



Research article

DTiGNN: Learning drug-target embedding from a heterogeneous biological network based on a two-level attention-based graph neural network

Saranya Muniyappan*, Arockia Xavier Annie Rayan and Geetha Thekkumpurath Varrieth

Computer Science and Engineering, CEG Campus, Anna University, Tamil Nadu, India

* **Correspondence:** Email: saranyamani18@gmail.com.

Abstract: Motivation: In vitro experiment-based drug-target interaction (DTI) exploration demands more human, financial and data resources. In silico approaches have been recommended for predicting DTIs to reduce time and cost. During the drug development process, one can analyze the therapeutic effect of the drug for a particular disease by identifying how the drug binds to the target for treating that disease. Hence, DTI plays a major role in drug discovery. Many computational methods have been developed for DTI prediction. However, the existing methods have limitations in terms of capturing the interactions via multiple semantics between drug and target nodes in a heterogeneous biological network (HBN). Methods: In this paper, we propose a DTiGNN framework for identifying unknown drug-target pairs. The DTiGNN first calculates the similarity between the drug and target from multiple perspectives. Then, the features of drugs and targets from each perspective are learned separately by using a novel method termed an information entropy-based random walk. Next, all of the learned features from different perspectives are integrated into a single drug and target similarity network by using a multi-view convolutional neural network. Using the integrated similarity networks, drug interactions, drug-disease associations, protein interactions and protein-disease association, the HBN is constructed. Next, a novel embedding algorithm called a meta-graph guided graph neural network is used to learn the embedding of drugs and targets. Then, a convolutional neural network is employed to infer new DTIs after balancing the sample using oversampling techniques. Results: The DTiGNN is applied to various datasets, and the result shows better performance in terms of the area under receiver operating characteristic curve (AUC) and area under precision-recall curve (AUPR), with scores of 0.98 and 0.99, respectively. There are 23,739 newly predicted DTI pairs in total.

Keywords: drug-target interaction prediction; similarity network integration; information entropy-based random walk; multi-view convolutional neural network; meta-graph-based representation learning; graph neural network

1. Introduction

In the field of drug discovery and development, the prediction of drug-target interaction (DTI) is a vital research field. DTI identifies the interaction between the chemical compound (a drug) and the target (a protein or gene) in the human body [1] through which the disease can be treated. In order to treat some particular disease, one should understand the drug's ability to bind with a particular target [2]. DTI prediction is essential for drug repurposing [3], side-effect prediction [4] and drug resistance [5]. However, the number of evaluated and recognized DTI pairs is limited. When a wet lab experiment is used for determining DTI, the process is very expensive and time-consuming. Currently, many researchers have started using computational methods to identify the DTIs to minimize these issues [3].

DTI prediction approaches are classified as follows: (i) docking simulation-based, (ii) ligand-based and (iii) chemogenomic-based approaches. Docking simulation-based methods [6–7] are limited, as they require the three-dimensional (3D) structure of the protein, which is not readily available for all of the target or protein families. In order to overcome the limitation of the docking simulation method, ligand-based approaches [8–9] are used to predict DTIs by coalescing a candidate ligand with the known ligands of the target protein. The performance of this method is poor because the known ligand information of the proteins is limited. Hence, chemogenomic approaches, which are generally computationally based, have been introduced to avoid the issues of the above-mentioned traditional methods. The study of systematically analyzing the interactions between chemical and biological entities is known as chemogenomic-based approaches [10]. The biological entities are known as targets, namely, kinases, GPCRs, ion channels (ICs), serine proteases, etc.

Many of the chemogenomic methods utilize drug-related information (e.g., chemical structure, ATC code), target-related information (e.g., protein structure, sequence) and known DTI information to predict unknown interactions between drug-target pairs. Most of the chemogenomic-based DTI prediction methods are categorized into three major groups: machine learning (ML) [11,12] methods, deep learning (DL) methods [13,14] and network-based inference (NBI) methods [15,16], which use ML or DL techniques. The literature has summarized and analyzed DTI prediction studies based on the above categories [17–19]. ML-based methods use two approaches: feature-based methods [20] in which the DTI pairs are represented as features; and similarity-based methods [21], which use the “guilt-by-association (GBA)” principle, where GBA states that similar targets may share a similar drug, and vice versa. On the other hand, DL-based methods [22] also predict the drug-target pairs by learning the features from the individual descriptors of drugs and targets respectively by using different kinds of deep neural network architectures, namely, convolutional neural networks (CNNs), long-short term memory, recurrent neural networks and graph convolutional networks (GCNs). Similarly, Bai et al. [23–25] developed many frameworks to design the drugs for the protein by using various DL algorithms.

