Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(7), 142-159
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

An In-Depth Analysis of Graph Neural Networks and Machine Learning Approaches for Drug Repositioning

Swapnil Biswas[1], Shraboni Biswas[2], Nusrat Sharmin[3]

[1]Department of Computer Science and Engineering, Kishoreganj University, Kishoreganj, Bangladesh
[2]Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh
[3]Department of Computer Science and Engineering, Military Institute of Science and Technology, Dhaka, Bangladesh
Email: swapnil.cse16@gmail.com

How to cite this paper: Biswas, S., Biswas, S. and Sharmin, N. (2025) An In-Depth Analysis of Graph Neural Networks and Machine Learning Approaches for Drug Repositioning. *Journal of Computer and Communications*, **13**, 142-159.
https://doi.org/10.4236/jcc.2025.137007

Received: June 18, 2025
Accepted: July 19, 2025
Published: July 22, 2025

Abstract

Drug repositioning aims to identify new therapeutic applications for existing drugs offering a faster and more cost-effective alternative to traditional drug discovery. Since approved drugs already have known safety profiles, this approach is especially valuable in urgent situations like pandemic. In this study, a computational method was explored for drug repositioning using both graph-based representation for Graph Neural Networks (GNN) and feature-based representations for Machine Learning (ML) classifiers. Both models were trained separately, and their prediction scores were combined to form an integrated model named TwinNetDR. This combined approach achieved the best performance, with a precision of 95.92%, outperforming the individual GNN and ML models. The results demonstrate the benefit of combining graph and feature-based learning for reliable drug repositioning.

Keywords

Drug Repositioning, Graph Neural Network, Machine Learning, Drug Disease Network, TwinNetDR

1. Introduction

Developing new drugs remains a major challenge in the biomedical field, despite advances in understanding diseases and biological systems [1]. Although research in pharmacology, biology, and genomics has progressed significantly, the process from drug discovery to approval is still long, costly, and uncertain [2]-[6]. On average, this process can take over ten years and cost billions of dollars, with a high failure rate, especially during early clinical trials due to unexpected side effects [7]-[10].

To deal with these problems, researchers have started focusing on drug repositioning as a useful approach. It focuses on finding new uses for already approved drugs, reducing both development time and cost [11] [12]. However, traditional repositioning still relies heavily on manual testing and analysis, which can take years. Computational drug repositioning addresses this limitation by using algorithms to discover complex patterns in large-scale biological data that humans may miss or take longer to identify [13]-[15]. These approaches can speed up the repositioning process, reduce the need for costly trials, and increase the chances of success [16]. Moreover, they offer more accurate predictions, improving the efficiency of drug development overall [17]-[20].

Among computational techniques, Machine Learning (ML) has gained wide attention for its ability to handle large and complex biological datasets and uncover useful patterns for drug discovery [21]. A common assumption in ML-based repositioning is that diseases with similar characteristics may respond to drugs that work in similar ways. This idea has been used to build models that predict unknown drug effects, including herbal compounds [22]. ML-based methods gained more attention during the COVID-19 pandemic, as quick and low-cost treatments were urgently needed. For example, Aghdam *et al.* proposed an ML framework to identify drug repositioning options for COVID-19 [23]. Other studies used matrix factorization as part of ML models to uncover drug-disease links related to the virus [24], showing the real-world value of ML in biomedical research.

Alongside ML, Graph Neural Networks (GNNs) have shown strong performance in tasks involving graph data, such as node classification [25]-[27], link prediction [28]-[30], graph classification [31]-[33], community detection [34]-[36], and anomaly detection [37]-[39]. Motivated by these successes, researchers have also explored GNNs for drug repositioning. For example, GDRnet approaches drug repositioning as a link prediction problem by using structural patterns in biomedical graphs [40]. Sun *et al.* proposed AdaDR, which combines graph convolution with node-specific and topological features for better predictions [41]. DRAGNN applies attention mechanisms in a heterogeneous graph to focus on important biomedical connections [42], and DTDGNN combines graph convolution and attention layers to improve the quality of learned features in biological networks [43].

Most of the existing works apply either ML or GNN individually for drug repositioning. However, an important question remains on whether combining both approaches can lead to better and more reliable predictions. From previous studies, it is evident that drug repositioning data can be represented in multiple forms: as structured, tabular data suitable for ML classifiers and as complex graph-based networks well-suited for GNN classifiers. Based on this, the study introduces **TwinNetDR**, a combined framework that brings both classifiers together to add a new framework for drug repositioning in drug research. First, a drug-disease feature vector (tabular) and association-network (graph) have been developed using protein association data. After that, a GNN is used to train from the graph structure, and an ML model is trained using tabular features from the same data. During evaluation their sigmoid prediction scores have been combined to make the final output more reliable.

The main contributions of this study are as follows:

1) This study combined ML and GNN models in a single framework for drug repositioning and proposed an integrated method referred to as TwinNetDR.

2) The study analyzed how different threshold values impact the model's performance.

2. Methodology

The methodology section is comprised of three main parts. First, the associations between drugs and disease have been represented in two ways: as feature vectors in a tabular format and as a network (graph) built using protein interaction data linked to the drugs and diseases. In the second part, GraphSAGE, a GNN classifier, and Random Forest (RF), a ML classifier, have been employed to predict potential drug repositioning candidates. In the final component, the prediction scores from both classifiers have been integrated through a decision algorithm to generate the final prediction of the proposed TwinNetDR.

2.1. Dataset

The dataset used in this study sourced from [44], is provided in CSV format. It contains information about three types of entities: drugs, diseases, and proteins. In total, the dataset includes 1186 drugs from DrugBank, 451 diseases from OMIM, and 1467 proteins from UniProt. The dataset contains three major binary interaction matrices that capture the relationships among these entities:

1) **Drug-Protein Interaction Matrix (DP):** A binary matrix of size 1186 × 1467, where each entry indicates whether a given drug interacts with a specific protein (1 for interaction, 0 for no interaction).

2) **Disease-Protein Interaction Matrix (XP):** A binary matrix of size 451 × 1467, representing the associations between diseases and proteins, indicating which proteins are linked to each disease.

3) **Drug-Disease Interaction Matrix (DX):** A binary matrix of size 1186 × 451, showing known therapeutic relationships where a drug is used to treat a particular disease (1 indicates treatment, 0 indicates no known association).

