

12 Aug

HUTECH UNIVERSITY
FACULTY of INFORMATION TECHNOLOGY



Khong
Khong
Mr
HUTECH

Đại học Công nghệ Tp.HCM

Enhancing Drug-drug Interaction Prediction via DGNN-DDI & EmerGNN

Final Project in Social Network Analysis Course

Khong
Module Code **CMP1048**

Instructor **Nhat-Tung Le**

Student **Hoang-Khang Nguyen**

Student ID **2186400244**

Ho Chi Minh City, November 21, 2024

HUTECH UNIVERSITY
FACULTY of INFORMATION TECHNOLOGY

—o0o—

Final Project in Social Network Analysis Course

**Enhancing Drug-drug Interaction Prediction via
DGNN-DDI leveraging EmerGNN**



Kharg

Instructor: **Nhat-Tung Le**

Student: **Hoang-Khang Nguyen**

Student ID: **2186400244**

Ho Chi Minh City, November 21, 2024

Contents

List of Figures	2
List of Tables	3
List of Acronyms & Abbreviations	4
1 OVERVIEW	5
1.1 Problems	5
Dù sao đi nữa, a vẫn iu em	
2 DATABASE & DATA PREPROCESSING	7
2.1 DrugBank	7
2.1.1 Structure of DrugBank XML	8
3 PROPOSED METHODS	9
3.1 Theoretical Foundation of Social Network Analysis	9
3.1.1 Graph Density	9
3.1.2 Degree Centrality	9
3.1.3 Closeness Centrality	10
3.1.4 Betweenness Centrality	10
3.1.5 Clustering Coefficient	11
BIBLIOGRAPHY	17

List of Figures

List of Tables

1	List of Abbreviations and their Definitions	4
2.1	Detailed Description and Structure of Datasets	7

Acronyms & Abbreviations

Table 1: List of Abbreviations and their Definitions

Abbreviation	Definition
DDI	DRUG-DRUG INTERACTION
MLP	MULTI-LAYER PERCEPTRON
GAT	GRAPH ATTENTION NETWORK
SAGPooling	SELF-ATTENTION POOLING
GCN	GRAPH CONVOLUTIONAL NETWORK
DGNN-DDI	DUAL GRAPH NEURAL NETWORK FOR DRUG-DRUG INTERACTIONS PREDICTION
SA-DMDNN	DIRECTED MESSAGE PASSING NEURAL NETWORK WITH SUBSTRUCTURE ATTENTION MECHANISM
IDOLpro	INVERSE DESIGN OF OPTIMAL LIGANDS FOR PROTEIN POCKETS
SBDD	STRUCTURE-BASED DRUG DESIGN
DDPM	DENOISING DIFFUSION PROBABILISTIC MODEL

1.1 Problems

RECENT advancements in artificial intelligence (AI), particularly in deep learning and graph learning models, have demonstrated their effectiveness in biomedical applications, especially for predicting drug-drug interactions (DDIs)¹. Traditional methods for predicting DDIs through clinical trials and experiments are costly and time-consuming. The application of advanced AI and deep learning techniques faces several challenges, including data resource availability and encoding, as well as the design of computational methods [Lin+23].

DRUG-DRUG interactions (DDIs) are a critical concern in the field of pharmacology and medicine, as they can lead to adverse drug reactions, reduced therapeutic efficacy, and even life-threatening conditions. With the increasing number of available drugs and the complexity of their interactions, it becomes essential to develop reliable computational methods for predicting potential DDIs. Traditional approaches, such as rule-based systems and statistical models, often fall short due to their inability to capture the complex and nonlinear relationships between drugs.

Graph Neural Networks (GNNs) have emerged as a powerful tool for modeling relational data and have shown great promise in various applications, including social networks, molecular chemistry, and recommender systems. In this study, we propose a novel Dual Graph Neural

¹DDIs involve changes in the effects of one drug caused by the presence of another drug in the human body, which is crucial for drug discovery and clinical research.

DDIs prediction is one of the applications of molecular representation. DDIs is referred to as a situation where the pleasant or adverse effects caused by the co-administration of two drugs, which may cause adverse drug events and side effects that damage the body. In order to avoid such events, it's urgent to develop computational approaches to detect DDIs [Mei23].

Network (DGNN) framework for DDI prediction, which leverages the strengths of GNNs to effectively capture the intricate relationships between drugs.

The main contributions of DGNN-DDI are as follows:

- We introduce a dual graph neural network architecture that simultaneously models drug-level and substructure-level interactions.
- We employ substructure attention mechanisms to enhance the representation learning of drug substructures.
- We perform a comprehensive evaluation of our model on real-world DDI datasets, demonstrating its superior performance compared to existing state-of-the-art methods.