Compared to these methods, NBI methods have shown significant advantages. These approaches visualize the data available in the database as a biological network [26] in which the entities are described as nodes and the known associations are described as edges. A biological network has

biomedical entities as nodes that are connected together through edges formed by interactions, associations or relations between the entities, such as a drug or compound, a target (protein, gene, enzyme, GPCR channel, miRNA, lncRNA, etc.), a disease, a side effect, a pathway and adverse drug events. The biological network can be homogeneous or heterogeneous [26]. A homogeneous biological network consists of one type of node and one type of edge, whereas a heterogeneous biological network (HBN) [26] is defined as $G = (V, E)$, where V denotes a set of nodes and E denotes a set of edges of the network, and V and E can be of different types.

Generally, heterogeneous biological NBI methods consider the presumption that similar entities reveal similar features. Along with this presumption, the NBI method can be interpreted as a link prediction problem to identify new associations (connections) between the entities in the HBN [15,16,27,28]. There are many applications in the field of the biomedical domain to find new associations or interactions between entities or nodes, such as drug-disease association, drug-drug interaction (DDI) protein-protein interaction, etc. NBI models [29] are generally based on the common neighborhood method, matrix factorization, matrix projection or diffusion, a random walk (RW), an RW with restart (RWR), substructure drug-target network-based inference, etc. The above methods [29] represent the network as matrices, and inferences can be carried out with the help of matrix operations, which are generally considered very efficient [29]. However, when the network becomes complex, either ML or DL methods are used to improve the performance of the system.

In recent times, DL and ML methods have been used in link prediction problems along with network embedding-based approaches [30], namely, node2vec, metapath2vec, metagraph2vec, GCNs, graph neural networks (GNNs), etc., which are more powerful. These models are used to represent or learn the features of the drug or target in their corresponding vector spaces. The following section gives detailed information about the existing literature on network embedding methods and their limitations.

2. Related works

In this section, we briefly discuss the existing methods or frameworks for DTI prediction using an HBN. DTINet [15] is such a system that integrates several types of drug- and target-related information. This HBN contains four types of nodes: drug, disease, protein and side effects, and six types of edges to identify unknown DTI pairs: DDI, drug-disease association, drug similarities, protein similarities, protein-protein interaction and protein-disease association. The DTINet learns the low-dimensional feature vector representation by matrix factorization in order to predict the DTI pairs. However, the DTINet is not suitable for predicting DTIs when the size of the network increases, because the number of features that represent each node in the network also increases [30]. Hence, DL techniques are mainly used to learn the features of this large network [31]. Wang et al. [32], Agyemang et al. [33], Zhang et al. [34] and Chen et al. [35] developed DL-based models that use the drug molecular structure and protein sequence information to generate feature descriptors for DTI prediction. However, the above works have limited perspectives on the similarities of drugs and targets.

The performance of the DL-based prediction system is improved when information related to the drug and target is collected from several sources rather than from a single source [31]. Hence, Zeng et al. [36] proposed a deepDTnet system that utilizes multiple perspectives on drug and target similarities to increase the performance of DTI prediction. However, the similarities considered from multiple sources or perspectives were not integrated into a single similarity network in order to respectively get the feature vector representation of the drug and target. Therefore, similarity network integration (SNI)

is an essential step in any DTI prediction model in order to avoid biased conclusions among the different types of similarities, and also to improve the prediction accuracy. Therefore, these multiple perspectives of drug similarity networks (DSNs) and target similarity networks (TSNs) need to be integrated. SNI is a computational way of combining or fusing multimodal or multi-perspective data into a single dataset. For example, mRNA expression, miRNA expression, DNA methylation, image data, etc., need to be fused for a given set of patients to identify their similarity [37]. There are many methods to integrate the different perspectives of similarities, such as RWs [38], RWRs [39], similarity network fusion (SNF) [37], similarity kernel fusion (SKF) [40], heuristics approaches, etc.

A RW is a popular mathematical space model that is applied in various branches, such as computer science and mathematics. In this model, the walker can move to another position based on the probability distribution [38]. Therefore, the walker selects the node that has a higher transition probability to which it will walk. After sufficient iteration, a random path is obtained, which can be used as a feature for link prediction, recommendation, etc. Hence, the same process can be applied to generate a feature vector representation of each node in the network by selecting the most similar node in each step of the RW. The RW method determines the closeness among the nodes by calculating the proximity between nodes in the network, which can be utilized as a distance measure. Therefore, Qin et al. [41] developed a RADAR framework for learning the feature vector representation of diseases to determine their similarities by constructing a multi-layer disease similarity network. Then, a general RW algorithm is carried out on these networks to learn the vector representation of disease nodes and provide the score between diseases, where the lower the distance, the higher the similarity.