Dataset Splittings

For both ML and GNN models, the dataset has been split into 80% training and 20% testing sets. Output labels have been derived from the drug-disease association matrix D_X.

2.2. Construction of Negative Samples

In this study, negative samples are drug-disease pairs with no known association.

These are marked as 0 in the drug-disease matrix D_X. On the other hand, positive samples are marked as 1. To keep the dataset balanced, a fixed ratio r has been used. This means r times of positive samples have been chosen as negative samples randomly from the dataset. The ratio is defined in (1).

$$\text{Number of negative samples} = r \times \text{Number of positive samples} \tag{1}$$

If $r < 1$, the model will learn to focus more on positive associations. If $r > 1$, it could become biased toward negative predictions due to the imbalance. In this work, a value of $r = 1$ has been used.

2.3. Feature Vector Construction for ML Classifier

For each drug-disease pair, a feature vector is constructed by combining the protein association profiles of both entities. The presence or absence of proteins associated with a drug-disease pair (*i*, *j*), where *i* is the drug index and *j* is the disease index, is encoded in the feature vector for each protein *k* as outlined in (2).

$$p_{ijk} = \begin{cases} 0.0, & \text{if neither drug } i \text{ nor disease } j \text{ is associated with protein } k \\ 0.3, & \text{if only drug } i \text{ is associated with protein } k \\ 0.6, & \text{if only disease } j \text{ is associated with protein } k \\ 1.0, & \text{if both drug } i \text{ and disease } j \text{ are associated with protein } k \end{cases} \tag{2}$$

The label for each drug-disease pair is obtained from the Drug-disease Interaction matrix (DX) by converting the original two-dimensional structure into a one-dimensional array, where each element denotes whether a known association exists (1) or not (0). This flattened array serves as the output label for the ML classifier. A small-scale example involving 3 drugs and 2 diseases is shown in Figure 1 to illustrate this flattening process.

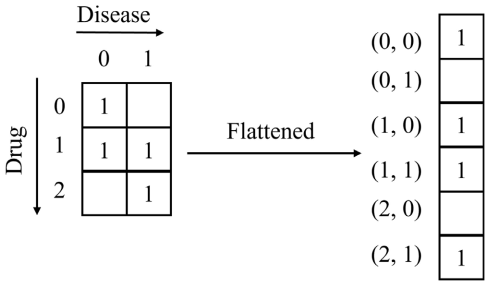


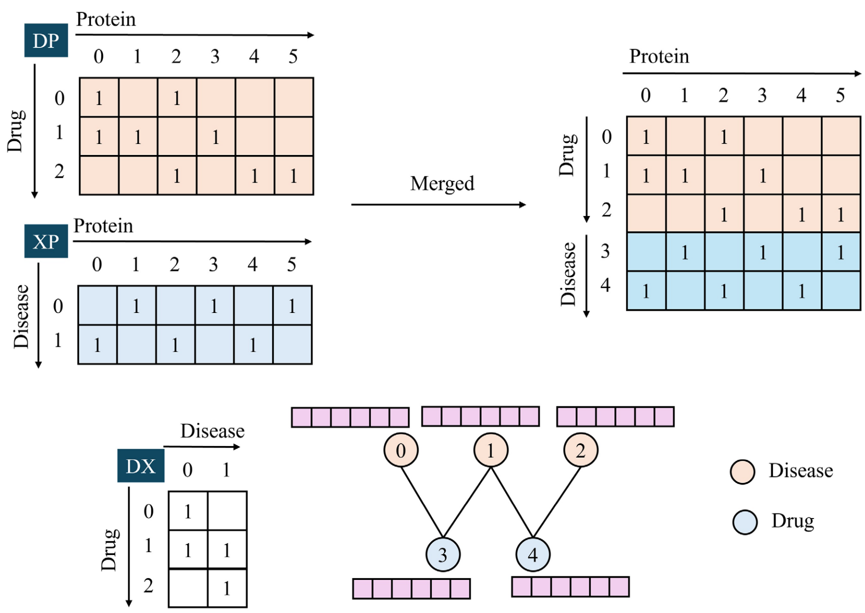
Figure 1. Drug-Disease matrix flattening for ML classifiers.

2.4. Graph Construction for GNN Classifier

A graph has been constructed from the dataset to be used with a GNN classifier. In this graph, drugs and diseases have been represented as nodes. Each node has association with a binary feature vector that reflects its interactions with proteins. Let *p* be the total number of proteins in the dataset, where *p* = 1467. Accordingly, each drug and disease node has been associated with a binary vector of length 1467. These feature vectors have been derived from two binary association matrices defined in the dataset: the Drug-Protein (DP) matrix and the Disease-Protein (XP) matrix.

The graph has been modeled as a bipartite structure consisting of two distinct types of nodes: drugs and diseases. Edges between them have been defined using the Drug-Disease interaction matrix (DX). An entry $DX_{i,j} = 1$ means there is a known link between drug *i* and disease *j*, and results in an edge between the corresponding nodes. Entries where $DX_{i,j} = 0$ are treated as negative edges. The total count of positive edges is represented as E^+, corresponds to the number of ones in the DX matrix. In this dataset, the number of known drug-disease associations is $\left|E^+\right| = 1827$.

All nodes, regardless of type, share the same feature space. Each node v_i is associated with a binary feature vector x_i of dimension p. The total number of nodes in the graph is $|V| = n_d + n_x$, where n_d is the number of drugs and n_x is the number of diseases. To distinguish between the two node types, drugs have been assigned node IDs from 0 to $n_d - 1$, and diseases from n_d to $n_d + n_x - 1$. A simplified example of the graph construction process involving three drugs, two diseases, and six proteins has been presented in **Figure 2**.



DP (Drug \ Protein)	0	1	2	3	4	5
0	1		1			
1	1	1		1		
2			1		1	1

XP (Disease \ Protein)	0	1	2	3	4	5
0		1		1		1
1	1		1		1	

Merged (Drug / Disease \ Protein)	0	1	2	3	4	5
0	1		1			
1	1	1		1		
2			1		1	1
3		1		1		1
4	1		1		1	

DX (Drug \ Disease)	0	1
0	1	
1	1	1
2		1

Figure 2. Graph construction process from dataset.

