inability to identify class 1, coupled with poor performance in classes 4 and 5, places it as the worst performing model.

TransE, while excelling in class 2 compared to both DistMult and HoLE, and showing strong performance in classes 4 and 5, was unable to learn classes 1 and 3 effectively, resulting in lower overall performance than the other two models.

Finally, while DistMult outperforms both TransE and RESCAL, it still falls short when compared to HoLE, which excels in three out of the five classes. HoLE and DistMult perform nearly identically in class 2, with DistMult significantly outperforming HoLE in class 3.

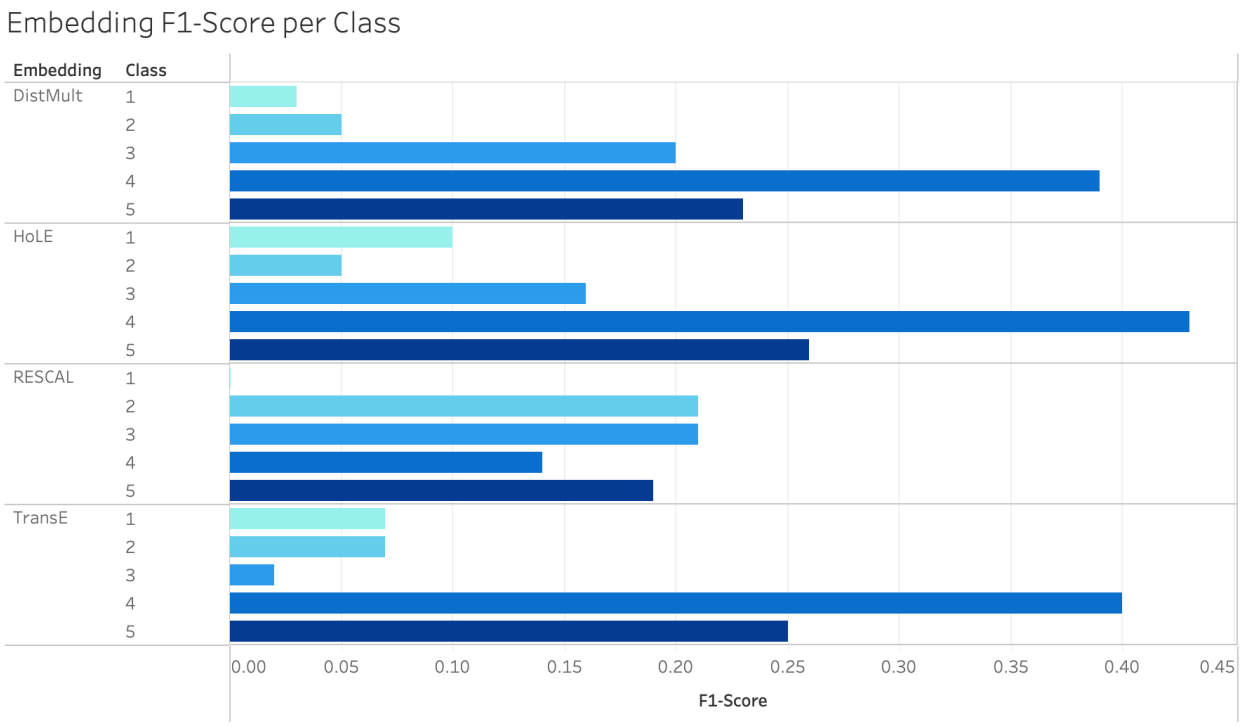

Figure 8. Graph Embedding F1-Score per Embedding per Class

For further comparison, the following figure presents the precision and recall metrics for the two best-performing models overall: the custom-weighted HoLE and DistMult. While the performance of both models in terms of learning each class is fairly similar, there are sufficient differences that allow HoLE (H) to outperform DistMult (DM) in the overall performance comparison.

For the negative class (class 0), HoLE achieves better precision, while DistMult performs better in recall. Regarding the positive classes, class 1 shows HoLE with nearly double the performance in both precision and recall, as also demonstrated in Figure 8. In class 2, both models have the same recall, but DistMult outperforms HoLE in precision by nearly 90%. Class 3 shows equal recall for both models, but DistMult has a significant edge in precision. Finally, for classes 4 and 5, while DistMult performs adequately, HoLE outperforms it in every category.