Besides, the RWR method is also used to identify the closeness between the nodes, with the only difference being that, in the RWR method, the walk can return to the start node (origin) by using a restart probability [39]. Similarly, Lee et al. [42] used protein-protein interaction, drug-drug interaction and known DTI data to construct the heterogeneous network. Then, the RWR algorithm was applied to each node to respectively get the weighted features of the drug and protein. Finally, these features were concatenated as drug-target pairs and given as an input to ML algorithms to identify the DTI pairs. However, the above methods select the neighboring similar node of the network randomly by checking the probability of first-hop neighbors; in addition, the restart probability is considered to be the same for all nodes in the network.

On the other hand, SNF [37] and SKF [40] methods are also applied to integrate the different similarity networks. SNF is carried out by using the edges that have higher scores in the network to update the final fused matrix of the network. Similarly, SKF [40] improves SNF by constructing two kernels, such as a normalized kernel and a sparse kernel. A sparse kernel removes the lesser similarities. After multiple iterations, the final integrated matrix is obtained. However, both methods lost information about the edges with weaker similarities, which may also contribute to the prediction of DTI. Further, some heuristic approaches only select the best similarity networks among all of the networks. The SNF-NN framework constructed by Jarada et al. [43] applies the SNF method to respectively fuse the different similarities of drugs and diseases by selecting the best similarities using a heuristic approach. Then, a highly tuned neural network model is applied to predict new drug-disease associations. Similarly, the DDR [16] and DTiGEMs+ [31] frameworks integrate multiple drug similarities and target similarities respectively by choosing the best similarities with the help of a heuristic algorithm to build a heterogeneous network for new DTI prediction. Here, both of the above frameworks choose limited perspectives of similarity networks alone for fusion, which can be a limitation of these frameworks.

In general, linear or nonlinear-based methods can be used to mine the features from the graphs or networks. Linear methods are represented as a straight-line path between two nodes of interest, whereas nonlinear methods are represented as subgraphs or meta-graphs in which the nodes of interest have more than one path and the edges are undirected. DDR [16] and DTiGEMs+ [31] frameworks use linear path structure-based feature extraction to determine the multiple path scores between the drug-target pairs. In this work, there are six different path structures of lengths 2 and 3. For each structure, the path score is calculated by using two different methods, such as the max and sum of the path. Totally, 24 dimensional features are extracted from two different graphs, G1 and G2. These extracted features are given as input to the ML algorithm, such as conditional random field and support vector machine classifiers to get the probability score of drug-target pairs.

Anand et al. [44] used topology-based feature representation to develop new drugs by using the properties of chemical structure. Here, the atoms and bonds are considered as nodes and edges, respectively. Similarly, Wang et al. [45] developed topology-based affinity score prediction between proteins by using the protein structures. Further, Fan et al. [46] and Fu et al. [47] proposed meta-path-guided neighbors to learn the representations of different types of nodes by using breadth-first search and depth-first search-based neighbors. The embeddings learned by different meta-paths for a node are fused to represent the node in a single feature vector representation with attention. In the HNEDTI [48] framework, a meta-path-based RW is carried out on a heterogeneous network to respectively represent the drug and target nodes by using low-dimensional embedding. On the other hand, Samizadeh and Minaei-Bidgoli [49] used a metapath2vec node embedding technique to obtain the feature vector representation of the nodes in the network. Later, these embedding vectors are fed as a concatenated feature vector into the binary classifier to identify new DTIs. Liu et al. [50,51] used hypergraph-based techniques to represent the proteins and drug molecules for prediction of the protein-ligand affinity and drug design, respectively. However, the above methods do not consider whether the nodes have more than one path between them. In addition, the meta-paths of different hop lengths used in these works [31,46–49] have not been given any weight or attention during the feature learning of nodes.

We have identified some limitations of the preceding works. The DSNs and TSNs are constructed by considering limited perspectives of the drugs and targets respectively. However, additional perspectives of similarity would have an impact on the performance of DTI prediction. The feature vector learned for each drug and target node in the DSN and TSN respectively by using the RW and RWR considers only the information about the local (immediate) neighbors to find the most appropriate similar nodes, whereas other (non-immediate) neighbors in the network may also contribute to DTI prediction. Integration of different similarity networks is carried out by selecting the best similarities based on the threshold; however, leaving out other similarities may affect the performance of DTI prediction. The feature vector representation of existing models (Metapath2vec, Meta-path guided GNN and so on) deals with nodes with multiple paths between them separately. However, it has not been represented as a meta-graph property to improve the prediction score and reliability of the DTI prediction. In addition, the node embedding learned by different meta-paths of varying length in the context of drug-target pairs is given equal attention, whereas different meta-path do not learn or generate the same feature vector representation. The above drawbacks affect the performance of DTI prediction.

To address these drawbacks, the following are our contributions.