2.5. Drug Repositioning Using ML

For using an ML classifier in drug repositioning, the task is set up as a binary classification problem. The goal is to predict if a specific drug-disease pair shows a valid therapeutic connection (label 1) or not (label 0). Each entity in the dataset has been represented by a feature vector of length 1467, corresponding to the total number of proteins. Due to this high dimensionality, the Random Forest classifier has been chosen for its ability to efficiently handle high-dimensional data in this study.

The implementation workflow has been illustrated in **Figure 3**. It has started with the preparation of feature vectors for each drug-disease pair using information extracted from the dataset. These feature vectors have been used to train the Random Forest classifier, which has learned to distinguish between valid and invalid treatment associations. Later, the model has been evaluated using standard performance metrics. Afterwards, drug-disease pairs predicted as valid treatments but absent in the original dataset have been identified as potential candidates for drug repositioning.

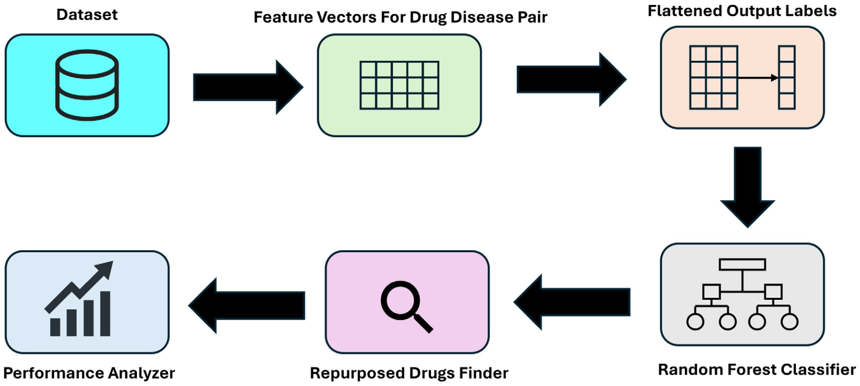


Figure 3. Implementation pipeline for identifying repurposed drugs using random forest.

The key hyperparameters of the Random Forest classifier used in our experiments are summarized in Table 1.

Table 1. Hyperparameters for the random forest model.

Parameter	Value
n_estimators	100
max_depth	20
class_weight	balanced
max_features	sqrt
random_state	42

2.6. Drug Repositioning Using GNN

In this study, drug repositioning has been treated as a link prediction problem. The goal is to predict if a connection (or edge) exists between a drug and a disease, indicating if the drug can potentially treat the disease. To solve this, a Graph Neural Network (GNN) model, GraphSAGE has been used. GraphSAGE is suitable for this task because, unlike some traditional GNNs that need the full graph during training, it learns from a node's neighbors by sampling and combining their features. This makes it useful for large biological networks, where new drugs or diseases might appear later, and the model can still make predictions for them.

The implementation has been done in several steps. First, a graph has been constructed using the dataset, where drugs and diseases are nodes, and edges show known links between them. Then, GraphSAGE layers have been applied to generate embeddings for each node by collecting information from nearby nodes. Finally, a decoder has been used that takes the embeddings of a drug and a disease and predicts whether a link exists between them. The complete workflow has been shown in Figure 4, and the core components involved in this process are explained in 2.6.1 and 2.6.2.

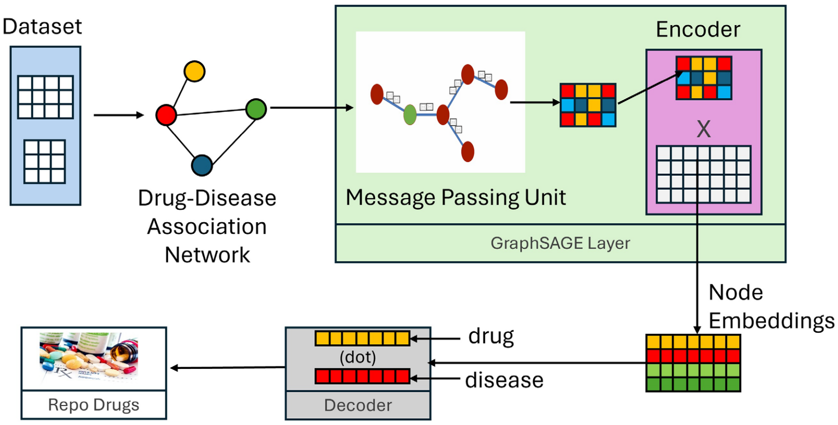


Figure 4. Implementation pipeline for identifying repurposed drugs using GraphSAGE.

Table 2 shows the selected hyperparameters used for the GraphSAGE model, and later the core components involved in this process are explained in 2.6.1 and 2.6.2.

Table 2. Key hyperparameters used for the GraphSAGE model.

Parameter	Value
Layers	2
Aggregator	mean
Hidden size	128
Learning rate	0.01
Epochs	200
Dropout	0.5
Optimizer	Adam
Loss function	Binary cross-entropy
Seed	42

2.6.1. Encoder for Generating the Node Embeddings

The encoder in GraphSAGE transforms the high-dimensional protein association features of each node into compact and meaningful vector representations called embeddings. These embeddings include information not only about the node itself but also about its nearby nodes. By capturing information about neighbors, the model learns to recognize drugs and diseases that are similar based on their protein associations. This helps the model recommend similar drugs for similar diseases during evaluation. Later, using this strategy, repurposed drugs can be identified. The encoding process follows the steps below:

1) The encoder takes two inputs: a feature matrix and an edge index. Each node representing a drug or a disease starts with a feature vector of 1467 values, based on protein interaction profiles. The edge index defines the connections between nodes in the graph.

2) After input, the encoder uses GraphSAGE layers to transform the original features into lower-dimensional embeddings. In each layer, node features are multiplied by trainable weight matrices. Then, each node updates its features by combining its own transformed values with information received from neighboring nodes. This process allows the model to understand the structure of the graph as well as from node-level data. As the data passes through the layers, the feature dimensions change from 1467 to 128 in the initial layer, and then to 32 at the next layer.