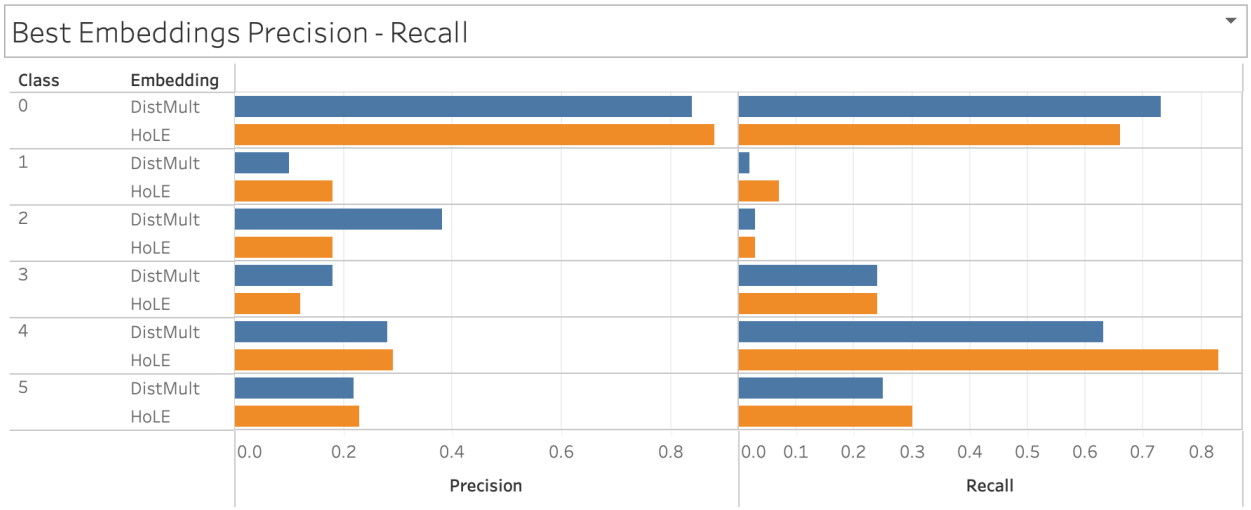

Figure 9. Top scoring embeddings Precision-Recall per Class

5.2.3 Methodology Comparison

While the previous comparisons focused on similar methodologies, the primary aim of this thesis is to identify the optimal model for each method and then compare them to determine which approach is most effective for addressing the problem of Drug-Drug interactions. This chapter focuses precisely on that objective, evaluating the best models from each methodology and comparing their performance.

For reference, the Path Analysis model that outperformed all others was the Random Forest model, which utilized upsampling of minority classes, downsampling of the majority (negative) class, and custom weights. In contrast, the Graph Embedding method identified the optimal model as a Random Forest, with input data preprocessed similarly to the Path Analysis method

(including up/downsampling), enhanced by the HoLE embedding, and supplemented with custom weights for each class.

To fully understand the capabilities and differences between each model, a comparison is necessary. Figure 10 presents the F1-score performance of each model across all classes.