- We utilize additional similarities of the drug and target respectively, such as pathway based similarity and protein family (Pfam) based similarity to improve the performance of the DTI

prediction.

- We propose an information entropy (IE)-based random walk (IERW) to learn the feature vector representation of the drug and target in each similarity network.
- We employ a multi-view convolutional neural network (MVCNN) in the context of SNI to consider the edges with a lower similarity score.
- We propose a novel node embedding technique to learn the features of each node by designing a meta-graph guided GNN.

3. Materials and methods

The details of the major components of the enhanced DTI prediction system are explained in this section. In order to improve the prediction performance, the proposed system addresses the drawbacks of the existing system described in Section 2. In general, the prediction of unknown associations between the entities is modeled as a link prediction problem. According to the GBA principle, similar targets may share similar drugs, and vice versa, as depicted in Figure 1.

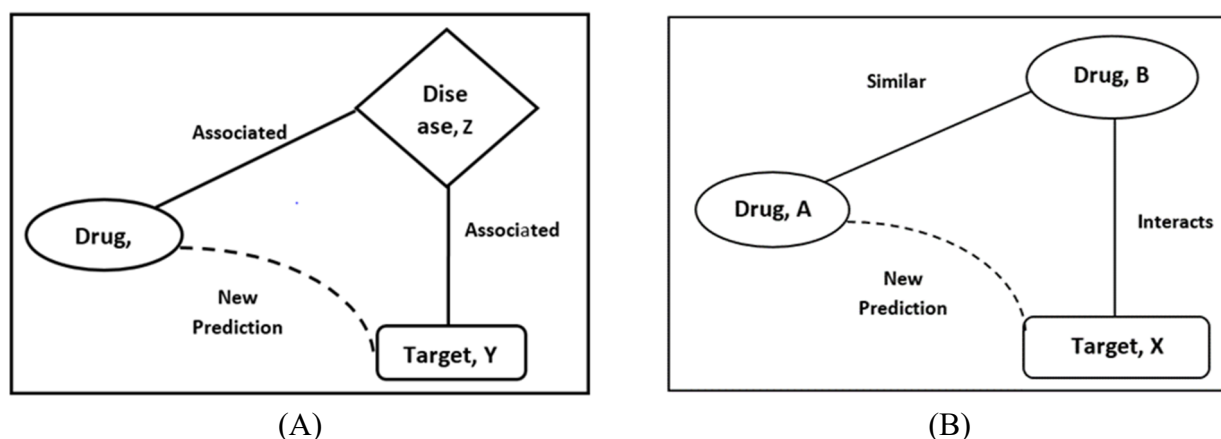


Figure 1. GBAs.

For example, Drug A is similar to Drug B, and Drug B interacts with Target X; then, there is a probability of Drug A interacting with Target X, as shown in Figure 1(A). Drug C is associated with Target Y, and Target Y is associated with Disease Z; then, there is a probability of Drug C interacting with Target Z, as shown in Figure 1(B). In our work, link prediction is carried out based on the GBA principle [52], where different information sources, different similarity measures and relations such as associations and interactions are used.

The major steps of the proposed architecture are the (i) processing of multiple similarity networks and SNI; (ii) construction of an HBN using an integrated DSN and integrated TSN, along with the other drug-related data and target-related data and known DTIs; (iii) meta-graph-based feature vector representation learning from the constructed HBN with imbalanced handling; and (iv) use of a CNN to predict the score of DTI pairs, as shown in Figure 2. A detailed description of each of the component is provided in the following sections.

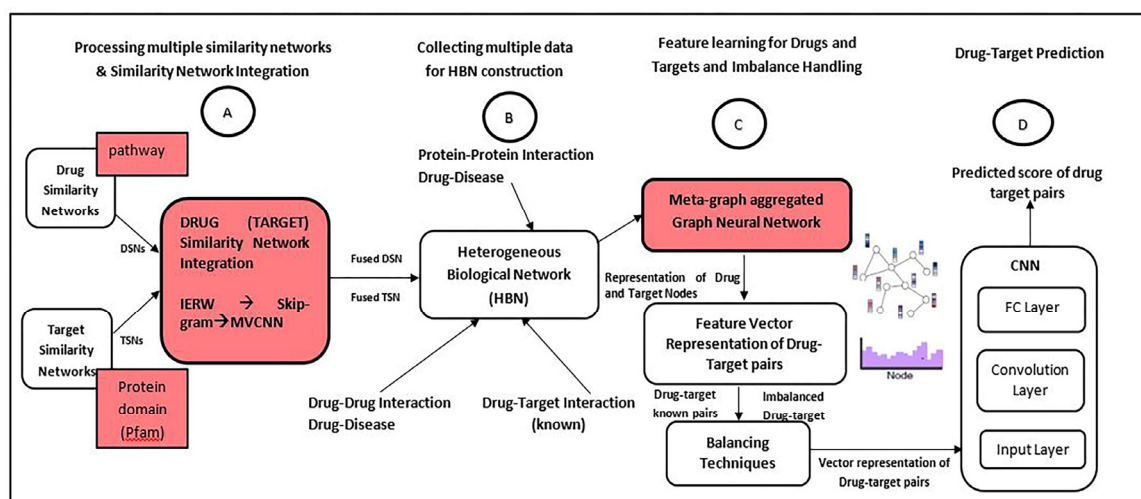


Figure 2. Architecture of DTiGNN system.