3) Afterwards, the training of the model uses the Binary Cross-Entropy (BCE) loss function. This loss helps guide the learning process by measuring how well the predicted outputs match the actual labels. Minimizing this loss leads to more accurate and meaningful node embeddings

2.6.2. Decoder for Link Prediction

Once the Encoder produces the node embeddings, the Decoder predicts whether an edge exists between a drug and a disease. The decoding process works as follows:

1) First, at first, the model is trained using the Encoder on both existing drug-disease links (positive edges) and absent links (negative edges) from the dataset. The purpose is to learn the model to predict whether a connection exists between a given drug and disease.

2) After the encoding, each drug and disease has its own embedding vector generated by the encoder. These vectors are low-dimensional representations that capture important features of each node. The decoder then calculates the similarity between a drug and a disease by taking the dot product of their embedding vectors. A higher similarity indicates a more significant link between the drug and the disease.

3) The dot product output undergoes a nonlinear activation function, the sigmoid function in this study to map the predicted value into a probabilistic range [0, 1]. This probability is then compared against a predefined threshold τ . If the probability exceeds τ , the model infers the presence of an edge, otherwise, it infers its absence as explained in (3) where H denotes the embedding of nodes.

$$y=\begin{cases}1, & \text{if } \sigma\left(H_{\text{drug}} \,.\, H_{\text{disease}}\right)>\tau \\ 0, & \text{otherwise}\end{cases} \tag{3}$$

2.7. TwinNetDR: Integrating GNN and ML

TwinNetDR improves drug repositioning predictions by combining GraphSAGE and Random Forest outputs. Both models are trained on the same drug-disease pairs for consistent evaluation. During testing, each model independently predicts associations, and TwinNetDR merges these predictions for the final decision. The workflow is shown in **Figure 5**. The following subsections explain how TwinNetDR handles rare repositioning cases and its decision-making strategy.

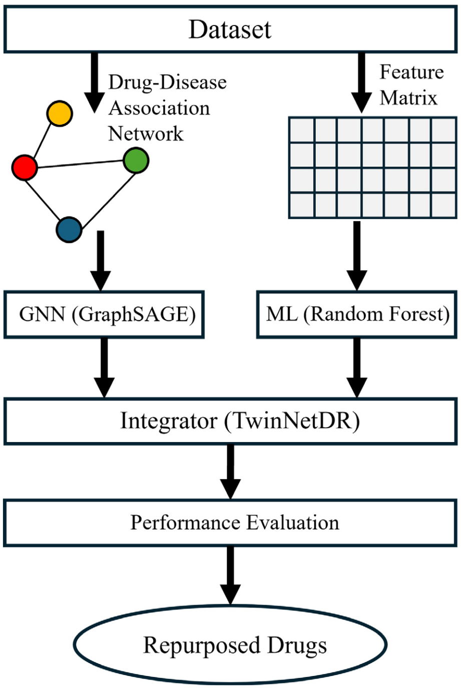


Figure 5. Workflow of TwinNetDR.

2.7.1. Handling Repurposed Drugs

Identifying new purposes for existing drugs is not common because true repositioning cases are not frequent. When our system predicts a drug could treat a disease that it's not currently known to help, this could mean one of two things. First, it might be a mistake—what we call a false positive. Second, it might actually be a valuable new discovery that hasn't been documented yet. This uncertainty is exactly why we developed TwinNetDR. TwinNetDR works smarter by using two different methods together. It combines Random Forest, which analyzes the data in table format, with GraphSAGE, which examines how drugs and diseases connect in a network. This dual approach gives us much more reliable results. When one method might make an error, the other can often catch and correct it. More importantly, when both methods agree on a potential new use that isn't in our original data, we can be more confident it's worth further investigation rather than just being a computer error. This way, we reduce simple mistakes while still finding promising new treatment possibilities.

2.7.2. Decision Making Process in TwinNetDR

Since repositioning cases are rare, most false positives are likely errors. TwinNetDR prioritizes high-confidence predictions: it favors negative predictions (0) to minimize false positives. A drug-disease pair is flagged only if:

1) Both models agree: If GraphSAGE and Random Forest both predict an asso-

ciation ($P_{\text{SAGE}} = 1$ and $P_{\text{RF}} = 1$), the association is confirmed.

2) Models disagree: If one predicts an association and the other does not, TwinNetDR first calculates the average confidence score as described in (4).

$$P_{\text{avg}} = \frac{P_{\text{SAGE}} + P_{\text{RF}}}{2} \tag{4}$$

- If P_{avg} exceeds a threshold (τ), the association is accepted (y= 1).
- • Otherwise, it is rejected (y= 0).

The decision-making process is explained in Algorithm 1, where $P_{\text{SAGE}}(X_i)$ and $P_{\text{RF}}(X_i)$ represent the sigmoid scores produced for the i^{th} drug-disease pair X_i by the GraphSAGE and Random Forest models, respectively, during evaluation.

Algorithm 1 TwinNet-DR Decision Algorithm

```
for $i \leftarrow 1$ to $n$ do
    $y[i] = 0$
end
for $i \leftarrow 1$ to $n$ do
    if $p_{SAGE}[X_i] > \tau$ and $p_{RF}[X_i] > \tau$ then
        $y[i] = 1$
        continue
    end
    else
        $P_{\text{avg}} \leftarrow \frac{P_{\text{SAGE}}(X_i)+P_{\text{RF}}(X_i)}{2}$
        if $p_{avg} > \tau$ then
            $y[i] \leftarrow 1$
        end
    end
end
```

3. Findings and Result Analysis

This section presents the performance analysis of Random Forest and GraphSAGE individually. Following that, the results of the combined model, TwinNet-DR, are reported. Finally, the factors influencing performance differences among the three models are discussed, along with an explanation of how TwinNet-DR achieves improved outcomes.