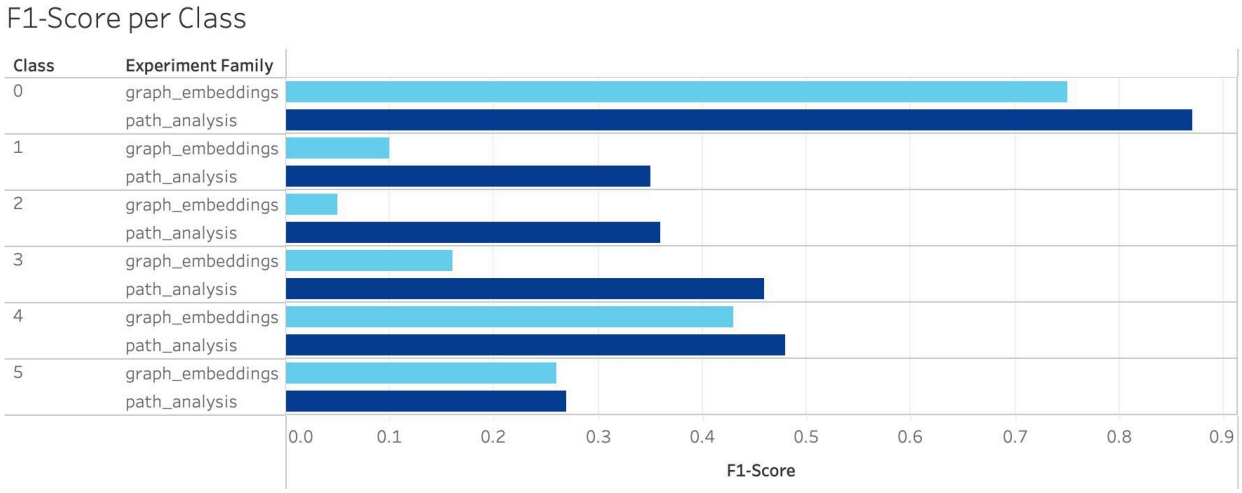

Figure 10. Path Analysis vs Graph Embeddings

Class 0, being the negative class, is of lesser interest; however, it is still important to evaluate all classes to determine which methodology performs better overall. As shown in Figure 10, the Graph Embedding (GE) model performs well in class 0, achieving an F1-score close to 0.75, but it is outperformed by the Path Analysis (PA) model, which achieves an F1-score close to 0.88. For classes 1, 2, and 3, the difference in performance is substantial, with the PA model surpassing the GE model by nearly 0.25 in all three classes. In classes 4 and 5, the performance difference between the models is smaller, but the PA model still outperforms the GE model by approximately 0.05 in class 4 and by 0.01 in class 5.

Thus, it is clear that the Path Analysis methodology is the superior model and overall approach.

Chapter 6: Conclusions and Future Extensions

The final chapter of this thesis will focus on the conclusions and potential future extensions. In the conclusions section, a comprehensive discussion of the overall experiments and the methodologies employed to conduct them will be provided. The future extensions section will outline additional methods and procedures that could enhance the research but were not included in this thesis, along with plans for the continued development of this project.

6.1 Conclusions

Throughout the thesis, we systematically explored the characteristics, strengths, and limitations of each approach in relation to the unique requirements of DDI tasks.The experiments conducted in this thesis underscored the importance of both methodologies in capturing distinct aspects of DDI data. Graph Embedding techniques leveraged structured relationships within the data, creating vector representations that mapped drug interactions within a high-dimensional space. This approach proved effective in establishing semantic relationships between drugs, thereby facilitating an understanding of underlying interaction patterns. However, it also faced limitations in adequately distinguishing between classes and in performance stability across different classes, suggesting a need for additional model optimization or more sophisticated embedding techniques.

Conversely, the Path Analysis approach, which focused on sequential patterns within the data, demonstrated a higher degree of accuracy and robustness across multiple classes. Path Analysis allowed for a structured breakdown of interaction pathways, resulting in a more nuanced understanding of interaction characteristics. This method ultimately surpassed Graph Embedding in terms of macro-average metrics, showing particular strength in handling both positive and negative classes with higher precision and recall, and ultimately producing the most reliable classification results.

In comparing the two methodologies, it became evident that Path Analysis offers greater practical advantages in terms of predictive power and overall performance in multi-class categorization. This finding highlights Path Analysis as a more effective methodology for DDI prediction in this study, though the contributions of Graph Embeddings should not be overlooked.

In conclusion, this thesis provides a meaningful contribution to the field of DDI prediction by examining and comparing two widely applicable methods, ultimately finding Path Analysis to be the more suitable approach within the constraints and scope of this study. Future work may focus on further refining Graph Embedding techniques or integrating both methods to harness the

structural insights of embeddings with the detailed pattern analysis of Path Analysis, potentially leading to a hybrid approach that combines the strengths of each for improved DDI prediction accuracy.