3.1. Processing of multiple similarity networks and SNI

In general, the similarity between biomedical entities can be carried out by using multiple views or perspectives. In some cases, each perspective may result in different scores or values among the entities, which will generate a biased conclusion. To overcome this issue, many researchers have shown that data integration techniques play an essential role in combining the data from different perspectives, thus improving the performance of finding similar objects in the field of biomedicine [53]. In the following section, the integration of multiple similarity measures of drug and target respectively is explained from different perspectives.

3.1.1. Similarity measures

There are numerous methods for determining similarities between drugs and targets. Each of them uses various properties of drugs and targets to strengthen or highlight the relationship between the entities (drugs or targets). In this work, based on the individual descriptors and interaction or association relations, 12 drug-drug similarity measures and 9 protein-protein similarity measures have been considered for the drug and target respectively. Different perspectives of the DSN and TSN are determined separately, which are explained in the following sections.

A. Drug similarity

In our work, we used the drug's individual descriptor measures, such as the chemical structure [54], ATC [55–56], target [57–58], disease [59–61], side effects [62], molecular function (MF) [63–65], cellular component (CC) [63–65] and biological process (BP) [63–65], as well as interaction relationship-based similarity measures such as DDI-based similarity [66,67], DTI-based similarity (Gaussian interaction profile kernel (GIP)) [55,68] and drug-disease association-based similarity [62,68] to determine the similarity between drug pairs. A pathway-based similarity [69] measure has been added as an additional similarity measure in our proposed work to improve the effect of DTI prediction. Figure 3 shows the different perspectives of drug similarity measures; the additional similarity measure is highlighted.

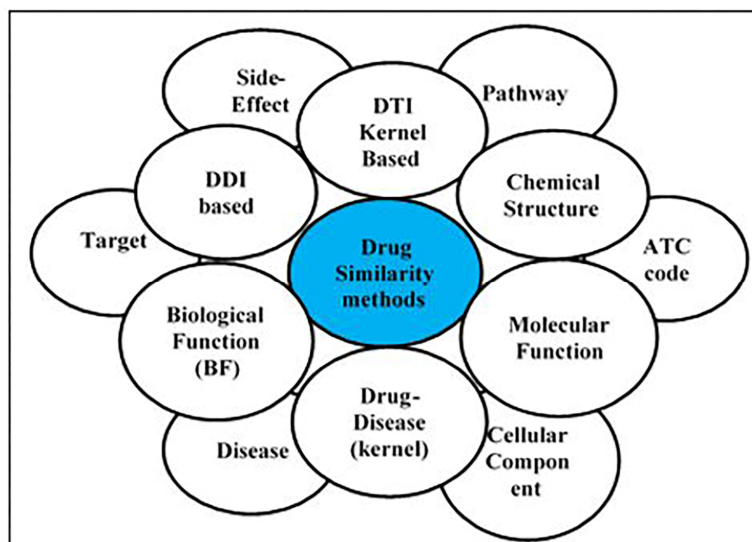


Figure 3. Similarity measures of drugs.

Pathway-based

The efficiency of the DTI prediction is increased by considering a pathway-based similarity measure as an additional similarity measure. The reason is that the therapeutic functions shared by similar drugs may not be binding to or interacting with the same target. However, the drugs may be interacting with a sequence of targets called pathways to treat the same disease [69]. Therefore, the drugs that are not similar via target-based similarity may be similar via pathway-based similarity [69]. Hence, the pathway-based similarity measure plays an important role in DTI prediction.

Information about drug pathways can be retrieved from the KEGG [70] database. When two drugs induce similar or overlapped pathways, they are said to be similar. To compute the similarity between drugs, the dice similarity score (DSC) between the pathways needs to be calculated, and this is given as $DSC(P_i, P_j) = \frac{2|P_i P_j|}{|P_i| + |P_j|}$, where P_i and P_j are pathways; it is expressed as sets of constituent genes.

The R package named BioCor [71] is used to find the DSC score by using the similarity between drugs that is determined by $S(d_i, d_j) = \max \{DSC(P_i, P_j)\}$.