3.1. Evaluation Metrics

After applying the classifiers, the performance of Random Forest, GraphSAGE, and TwinNetDR has been evaluated using standard classification metrics, including accuracy, recall, precision, and F1-score. These metrics provide a quantitative understanding of how well each model identifies valid drug-disease associations. The formulas used for calculating these metrics are provided in expressions (5) to (8).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

- **TP (True Positives):** TP denotes the number of true positives, meaning the model correctly predicts a positive case. In this study, TP means the model correctly predicts a drug-disease pair that is indeed a valid treatment.
- **TN (True Negatives):** TN stands for the number of true negatives, meaning the model correctly predicts a negative case. In this study, TN means the model correctly predicts that a drug-disease pair is not associated, so the drug cannot be used to repurpose for that disease.
- **FP (False Positives):** FP is the count of false positives, meaning the model predicts a positive case, but the actual case is negative. For drug repositioning, FP means the model predicts a drug-disease pair as associated (repurposed candidate), even though no such association exists in the dataset. These predicted pairs can represent potential new repurposing candidates suggested by the model.
- **FN (False Negatives):** FN denotes the number of false negatives, meaning the model predicts a negative case, but the actual case is positive. In this study, FN means the model fails to identify a drug-disease pair that truly has a valid treatment association, missing a possible repurposing opportunity. It is denoted as a misclassification by the model.

3.2. Result Analysis

The performance of Random Forest, GraphSAGE, and TwinNet-DR in identifying repurposed drugs has been evaluated. To compare them, several metrics have been calculated, such as accuracy, precision, recall, F1-score, and AUC (Area Under the Curve). The results are shown in Table 3. Among the three models, GraphSAGE performs better than Random Forest in all the metrics. For example, the AUC score of GraphSAGE is 95.17%, while Random Forest gets 93.90%. GraphSAGE also scores higher in accuracy, precision, recall, and F1-score. The most noticeable improvement is in recall, which shows how well the model finds true drug-disease associations. This improvement happens because GraphSAGE uses graph structure to learn from the data. Random Forest treats every drug-disease pair separately and does not consider how drugs and diseases are connected. On the other hand, GraphSAGE looks at the links between them. For example, if Drug-1 is connected to Disease-1 and Drug-2 is also connected to Disease-1, then in a graph, both drugs are neighbors of the same disease. GraphSAGE can learn this connection and understand that Drug-1 and Drug-2 might be similar. When it builds the feature for Disease-1, it combines information from both drugs. This kind of learning is not possible with Random Forest, which only sees each pair on its own.

Table 3. Comparison of the performance of various models.

Model	Accuracy	Precision	Recall	F1-score	AUC
Random Forest	84.15%	84.15%	84.15%	84.15%	93.90%
GraphSAGE	90.71%	87.44%	95.08%	91.10%	95.17%
TwinNet-DR	90.20%	95.92%	94.61%	95.26%	97.53%

TwinNet-DR performs even better. It achieves the highest AUC score of 97.53 percent, which is better than both GraphSAGE and Random Forest. Its accuracy is slightly lower than GraphSAGE, but it gives a much higher precision of 95.92 percent. In drug repositioning, high precision is very important because it means fewer false positives. If a model predicts fewer wrong drug-disease associations, its results are more trustworthy. So, TwinNet-DR provides more reliable suggestions for repurposed drugs with fewer errors. In addition, Figure 6 presents the combined ROC curves of all three models. The curve for TwinNet-DR lies closest to the top-left corner, clearly indicating superior overall performance compared to the others.

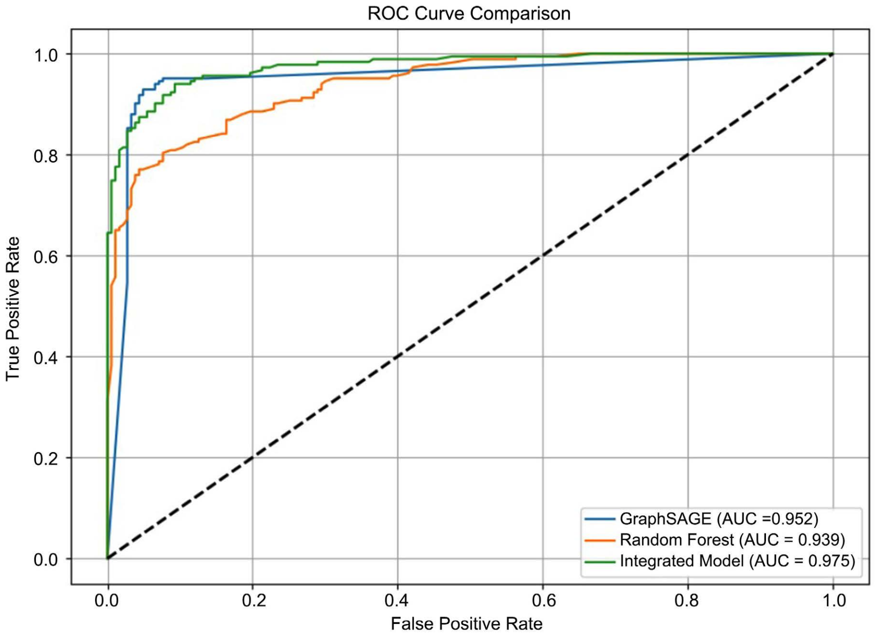


Figure 6. The ROC curves of GraphSAGE, Random Forest and TwinnetDR for Drug Repositioning.

3.3. Impact of Threshold on Performance of TwinNetDR

In this study, selecting an appropriate threshold is crucial. If the threshold is set too low, the model produces a large number of false positives. This leads to confusion between true repurposed candidates and incorrect ones, as many negative samples are mistakenly predicted as positive. Conversely, if the threshold is too high, the model becomes overly conservative, predicting most samples as negative. This drastically reduces the chance of identifying true repurposed drug-disease pairs, rendering the model ineffective. Even a midpoint threshold of 0.5 may not be ideal. Although it offers a balanced outcome, it lacks the necessary bias toward predicting negatives in this study's context.

The model was tested using different decision thresholds. For each threshold, precision, recall, and F1-score were calculated. The precision-recall curve in **Figure 7** shows how performance changed with varying thresholds. The best performance was observed at threshold 0.6, where the model achieved an F1-score of 95.26%, with precision 95.92% and recall 94.61%. This threshold also gave the highest accuracy of 90.20%. Based on these results, threshold 0.6 was selected as the final decision point.