6.2 Future Extensions

While this thesis demonstrates a series of successful experiments, there remains potential to further enhance the performance of the models. Several factors could expand the capabilities of these methodologies. For instance, experimenting with a wider range of models, including Neural Networks, could help identify higher-performing alternatives. Additionally, the model's ability to accurately represent each class could be improved by consulting a domain expert to assist in annotating each class. As this is an MSc thesis, access to such expertise was limited, and class definitions were established with limited domain knowledge, occasionally resulting in class groupings that may not be entirely accurate or easily distinguishable.

The immediate next step for this project involves continued experimentation with fine-tuning the Graph Embedding methodology and experimenting using various GGN or rule-based approaches. The goal is to enhance its performance to approach the capabilities of the Path Analysis model or to identify the performance limits of the embedding approach.

References

[1] K. Bougiatiotis, F. Aisopos, A. Nentidis, A. Krithara, and G. Paliouras, “Drug-Drug interaction prediction on a biomedical literature knowledge graph,” in *Lecture Notes in Computer Science*, Cham: Springer International Publishing, 2020, pp. 122–132. Accessed: Sep. 11, 2024. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-59137-3_12

[2] Z.-H. Ren *et al.*, “BioDKG–DDI: predicting drug–drug interactions based on drug knowledge graph fusing biochemical information,” *Briefings in Functional Genomics*, vol. 21, no. 3, pp. 216–229, Apr. 2022, doi: 10.1093/bfgp/elac004.

[3] Y.-H. Feng and S.-W. Zhang, “Prediction of Drug-Drug Interaction Using an Attention-Based Graph Neural Network on Drug Molecular Graphs,” *Molecules*, vol. 27, no. 9, p. 3004, May 2022, doi: 10.3390/molecules27093004.

[4] T. Hoa Vo, “On the road to explainable AI in drug-drug interactions prediction: A systematic review,” *Computational and Structural Biotechnology Journal*, vol. 20, pp. 2112–2123, Apr. 2022, doi: 10.1016/j.csbj.2022.04.021.

[5] HongYun Cai, Vincent W. Zheng, Kevin Chen-Chuan Chang, “A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications,” *IEEE Xplore*, Feb. 2018. https://ieeexplore.ieee.org/abstract/document/8294302

[6] “DrugBank Online,” *DrugBank*. https://go.drugbank.com/ (accessed Sep. 11, 2024).

[7] “Neo4j Graph Database & Analytics,” *Graph Database & Analytics*, May 16, 2020. https://neo4j.com/ (accessed Sep. 11, 2024).

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[9] Scikit-learn Developers, “RandomForestClassifier,” *scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed Sep. 11, 2024).

[10] K.Corrie and J.G. Hardman, “Mechanisms of drug interactions: Pharmacodynamics and pharmacokinetics” *Anaesthesia & Intensive Care Medicine*, pp. 305–308, Jul. 2014, doi: 10.1016/j.mpaic.2014.04.005.

[11] L. Sun *et al.*, “Pharmacokinetic and Pharmacodynamic Drug–Drug Interactions: Research

Methods and Applications," *Metabolites*, vol. 13, no. 8, p. 897, Jul. 2023, doi: 10.3390/metabo13080897.

[12] O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. 90001, pp. 267D – 270, Jan. 2004, doi: 10.1093/nar/gkh061.

[13] T. C. Rindflesch and M. Fiszman, "The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text," *Journal of Biomedical Informatics*, vol. 36, no. 6, pp. 462–477, Dec. 2003, doi: 10.1016/j.jbi.2003.11.003.

[14] Y.-H. Feng, S.-W. Zhang, and J.-Y. Shi, "DPDDI: a deep predictor for drug-drug interactions," *BMC Bioinformatics*, vol. 21, no. 1, Sep. 2020, doi: 10.1186/s12859-020-03724-x.

[15] B. Kotnis and V. Nastase , "Analysis of the Impact of Negative Sampling on Link Prediction in Knowledge Graphs," Aug. 2017.

[16] H. He, G. Chen, and C. Yu-Chian Chen, "3DGT-DDI: 3D graph and text based neural network for drug–drug interaction prediction," *Briefings in Bioinformatics*, vol. 23, no. 3, Apr. 2022, doi: 10.1093/bib/bbac134.