Table 2.1: Detailed Description and Structure of Datasets

Dataset	Description and Structure
ANTI-COVID-19 DRUG	A dataset containing information on drugs and compounds tested for activity against COVID-19. It includes drug names, chemical structures, activity data, clinical trial statuses, and target proteins.
DRUGBANK	A comprehensive database containing detailed drug data and drug-target interactions. Structure includes drug information (name, chemical structure, mechanism of action), pharmacological data, interactions, and clinical trial information.
TWOSIDES	A dataset focused on drug-drug interactions and their side effects. It includes pairs of drugs, interaction effects, frequency of side effects, and severity ratings.
CROSSDOCKED	A dataset providing information on docking poses of small molecules in various protein pockets. Structure includes protein targets, small molecule ligands, docking scores, and pose coordinates.
MOAD	The Mother of All Databases (MOAD), which contains high-quality data on protein-ligand complexes. Structure includes protein-ligand complexes, binding affinity data, experimental conditions, and crystallographic information.

2.1 DrugBank

DRUGBANK is a comprehensive resource that combines detailed drug data with comprehensive drug target information. It is widely used in bioinformatics research,

drug discovery, and pharmaceutical applications. This document provides a detailed guide on how to preprocess and utilize the DrugBank database.

The DrugBank database can be downloaded in various formats, including XML, CSV, and SQL. Each format contains rich information about drugs, including their chemical properties, targets, interactions, and pathways. For this guide, we will focus on the XML format, as it is the most comprehensive.

2.1.1 Structure of DrugBank XML

The DrugBank XML file contains multiple entries, each corresponding to a single drug. Each drug entry includes:

3.1 Theoretical Foundation of Social Network Analysis

3.1.1 Graph Density

Graph density is a measure of how "complete" a graph is, defined as the ratio of the actual number of edges to the maximum possible number of edges.

- For undirected graphs:

$$D = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}| - 1)}$$

where:

- $|\mathcal{E}|$: The number of edges in the graph.
- $|\mathcal{V}|$: The number of vertices (nodes) in the graph.

- For directed graphs:

$$D = \frac{|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}| - 1)}$$

Here, there is no factor of 2 since edges are directed and each pair of nodes can have up to two edges (one in each direction).

3.1.2 Degree Centrality

Degree centrality quantifies how well-connected a node is in the network. It is calculated as the degree of the node normalized by the maximum possible degree.

- For undirected graphs:

$$C_D(v) = \frac{\deg(v)}{|\mathcal{V}| - 1}$$


where $\deg(v)$ is the degree of node v , representing the number of edges connected to v .

- **For directed graphs:**

$$C_D^{\text{in}}(v) = \frac{\deg^{\text{in}}(v)}{|\mathcal{V}| - 1}, \quad C_D^{\text{out}}(v) = \frac{\deg^{\text{out}}(v)}{|\mathcal{V}| - 1}$$

where:

- $\deg^{\text{in}}(v)$: The in-degree of node v (number of incoming edges).
- $\deg^{\text{out}}(v)$: The out-degree of node v (number of outgoing edges).

 **Note.** Số đo này giúp đo số lượng các mối quan hệ trực tiếp của một tác nhân với các thành viên khác trong mạng xã hội ■


3.1.3 Closeness Centrality

Closeness centrality measures the average length of the shortest paths from a node v to all other nodes. It quantifies how quickly information spreads from v to other nodes in the network.

$$C_C(v) = \frac{|\mathcal{V}| - 1}{\sum_{u \in \mathcal{V}, u \neq v} d(v, u)}$$

where:

- $d(v, u)$: The shortest path distance between nodes v and u .
- $|\mathcal{V}|$: The total number of nodes in the graph.

 **Note.** Số đo này tương ứng với thời gian cần thiết để thông tin truyền từ một actor tới các actor khác. Khoảng cách càng nhỏ, khả năng truyền tin càng lớn ■

3.1.4 Betweenness Centrality

Betweenness centrality quantifies the importance of a node as a bridge for information flow between other nodes. It is defined as:

$$C_B(v) = \sum_{s \neq v \neq t \in \mathcal{V}} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where:

- σ_{st} : The total number of shortest paths between nodes s and t .
- $\sigma_{st}(v)$: The number of those shortest paths that pass through node v .

4. Betweenness Centrality (Normalized)

Betweenness centrality measures the importance of a node v in facilitating information flow by counting how often it lies on shortest paths between other nodes.

Unnormalized Form

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where:

- σ_{st} : The total number of shortest paths between nodes s and t .
- $\sigma_{st}(v)$: The number of shortest paths between s and t that pass through node v .

Normalized Form

Normalization ensures that the centrality score lies in the range $[0, 1]$. The normalization factor depends on whether the graph is directed or undirected.