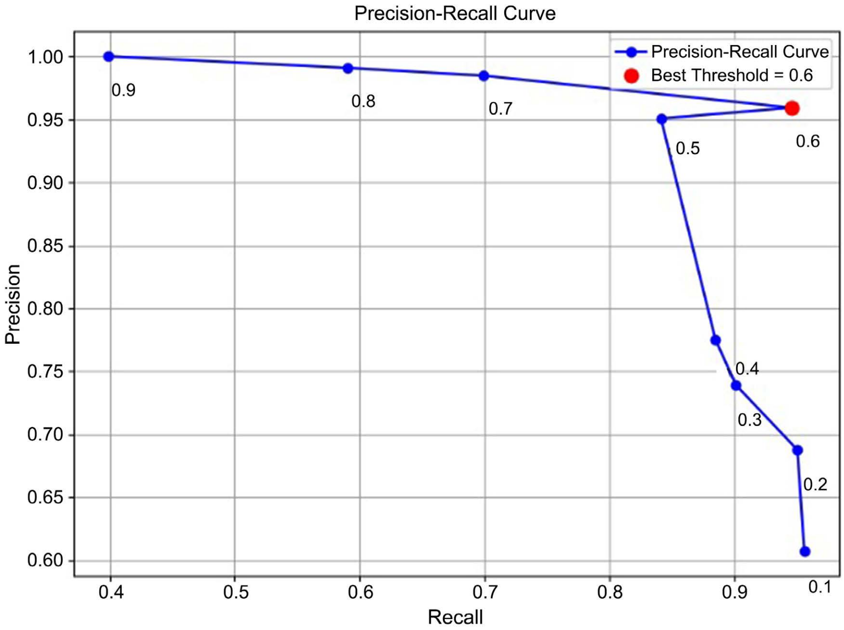


Figure 7. Impact of Threshold on the performance of TwinNetDR.

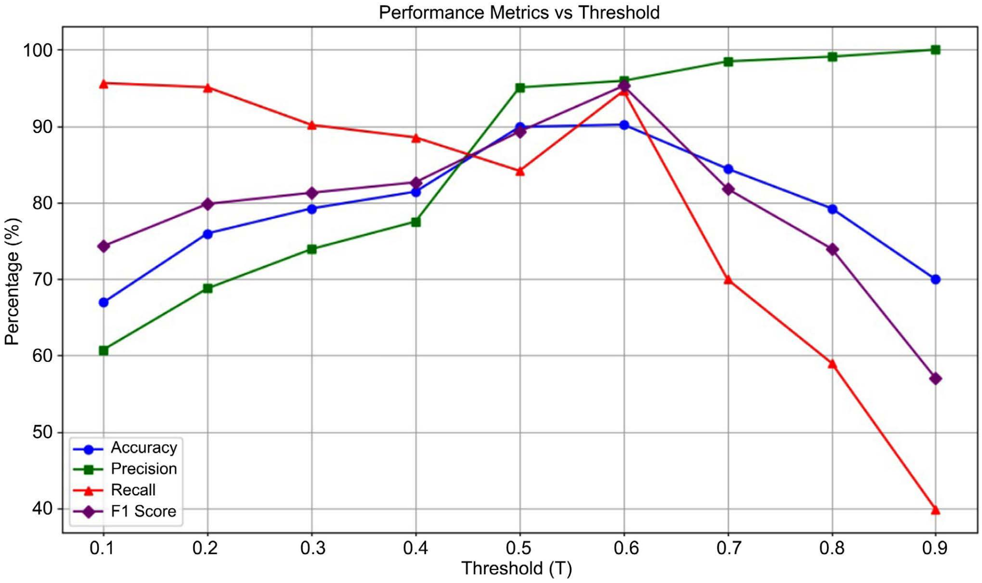


Figure 8. Impact of Threshold on the performance of TwinNetDR.

Figure 8 further illustrates how the performance metrics vary with changes in the threshold. As the threshold increases, accuracy, recall, and F1-score show a rising trend up to a certain point. Notably, at a threshold of 0.6, most of these metrics reach their highest values, indicating that this point offers the best balance between identifying true positives and minimizing false positives for this study.

In summary, extremely low or high thresholds reduce model effectiveness. A threshold of 0.6 provides the best balance, providing the best accuracy while maintaining a good balance between precision and recall.

3.4. Top Prediction by TwinNetDR

The highest ranked drug-disease association predicted by the proposed TwinNetDR is between Amoxapine, with DrugBank ID DB00543 and Drug ID 922 in the dataset, and Diabetes Mellitus, with OMIM ID 125853 and Disease ID 42 in the dataset. This association received a sigmoid score of **91.3%** from TwinNet-DR. The same association scored **88.5%** using the Random Forest model and **94.1%** using the GraphSAGE model. This pair was not part of the known positive samples. This prediction is also supported by the study of Li *et al.* [43], which reported Amoxapine as a candidate for diabetes treatment using a drug–target–disease graph neural network.

4. Conclusions

In this study, both traditional machine learning and graph neural network models have been explored to address the problem of drug repurposing. A dataset containing known and unknown drug-disease associations was used to evaluate and compare the performance of Random Forest, GraphSAGE, and TwinNet-DR. Several evaluation metrics such as accuracy, precision, recall, F1-score, and AUC have been used to assess the models. Among all, TwinNet-DR has shown the best performance with the highest precision of 95.92%, followed by GraphSAGE and Random Forest. The experimental results have confirmed that learning from neighborhood relationships from graph based representation of drug-disease association network can enhance model performance. Furthermore, a threshold-based analysis has been conducted to understand the sensitivity of the model and to select the optimal threshold for balancing precision and recall. A threshold of 0.6 has been identified as the most suitable choice.