[17] Y. Zhong *et al.*, "DDI-GCN: Drug-drug interaction prediction via explainable graph convolutional networks," *Artificial Intelligence in Medicine*, vol. 144, p. 102640, Oct. 2023, doi: 10.1016/j.artmed.2023.102640.

[18] A. Bordes, Nicolas Usunier, and J. Weston, "Translating Embeddings for Modeling Multi-relational Data".

[19] M. Nickel, L. Rosasco, and T. Poggio, "Holographic Embeddings of Knowledge Graphs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016, doi: 10.1609/aaai.v30i1.10314.

[20] "DistMult — pykeen 1.10.2 documentation." Accessed: Oct. 08, 2024. [Online]. Available: https://pykeen.readthedocs.io/en/stable/api/pykeen.models.DistMult.html

[21] D. Fensel *et al.*, "Introduction: What is a knowledge graph?," in *Knowledge Graphs*, Cham: Springer International Publishing, 2020, pp. 1–10. Accessed: Oct. 22, 2024. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-37439-6_1

[22] "Hetionet," An integrative network of biomedical knowledge. Accessed: Oct. 22, 2024. [Online]. Available: https://het.io/

[23] “PharmGKB,” PharmGKB. Available: https://www.pharmgkb.org/

[24] “Bio2RDF v2.7a.” Accessed: Oct. 22, 2024. [Online]. Available: https://bio2rdf.org/

[25] G. Yu, Q. Ye, and T. Ruan, “Enhancing Error Detection on Medical Knowledge Graphs via Intrinsic Label,” *Bioengineering*, vol. 11, no. 3, p. 225, Feb. 2024, doi: 10.3390/bioengineering11030225.

[26] Y. Zhao, D. Bollegala, S. Hirose, Y. Jin, and T. Kozu, “Community knowledge graph abstraction for enhanced link prediction: A study on PubMed knowledge graph,” *Journal of Biomedical Informatics*, vol. 158, p. 104725, Oct. 2024, doi: 10.1016/j.jbi.2024.104725.

[27] I. A. Ebeid, “MedGraph: A semantic biomedical information retrieval framework using knowledge graph embedding for PubMed,” *Frontiers in Big Data*, vol. 5, Oct. 2022, doi: 10.3389/fdata.2022.965619.

[28] M. Nickel, V. Tresp , and H.-P. Kriegel, “A Three-Way Model for Collective Learning on Multi-Relational Data,” *ICML’11: Proceedings of the 28th International Conference on International Conference on Machine Learning*, Jun. 2011.

[29] IBM, “Random Forest,” IBM. Available: https://www.ibm.com/topics/random-forest

[30] “PubMed,” PubMed. Available: https://pubmed.ncbi.nlm.nih.gov/

[31] “MEDLINE Overview.” Accessed: Nov. 22, 2024. [Online]. Available: https://www.nlm.nih.gov/medline/medline_overview.html

Appendix

UMLS

The Unified Medical Language System (UMLS) is a comprehensive repository developed by the US National Library of Medicine (NLM) to integrate various biomedical vocabularies and terminologies. Its purpose is to address the challenge of terminological inconsistency in biomedical research by offering a single framework for standardizing how medical concepts are named and related across fields. UMLS incorporates over two million terms representing approximately 900,000 concepts and establishes more than 12 million relationships between these concepts, making it one of the most extensive resources for biomedical terminology [12].

UMLS consists of three main components:

- **Metathesaurus**: This is the central component of UMLS and consists of interrelated biomedical concepts derived from more than 60 vocabularies, including Medical Subject Headings (MeSH), Gene Ontology, NCBI taxonomy, and SNOMED CT. The Metathesaurus groups synonymous terms into unified concepts and organizes these concepts into a network of relationships, which may be hierarchical (e.g., "is-a" or "part-of" relationships) or associative (e.g., " caused' or 'found in').
- **Semantic Network**: Each concept in the Metathesaurus is classified into one of 135 high-level semantic types. This system enables users to categorize and retrieve concepts based on their general type, such as "disease," "gene," or "organism," facilitating both broad and specialized biomedical searches.
- **SPECIFIC dictionary and dictionary tools**: These resources provide tools for handling variations in biomedical terminology. They help create lexicons of term variations, including plural forms, derivations, and other linguistic modifications. This supports more robust information retrieval by dealing with variability in how terms are expressed.