For undirected graphs:

$$C_B^{\text{norm}}(v) = \frac{C_B(v)}{\frac{(|V|-1)(|V|-2)}{2}}$$

where:


- $|V|$: The total number of nodes in the graph.
- $\frac{(|V|-1)(|V|-2)}{2}$: The total number of pairs of nodes (excluding v).

For directed graphs:

$$C_B^{\text{norm}}(v) = \frac{C_B(v)}{(|V|-1)(|V|-2)}$$

where:

- $(|V|-1)(|V|-2)$: The total number of ordered pairs of nodes (excluding v).

 **Note.** Số đo này càng lớn thì actor càng quan trọng trong việc kiểm soát thông tin và giao dịch trong mạng. ThS. Lê Nhật Tùng

■

3.1.5 Clustering Coefficient

The clustering coefficient measures the tendency of a node's neighbors to form a complete subgraph (triangle).

- **For undirected graphs:**

$$C(v) = \frac{2T(v)}{\deg(v)(\deg(v)-1)}$$

where $T(v)$ is the number of triangles involving node v .

- **For directed graphs:**

$$C(v) = \frac{T(v)}{\deg^{\text{in}}(v) \deg^{\text{out}}(v) - \deg^{\text{loop}}(v)}$$

where:

- $\deg^{\text{loop}}(v)$: The number of self-loops at node v .

6. Jaccard Similarity

The Jaccard similarity index measures the similarity between the neighbor sets of two nodes:

$$J(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

where $N(u)$ and $N(v)$ represent the sets of neighbors of nodes u and v , respectively.

7. Adamic-Adar Index

The Adamic-Adar index evaluates the importance of common neighbors between two nodes:

$$A(u, v) = \sum_{w \in N(u) \cap N(v)} \frac{1}{\log(\deg(w))}$$

where $\deg(w)$ is the degree of node w .

Modularity in Community Detection

Modularity measures the quality of a graph partition P by comparing the actual edge density within communities to the expected edge density in a random graph.

General Formula

$$Q = \frac{1}{2|E|} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2|E|} \right] \delta(c_i, c_j)$$

where:

- A_{ij} : Adjacency matrix element; $A_{ij} = 1$ if there is an edge between nodes i and j , otherwise $A_{ij} = 0$.
- $k_i = \sum_j A_{ij}$: Degree of node i .
- $|E|$: Total number of edges in the graph.
- c_i, c_j : Communities to which nodes i and j belong.
- $\delta(c_i, c_j)$: Kronecker delta, $\delta(c_i, c_j) = 1$ if $c_i = c_j$, otherwise $\delta(c_i, c_j) = 0$.

Directed Graphs

For directed graphs, modularity accounts for in-degrees and out-degrees:

$$Q = \frac{1}{|E|} \sum_{i,j} \left[A_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{|E|} \right] \delta(c_i, c_j)$$

where:

- $k_i^{\text{out}} = \sum_j A_{ij}$: Out-degree of node i .
- $k_j^{\text{in}} = \sum_i A_{ij}$: In-degree of node j .

Interpretation of Modularity

- $Q > 0$: The graph has significant community structure.
- $Q = 0$: The graph has no more community structure than a random graph.

Steps to Compute Modularity

- Construct the adjacency matrix A .
- Compute node degrees (k_i , k_i^{in} , or k_i^{out}).
- Partition the nodes into communities c_1, c_2, \dots, c_k .
- Use the appropriate formula to compute Q .

8. Community Detection Algorithms

8.1. Louvain Algorithm

The Louvain algorithm detects communities by optimizing modularity in a hierarchical manner.

Pseudo-code:

- Initialize each node as its own community.
- Repeat until no improvement:
 - For each node:
 - Move it to the community that maximizes modularity.
- Aggregate nodes in the same community into a single node.
- Repeat steps 1-3 on the aggregated graph.

8.2. Girvan-Newman Algorithm

The Girvan-Newman algorithm divides a graph into communities by iteratively removing edges with the highest betweenness centrality.

Pseudo-code:

1. Compute betweenness centrality for all edges.
2. Remove the edge with the highest betweenness centrality.
3. Repeat until the graph is divided into the desired number of communities.

9. PageRank Algorithm

PageRank is a ranking algorithm that assigns scores to nodes based on their importance.

$$PR(v) = (1 - d) + d \sum_{u \in \text{In}(v)} \frac{PR(u)}{\text{deg}^{\text{out}}(u)}$$

where:

- d : Damping factor (typically 0.85).
- $\text{In}(v)$: The set of nodes with edges pointing to v .
- $\text{deg}^{\text{out}}(u)$: The out-degree of node u .