Although the proposed approach has demonstrated promising results, there are certain limitations that create opportunities for future improvements. This study has used one graph neural network model and one machine learning model to explore drug repurposing. In the future, other graph-based models such as Graph Convolutional Network (GCN) and Graph Attention Network (GAT) can be applied to observe how they perform in this task. From traditional machine learning, models like Support Vector Machine (SVM) and Logistic Regression (LR) can also be considered to compare results. Moreover, the current system has been tested only on the dataset used during training. It can be improved to handle new or unseen data in future studies. Using more diverse and larger datasets can also help to make the system more useful for real-world applications.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Rifaioglu, A.S., Atas, H., Martin, M.J., Cetin-Atalay, R., Atalay, V. and Doğan, T. (2018) Recent Applications of Deep Learning and Machine Intelligence on in Silico Drug Discovery: Methods, Tools and Databases. *Briefings in Bioinformatics*, **20**, 1878-1912. https://doi.org/10.1093/bib/bby061

[2] Li, J., Zheng, S., Chen, B., Butte, A.J., Swamidass, S.J. and Lu, Z. (2015) A Survey of Current Trends in Computational Drug Repositioning. *Briefings in Bioinformatics*, **17**, 2-12. https://doi.org/10.1093/bib/bbv020

[3] Paul, S.M., Mytelka, D.S., Dunwiddie, C.T., Persinger, C.C., Munos, B.H., Lindborg, S.R., *et al.* (2010) How to Improve R&D Productivity: The Pharmaceutical Industry's Grand Challenge. *Nature Reviews Drug Discovery*, **9**, 203-214. https://doi.org/10.1038/nrd3078

[4] Zong, N., Wen, A., Moon, S., Fu, S., Wang, L., Zhao, Y., *et al.* (2022) Computational Drug Repurposing Based on Electronic Health Records: A Scoping Review. *npj Digital Medicine*, **5**, Article No. 77. https://doi.org/10.1038/s41746-022-00617-6

[5] Chan, H.C.S., Shan, H., Dahoun, T., Vogel, H. and Yuan, S. (2019) Advancing Drug Discovery via Artificial Intelligence. *Trends in Pharmacological Sciences*, **40**, 592-604. https://doi.org/10.1016/j.tips.2019.06.004

[6] Prasad, V. and Mailankody, S. (2017) Research and Development Spending to Bring a Single Cancer Drug to Market and Revenues after Approval. *JAMA Internal Medicine*, **177**, 1569-1575. https://doi.org/10.1001/jamainternmed.2017.3601

[7] Krantz, A. (1998) Diversification of the Drug Discovery Process. *Nature Biotechnology*, **16**, 1294-1294. https://doi.org/10.1038/4243

[8] Parvathaneni, V., Kulkarni, N.S., Muth, A. and Gupta, V. (2019) Drug Repurposing: A Promising Tool to Accelerate the Drug Discovery Process. *Drug Discovery Today*, **24**, 2076-2085. https://doi.org/10.1016/j.drudis.2019.06.014

[9] DiMasi, J.A., Grabowski, H.G. and Hansen, R.W. (2016) Innovation in the Pharmaceutical Industry: New Estimates of R&D Costs. *Journal of Health Economics*, **47**, 20-33. https://doi.org/10.1016/j.jhealeco.2016.01.012

[10] Wong, C.H., Siah, K.W. and Lo, A.W. (2018) Estimation of Clinical Trial Success Rates and Related Parameters. *Biostatistics*, **20**, 273-286. https://doi.org/10.1093/biostatistics/kxx069

[11] Ashburn, T.T. and Thor, K.B. (2004) Drug Repositioning: Identifying and Developing New Uses for Existing Drugs. *Nature Reviews Drug Discovery*, **3**, 673-683. https://doi.org/10.1038/nrd1468

[12] Xue, H., Li, J., Xie, H. and Wang, Y. (2018) Review of Drug Repositioning Approaches and Resources. *International Journal of Biological Sciences*, **14**, 1232-1244. https://doi.org/10.7150/ijbs.24612

[13] Luo, H., Li, M., Yang, M., Wu, F., Li, Y. and Wang, J. (2020) Biomedical Data and Computational Models for Drug Repositioning: A Comprehensive Review. *Briefings in Bioinformatics*, **22**, 1604-1619. https://doi.org/10.1093/bib/bbz176

[14] Zhao, Q., Yu, H., Ji, M., Zhao, Y. and Chen, X. (2019) Computational Model Development of Drug-Target Interaction Prediction: A Review. *Current Protein & Peptide Science*, **20**, 492-494. https://doi.org/10.2174/1389203720666190123164310

[15] Martínez, V., Navarro, C., Cano, C., Fajardo, W. and Blanco, A. (2015) DrugNet: Network-Based Drug-Disease Prioritization by Integrating Heterogeneous Data. *Artificial Intelligence in Medicine*, **63**, 41-49. https://doi.org/10.1016/j.artmed.2014.11.003

[16] Baker, N.C., Ekins, S., Williams, A.J. and Tropsha, A. (2018) A Bibliometric Review of Drug Repurposing. *Drug Discovery Today*, **23**, 661-672. https://doi.org/10.1016/j.drudis.2018.01.018

[17] Shim, J.S. and Liu, J.O. (2014) Recent Advances in Drug Repositioning for the Discovery of New Anticancer Drugs. *International Journal of Biological Sciences*, **10**, 654-663. https://doi.org/10.7150/ijbs.9224

[18] Mohamed, K., Yazdanpanah, N., Saghazadeh, A. and Rezaei, N. (2021) Computational Drug Discovery and Repurposing for the Treatment of COVID-19: A Systematic Review. *Bioorganic Chemistry*, **106**, Article 104490. https://doi.org/10.1016/j.bioorg.2020.104490

[19] Traylor, J.I., Sheppard, H.E., Ravikumar, V., Breshears, J., Raza, S.M., Lin, C.Y., *et al.* (2020) Computational Drug Repositioning Identifies Potentially Active Therapies for Chordoma. *Neurosurgery*, **88**, 428-436. https://doi.org/10.1093/neuros/nyaa398

[20] Bai, L., Scott, M.K.D., Steinberg, E., Kalesinskas, L., Habtezion, A., Shah, N.H., *et al.* (2021) Computational Drug Repositioning of Atorvastatin for Ulcerative Colitis. *Journal of the American Medical Informatics Association*, **28**, 2325-2335. https://doi.org/10.1093/jamia/ocab165

[21] Jarada, T.N., Rokne, J.G. and Alhajj, R. (2020) A Review of Computational Drug Repositioning: Strategies, Approaches, Opportunities, Challenges, and Directions. *Journal of Cheminformatics*, **12**, Article No. 46. https://doi.org/10.1186/s13321-020-00450-7

[22] Kim, E., Choi, A. and Nam, H. (2019) Drug Repositioning of Herbal Compounds via a Machine-Learning Approach. *BMC Bioinformatics*, **20**, Article No. 247. https://doi.org/10.1186/s12859-019-2811-8

[23] Aghdam, R., Habibi, M. and Taheri, G. (2021) Using Informative Features in Machine Learning Based Method for COVID-19 Drug Repurposing. *Journal of Cheminformatics*, **13**, Article No. 70. https://doi.org/10.1186/s13321-021-00553-9