B. Target similarity

To improve the accuracy of the DTI prediction, different perspectives on target or protein similarities have been considered. In our work, we used the protein individual descriptor measures, such as protein sequences (sequence alignment based) [60–61], structures (C_α atoms based RMSD) [72–74], MFs, CCs and BPs [64,75], and interaction relationship-based similarity measures, such as PPI-based similarity [76], DTI-based similarity (GIP) [55,68] and protein-disease association-based similarity [62,68], to determine the similarity between two proteins. In our work, to enhance the performance of DTI prediction, protein domain-based similarity [77–78] has been considered as an additional similarity measure. Figure 4 shows different perspectives of protein similarity measures in which the additional similarity measure is highlighted.

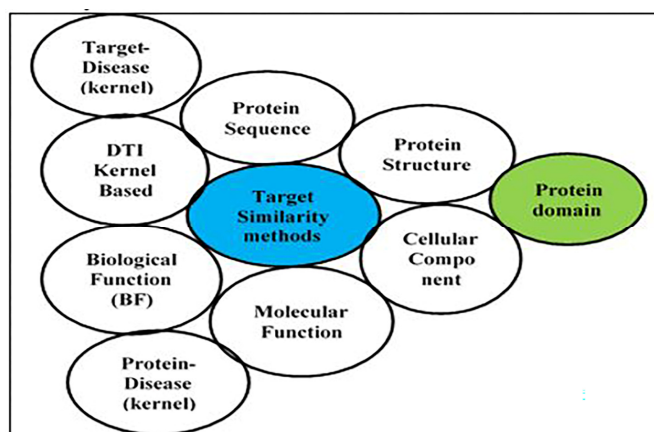


Figure 4. Similarity measures of proteins.

Protein domain similarity (Pfam)

The tools that are used to find the similarity based on the protein sequences follow a strict threshold value and miss certain matching hits in the sequences; thus, the accuracy of the similarity measure is degraded [77]. Therefore, the tool produces the result as “not similar,” but they are actually similar. In order to overcome this issue, protein domain co-occurrence is added as an additional feature to the pairwise sequence comparison tools. The proteins that are not similar via protein sequences may be similar via a protein family or domain-based similarity measure. Hence, the performance of the prediction can be improved.

Protein domain information is available in the Pfam database [78]. Here, each protein is expressed by a binary vector (domain fingerprint) in which the presence and absence of the protein domain are given by 1 and 0, respectively. The similarity score is measured by finding the Jaccard score between the pairs of domain fingerprint vectors of proteins p_i and p_j as follows:

$$PDSS(p_i, p_j) = \frac{FPV(p_i) \cap FPV(p_j)}{FPV(p_i) \cup FPV(p_j)},$$

where $FPV(.)$ is the fingerprint vector of the protein and $PDSS(p_i, p_j)$ is the protein domain-based similarity score between proteins p_i and p_j .

After constructing the DSNs and TSNs from multiple sources of information, our proposed work integrates all perspectives into a single similarity network for the drug and target respectively without missing any of the similarity networks. Hence, the accuracy of the DTI prediction system is improved.

3.1.2. SNI

SNI of the drugs and targets respectively aims to fuse the information available in multiple similarity networks into a single similarity network through which the performance of the DTI prediction system can be improved. The fused network represents shared and complementary knowledge across the networks. In this work, our goal is to consider all of the information available in all 12 drug-DSNs and nine protein-protein similarity networks, since some of the perspectives may contribute to some of the drug-target pairs during the prediction of DTI. One method to integrate the networks from multiple perspectives is the nonlinear similarity integration method called SNF [37]. In

DTiGEMs+ [31], the best subset of similarity networks is heuristically selected, and then SNF is applied for integration, which removes the noise and redundant information. In this method, stronger similarity edges are retained, whereas weaker similarity edges are removed. However, the edges that are less important also contribute to the prediction of DTI pairs.

Our aim is to utilize the weaker edges in each of the similarity networks, rather than choosing the stronger edges. In order to accommodate all of the edges of the nodes, we used a representation learning-based method to separately get the feature vector of each node in each of the similarity networks. To get the feature vector representation of each node, the walk sequences are generated by applying the RW in a multi-layer similarity network for each node [41]. However, the walker considers only the edge weight of immediate neighbors; the distant neighbors (non-local or non-immediate), which can also be semantically similar, have not been considered. Therefore, in our work, to select the best neighbor node to which to walk, a novel IE-based RW is carried out on each of the networks separately to generate the walk sequences for each node. The feature vector representation of the walk sequence is generated for all nodes by using the skip-gram (SG) model for each of the perspectives separately for the drug and target respectively. Here, each drug node consists of 12 feature vectors, and each protein node consists of 9 feature vectors. In order to get the single feature vector representation by utilizing all perspectives of the similarity network, several functions are used, like the average, geometric mean, min or max [46,47], etc. However, the importance or grade of the feature vector is not considered during the integration. In order to consider the feature vector representations of all perspectives without losing any information about their importance, a MVCNN model is used.