UMLS is designed to address two major barriers to efficient retrieval and use of biomedical information: the variety of names used to express the same concept and the absence of a standard format for distributing terminologies. By incorporating a large number of controlled vocabularies from different biomedical fields, UMLS provides a common platform for standardizing these terminologies. For example, concepts such as diseases, drugs, anatomical structures, and medical procedures from different sources are grouped together, simplifying tasks such as medical data integration, electronic health record (EHR) management, and clinical decision support.

A key function of UMLS is its ability to link various biomedical resources. For example, concepts from MeSH, used in medical literature indexing, can be linked to Gene Ontology entries, used to annotate gene products. This interconnection allows researchers to

cross-reference information across multiple databases, such as linking genetic data to clinical disease manifestations.

UMLS offers significant advantages to the biomedical community:

- **Comprehensive integration**: UMLS draws from a vast array of vocabularies and domains, allowing it to serve as a central hub for standardized medical terminology. It includes data from the most important vocabularies, such as SNOMED CT for clinical terms, Gene Ontology for genomic data, and MeSH for literature indexing.
- **Terminology Standardization**: Through the unification of synonyms and hierarchical relationships, UMLS facilitates data exchange and interoperability between medical systems, making it easier for researchers and clinicians to share and analyze data across institutions and databases.
- **Extensive relationships and external links**: UMLS not only incorporates internal relationships between concepts, but also provides external cross-references to databases such as GenBank and OMIM, bridging clinical information and molecular biology. These references are invaluable in linking clinical phenotypes to underlying genetic causes.

UMLS is a vital tool in the biomedical field, enabling the standardization and integration of diverse terminologies in research and clinical practice. Its comprehensive knowledge base supports tasks such as data retrieval, ontology integration, and natural language processing, making it an indispensable resource for advancing biomedical research, health informatics, and clinical decision making.

SemRep

SemRep is a comprehensive semantics repository designed to improve the quality of schema and ontology mapping by leveraging a wide range of background knowledge sources. It was developed to address the limitations of existing resources such as WordNet, which, despite its popularity, lacks the coverage and up-to-date concepts required for effective semantic matching. SemRep integrates manually curated resources, such as WordNet and UMLS, and the relationships automatically extracted from Wikipedia, providing a richer and more diverse repository of concepts and their semantic relationships [13].

SemRep combines different types of semantic relations, such as equivalence (e.g., car = car), hierarchical relations (e.g., car is-a vehicle), and part-whole relations (e.g., roof part-house). . The primary goal of the repository is to help determine the semantic mappings between two schemas or ontologies. Specifically, it aims to resolve issues arising from synonymous or homonymous concepts, which are common in real-world applications. For example, SemRep can

distinguish that "mouse" in a computer context is not the same as animal "mouse," a task that many basic matching algorithms struggle with.

The repository has been created using data from four main sources: WordNet, Wikipedia, UMLS and OpenThesaurus. WordNet and UMLS offer manually curated knowledge, providing high-quality but sometimes outdated and domain-specific information. Wikipedia, with its vast and constantly updated content, greatly expands the scope of SemRep by covering modern concepts and more general knowledge. OpenThesaurus, although focused on the German language, contributes additional synonyms and relationships.

In the context of schema and ontology mapping, background knowledge such as that found in SemRep plays a critical role. Mapping between schemas usually involves finding correspondences between concepts in different ontologies. This task becomes more complicated when the concepts in question do not have obvious lexical similarities. For example, "car" and "cars" are synonyms, but without a repository like SemRep, a matching algorithm might miss this connection. SemRep not only identifies these relationships but also specifies the specific nature of the relationship—whether the two concepts are synonymous, hierarchical, or part of a larger structure.

SemRep's ability to resolve indirect relationships is a significant improvement over traditional repositories such as WordNet. For example, it can connect "car" to "cars" through a direct equivalence relation, and further connect "car" to "vehicle" through an is-a (alias) relation. These features make SemRep particularly useful for tasks that require a fine-grained understanding of the relationships between concepts, such as ontology fusion, schema integration, and text mining.