Pseudo-code:

1. Initialize $PR(v) = 1 / |\text{mathcal{V}}|$ for all nodes.
2. Repeat until convergence:
 - a. For each node v :

$$PR(v) = (1 - d) + d * \text{sum}(PR(u) / \text{out-degree}(u)) \text{ for all } u \text{ pointing to } v$$

9. PageRank Algorithm with Matrix Representation

The PageRank algorithm can be expressed in matrix form, allowing for efficient computation in large-scale networks.

Mathematical Formulation

$$\mathbf{PR} = d \cdot \mathbf{A} \cdot \mathbf{PR} + (1 - d) \cdot \mathbf{e}$$

where:

- \mathbf{PR} : A column vector of PageRank scores for all nodes.

- d : The damping factor, typically set to 0.85, representing the probability of continuing a random walk.
- \mathbf{A} : The transition matrix, defined as:

$$A_{ij} = \begin{cases} \frac{1}{\deg^{\text{out}}(j)} & \text{if there is a link from node } j \text{ to node } i, \\ 0 & \text{otherwise.} \end{cases}$$

- \mathbf{e} : A vector of size $|\mathcal{V}|$ with all elements equal to $\frac{1}{|\mathcal{V}|}$.

Iterative Solution

The PageRank vector \mathbf{PR} is computed iteratively:

$$\mathbf{PR}^{(k+1)} = d \cdot \mathbf{A} \cdot \mathbf{PR}^{(k)} + \frac{1-d}{|\mathcal{V}|} \cdot \mathbf{1}$$

where $\mathbf{1}$ is a column vector with all entries equal to 1.

Handling Dead Ends and Spider Traps

- **Dead ends**: Nodes with no outgoing edges are resolved by redistributing their probabilities uniformly across all nodes.
- **Spider traps**: Subgraphs that trap random walks are handled by introducing the damping factor d .

Pseudo-code

Input: Adjacency matrix A , damping factor d , tolerance ϵ

Output: PageRank vector \mathbf{PR}

1. Initialize \mathbf{PR} with uniform values: $\mathbf{PR}(i) = 1 / |\mathcal{V}|$ for all i .
2. Normalize A into the transition matrix M :
 $M(i, j) = A(i, j) / \text{out-degree}(j)$ if $\text{out-degree}(j) > 0$, otherwise $1 / |\mathcal{V}|$.
3. Repeat:
 - a. Compute new \mathbf{PR} :
 $\mathbf{PR}_{\text{new}} = d \cdot M \cdot \mathbf{PR} + (1 - d) / |\mathcal{V}| \cdot \mathbf{1}$.
 - b. Check convergence:
If $||\mathbf{PR}_{\text{new}} - \mathbf{PR}|| < \epsilon$, stop.
 - c. Update \mathbf{PR} : $\mathbf{PR} = \mathbf{PR}_{\text{new}}$.
4. Return \mathbf{PR} .

Expanded Matrix Formulation

In matrix notation:

$$\mathbf{PR} = (1 - d) \cdot \mathbf{e} + d \cdot \mathbf{M} \cdot \mathbf{PR}$$

The equation can also be solved directly as:

$$\mathbf{PR} = (I - d \cdot \mathbf{M})^{-1} \cdot (1 - d) \cdot \mathbf{e}$$

where I is the identity matrix, and \mathbf{M} is the stochastic matrix derived from the adjacency matrix.

BIBLIOGRAPHY

- [Tho17] Max Welling Thomas N. Kipf. “Semi-Supervised Classification with Graph Convolutional Networks.” In: (2017). URL: <https://arxiv.org/abs/1609.02907>.
- [Vel+18] Petar Veličković et al. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML].
- [Jun19] Jaewoo Kang Junhyun Lee Inyeop Lee. “Self-Attention Graph Pooling.” In: (2019). URL: <https://arxiv.org/abs/1904.08082>.
- [Lin+23] Xuan Lin et al. “Comprehensive evaluation of deep and graph learning on drug–drug interactions prediction.” In: *Briefings in Bioinformatics* 24.4 (July 2023). ISSN: 1477-4054. DOI: 10.1093/bib/bbad235. URL: <http://dx.doi.org/10.1093/bib/bbad235>.
- [Mei23] Xiujuan Lei Mei Ma. “A dual graph neural network for drug–drug interactions prediction based on molecular structure and interactions.” In: (2023). URL: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1010812>.
- [Zha+23] Yongqi Zhang et al. *Emerging Drug Interaction Prediction Enabled by Flow-based Graph Neural Network with Biomedical Network*. 2023. arXiv: 2311.09261 [q-bio.QM].
- [Kad+24] Amit Kadan et al. *Guided Multi-objective Generative AI to Enhance Structure-based Drug Design*. 2024. arXiv: 2405.11785 [physics.chem-ph].