The following Figure 5 depicts the flow of our proposed SNI technique, which consists of the following three steps: (i) generation of walk sequences using an IERW; (ii) feature vector representation using the SG model; and (iii) fusion of the feature vector representation of each node using a MVCNN through which the prediction of DTI is carried out. The following sections will describe the above steps in a detailed manner.

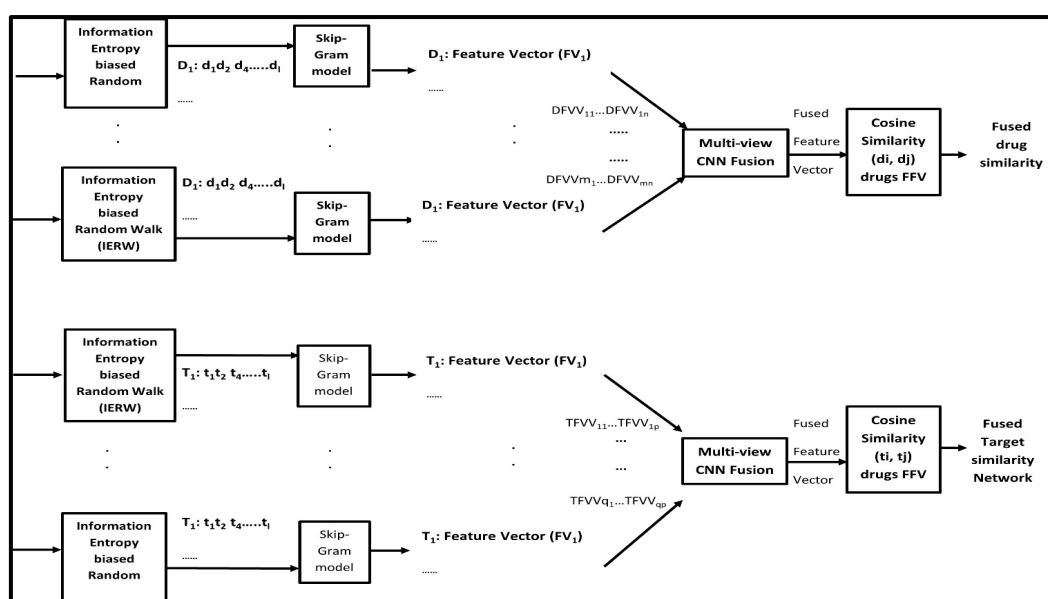


Figure 5. SNI.

A. Construction of feature vector representations

The feature vector representation of each entity (node) from each perspective is constructed by generating the walk sequences for each entity using our novel IE-based RW. Then, the generated sequences are fed into an SG neural network model. The feature vector of each node is generated from each perspective separately. Therefore, each drug node is associated with 12 feature vectors, and each protein (target) node is associated with 9 feature vectors.

(i) Generation of walk sequences

Generally, in the RW method, the walker moves to the next node in the network based on the transition probability score, which is calculated by using only the edge weight of immediate neighbors. Our novel approach selects the appropriate node to walk on the network by considering the neighbor nodes at up to two levels (immediate neighbors and non-immediate neighbors) to improve the accuracy of the walk sequences generated for each node. In addition, our IERW quantifies the uncertainty of the neighbors considered. Therefore, to improve the effectiveness of the random walker, our proposed work selects the best neighbor node to walk on the network for the current node based on the IE scores of both its immediate and non-immediate neighbors. IE is calculated based on metrics such as the edge weight, degree, node strength, betweenness, paths, etc. [79]. The IERW helps to improve the performance of DTI prediction. Algorithm 1 given below provides the steps for an IE-based RW to generate the walk sequences.

Algorithm 1: Generation of walk sequences using IERW

Input: DSNs (12 views), TSNs (9 views)

Output: walk sequences of similar nodes for each node in all views of the drug and target separately

1. *Begin*
2. *for all perspectives* $p \in PS$ *do*
3. *for all nodes* $i \in N$ *do*
4. *for* $l=0$ *to* *max walk-length* *do*
5. walk sequences of each entity ($walk_seq_ESN_{pi}$)= $\{\}$
6. *for each neighbor* (j) *of node* $i \in N_j$ *do*
7. calculate transition probability, $\pi_{i \rightarrow j} = \frac{A_{ij}(IE)^\beta}{\sum_{l \in N_i} A_{il}(IE)^\beta}$
8. calculate IE using the non-immediate neighbors of node i ,
 $IE = -\sum_{j \in N_i} P_j \log P_j$
9. calculate P_j -the strength of neighbor j of node i using non-immediate neighbors (k)
$$P_j = \frac{W_{jk}}{\sum_{m \in N_j} W_{jm}}$$
10. *best_neigh*= select the neighbor based on high transition probability score
11. append the *best_neigh* to $walk_seq_ESN_{pi}$
12. *end for*
13. *end for*
14. *end for*
15. *end for*
16. *End*