[24] de Siqueira Santos, S. and Torres, M. and Galeano, D. and Sánchez, M.M. and Cernuzzi, L. and Paccanaro, A. (2022) Machine Learning and Network Medicine Approaches for Drug Repositioning for Covid-19. *Patterns*, **3**, Article 100396. https://doi.org/10.1016/j.patter.2021.100396

[25] Zhang, Y., Xu, Y. and Zhang, Y. (2023) A Graph Neural Network Node Classification Application Model with Enhanced Node Association. *Applied Sciences*, **13**, Article 7150. https://doi.org/10.3390/app13127150

[26] Li, K., Huang, Z. and Jia, Z. (2023) RAHG: A Role-Aware Hypergraph Neural Network for Node Classification in Graphs. *IEEE Transactions on Network Science and Engineering*, **10**, 2098-2108. https://doi.org/10.1109/tnse.2023.3243058

[27] Wang, K., An, J., Zhou, M., Shi, Z., Shi, X. and Kang, Q. (2023) Minority-Weighted Graph Neural Network for Imbalanced Node Classification in Social Networks of Internet of People. *IEEE Internet of Things Journal*, **10**, 330-340. https://doi.org/10.1109/jiot.2022.3200964

[28] Cai, L., Li, J., Wang, J. and Ji, S. (2021) Line Graph Neural Networks for Link Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 5103-5113. https://doi.org/10.1109/tpami.2021.3080635

[29] Skarding, J., Hellmich, M., Gabrys, B. and Musial, K. (2022) A Robust Comparative Analysis of Graph Neural Networks on Dynamic Link Prediction. *IEEE Access*, **10**, 64146-64160. https://doi.org/10.1109/access.2022.3175981

[30] Chen, M., Huang, P., Lin, Y. and Cai, S. (2021) SSNE: Effective Node Representation for Link Prediction in Sparse Networks. *IEEE Access*, **9**, 57874-57885. https://doi.org/10.1109/access.2021.3073249

[31] Ji, J., Jia, H., Ren, Y. and Lei, M. (2023) Supervised Contrastive Learning with Structure Inference for Graph Classification. *IEEE Transactions on Network Science and Engineering*, **10**, 1684-1695. https://doi.org/10.1109/tnse.2022.3233479

[32] Gao, J., Gao, J., Ying, X., Lu, M. and Wang, J. (2021) Higher-Order Interaction Goes Neural: A Substructure Assembling Graph Attention Network for Graph Classification. *IEEE Transactions on Knowledge and Data Engineering*, **35**, 1594-1608. https://doi.org/10.1109/tkde.2021.3105544

[33] Xu, Y., Wang, J., Guang, M., Yan, C. and Jiang, C. (2023) Multistructure Graph Classification Method with Attention-Based Pooling. *IEEE Transactions on Computational Social Systems*, **10**, 602-613. https://doi.org/10.1109/tcss.2022.3169219

[34] Sun, J., Zheng, W., Zhang, Q. and Xu, Z. (2022) Graph Neural Network Encoding for Community Detection in Attribute Networks. *IEEE Transactions on Cybernetics*, **52**, 7791-7804. https://doi.org/10.1109/tcyb.2021.3051021

[35] Xie, H. and Ning, Y. (2023) Community Detection Based on BernNet Graph Convolutional Neural Network. *Journal of the Korean Physical Society*, **83**, 386-395. https://doi.org/10.1007/s40042-023-00823-9

[36] He, C., Zheng, Y., Fei, X., Li, H., Hu, Z. and Tang, Y. (2022) Boosting Nonnegative Matrix Factorization Based Community Detection with Graph Attention Auto-Encoder. *IEEE Transactions on Big Data*, **8**, 968-981. https://doi.org/10.1109/tbdata.2021.3103213

[37] Kim, H., Lee, B.S., Shin, W. and Lim, S. (2022) Graph Anomaly Detection with Graph Neural Networks: Current Status and Challenges. *IEEE Access*, **10**, 111820-111829. https://doi.org/10.1109/access.2022.3211306

[38] Wang, X., Jin, B., Du, Y., Cui, P., Tan, Y. and Yang, Y. (2021) One-Class Graph Neural Networks for Anomaly Detection in Attributed Networks. *Neural Computing and Applications*, **33**, 12073-12085. https://doi.org/10.1007/s00521-021-05924-9

[39] Daniel, G.V., Chandrasekaran, K., Meenakshi, V. and Paneer, P. (2023) Robust Graph Neural-Network-Based Encoder for Node and Edge Deep Anomaly Detection on Attributed Networks. *Electronics*, **12**, Article 1501. https://doi.org/10.3390/electronics12061501

[40] Doshi, S. and Chepuri, S.P. (2022) A Computational Approach to Drug Repurposing Using Graph Neural Networks. *Computers in Biology and Medicine*, **150**, Article 105992. https://doi.org/10.1016/j.compbiomed.2022.105992

[41] Sun, X., Jia, X., Lu, Z., Tang, J. and Li, M. (2023) Drug Repositioning with Adaptive Graph Convolutional Networks. *Bioinformatics*, **40**, btad748. https://doi.org/10.1093/bioinformatics/btad748

[42] Meng, Y., Wang, Y., Xu, J., Lu, C., Tang, X., Peng, T., *et al.* (2023) Drug Repositioning Based on Weighted Local Information Augmented Graph Neural Network. *Briefings in Bioinformatics*, **25**, bbad431. https://doi.org/10.1093/bib/bbad431

[43] Li, W., Ma, W., Yang, M. and Tang, X. (2024) Drug Repurposing Based on the DTD-GNN Graph Neural Network: Revealing the Relationships among Drugs, Targets and Diseases. *BMC Genomics*, **25**, Article No. 584. https://doi.org/10.1186/s12864-024-10499-5

[44] Wu, G. and Liu, J. (2019) Predicting Drug-Disease Treatment Associations Based on Topological Similarity and Singular Value Decomposition. 2019 *IEEE International Conference on Bioinformatics and Biomedicine* (*BIBM*), San Diego, 18-21 November 2019, 153-158. https://doi.org/10.1109/bibm47256.2019.8983205