While WordNet is widely used in many semantic mapping tools, its limited scope, particularly in modern terminology and in specific domains, makes it inadequate for more complex or large-scale tasks. Wikipedia, on the other hand, has extensive coverage but lacks the rigorous curation of hand-developed resources. By combining these sources, SemRep achieves both the depth of curated resources and the breadth of automatically extracted knowledge, making it more powerful for task matching.

SemRep's architecture is designed to efficiently handle a large number of concepts. Its internal structure is based on a graph-based model, where concepts are nodes and relationships are edges. This allows for fast relationship lookups and the ability to traverse indirect relationships, which is critical for determining the most relevant semantic correspondence between two concepts. For example, SemRep can quickly recognize that a "kitchen chair" is a type of chair by analyzing compound words and evaluating step-by-step relationships.

SemRep represents a significant advance in the field of semantic mapping by integrating multiple knowledge sources and focusing on both the quality and coverage of semantic relations. Its ability to resolve direct and indirect relationships between concepts makes it an invaluable

tool for improving the accuracy and depth of schema and ontology mapping, while its scalability ensures that it can continue to evolve as new knowledge sources become available.

PyKeen

In the dynamic landscape of drug-drug interaction (DDI) research within knowledge graphs, the pursuit of effective and efficient knowledge graph integrations is crucial. PyKEEN, a Python library, emerges as a formidable tool tailored for this purpose. This section begins an extensive exploration of PyKEEN, elucidating its fundamental features, functions, and deep significance in the context of DDI knowledge graph construction and prediction.

PyKEEN, short for Python Knowledge Graph EmbeddiNg, is an open source Python library designed expressly for knowledge graph embedding tasks. Its genesis is rooted in the need to facilitate the encoding of entities and relationships within knowledge graphs in low-dimensional vector representations. This encoding, often referred to as embedding, enables the exploration of complex relationships and predictive modeling within knowledge graphs.

PyKEEN specializes in developing knowledge graph embeddings, a process that transforms entities and relationships within knowledge graphs into continuous vector spaces. These embeddings capture the semantics and structural information of the underlying knowledge graph. It also offers a comprehensive set of knowledge graph embedding models, ranging from traditional approaches like TransE and TransH to cutting-edge techniques like ComplEx and RotatE. Researchers can select and experiment with models tailored to their specific DDI research goals.

Moreover PyKEEN is designed to handle knowledge graphs of different sizes and complexities. Its scalability and optimized implementation ensure that even large-scale knowledge graphs can be efficiently integrated and used for DDI prediction. Furthermore, the library provides robust mechanisms for evaluating the quality and effectiveness of knowledge graph implementations. Researchers can use metrics such as average rank and average reciprocal rank to evaluate the performance of their DDI prediction models.

Neo4j

In the evolving landscape of biomedical research, where exploring complex relationships and interactions is paramount, Neo4j is an indispensable tool, harnessing the power of graph database technology [7]. This section begins a comprehensive exploration of Neo4j, clarifying its fundamental features, underlying goals, and central role in advancing drug-drug interaction (DDI) research within the dynamic domain of knowledge graphs.

Neo4j is a graph database management system designed to address the complex nature of relationships that permeate complex datasets. Born from the need to efficiently model and search interconnected data structures, the genesis of Neo4j is based on graph theory - a branch of mathematics that underpins the representation of entities and their relationships through nodes and edges. Neo4j's fundamental goal is to harness the power of graphs to structure, store, and retrieve data in a way that seamlessly aligns with the inherent complexity of real-world systems and interactions.

In summary, Neo4j proves to be a powerful tool for studying and understanding drug-drug interactions within the intricate framework of knowledge graphs. Its graph-based structure, efficient querying, scalability, and flexibility make it an essential resource in DDI research. As detailed in this thesis, Neo4j plays a crucial role in analyzing and interpreting drug interactions within knowledge graphs, contributing valuable insights to the field of biomedical research by uncovering the complexities of drug interactions and relationships.