In the above algorithm, PS denotes whole perspectives, p denotes the current perspective, i denotes the current node, N indicates whole nodes, l denotes the walk length, $walk_seq_ESN_{pi}$ represents

walk sequences of each entity in the i^{th} perspective, j denotes the neighbor of current node i , N_j denotes all neighbors of node i and k denotes the non-immediate neighbor nodes.

To include the non-immediate neighbors (in level 2) in addition to immediate neighbors of a node during the relevant node selection for walk, the network attribute node strength [79] is used to calculate the probability distribution P_j , where j denotes neighbors of the current node i . P_j is calculated by using Eq (3.1) as given below.

$$P_j = \frac{w_{jk}}{\sum_{m \in N_j} w_{jm}} \quad 3.1$$

where w_{jk} is the edge weight between neighbor j of current node i and the neighbor of neighbor j of current node i . In the case of normal RW, neighbors of neighbors (non-immediate neighbors) are not considered. For example, considering Figure 6, IE is calculated for the immediate neighbor nodes D2, D3 and D4, which includes the probability distribution of the nodes in level 2, namely, D5, D6, D7, D8, D9 and D10. Hence, IE [80] is calculated as given in Eq (3.2).

$$IE = -\sum_{j \in N_i} P_j \log P_j \quad 3.2$$

Once the IE of the immediate neighbor nodes based on the probability distribution of non-immediate neighbors is determined, then the transition probability [80] needs to be determined to make a walk on the network from the current node, as given in Eq (3.3).

$$\pi_{i \rightarrow j} = \frac{A_{ij}(IE)^{\beta}}{\sum_{l \in N_i} A_{il}(IE)^{\beta}} \quad 3.3$$

where A_{ij} is the edge weight between the current node i and the direct neighbor node; the IE is calculated as given in Eq (3.2). Here, β is the tunable parameter, where $\beta > 0$ implies that the IE of the neighbor is low and the node strength is high, whereas $\beta < 0$ implies the opposite. Based on this transition probability, a suitable node is added to the walk sequence.

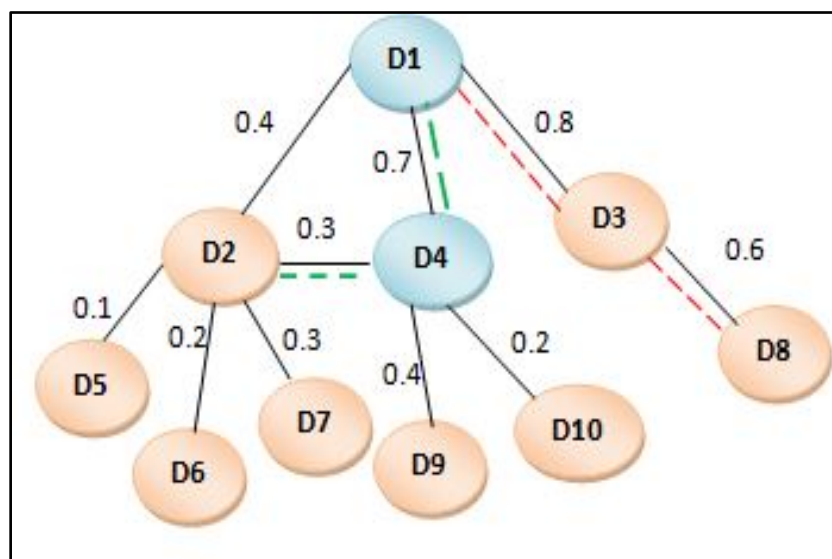


Figure 6. Sample walk on DSN.

Considering Figure 6 again, when a general RW is used, the walker chooses D3 as the next node rather than D2 and D4 because it has a higher transition probability; hence, the walk sequence of node D1 is D1, D3 and D8. Therefore, the context neighbors generated by using this RW algorithm missed the neighbor nodes D2 and D4. When we use our proposed method (IERW), the walker from node D1 selects node D4 by calculating the transition probability using the IE of drug D2, drug D3 and drug D4 based on Eqs (3.1) to (3.3). Similarly, from node D4, the walk is to node D2; hence, the walk sequence of D1 is D1, D4 and D2. This sequence considers the immediate context and IE to improve the accuracy of the prediction.

Once the IERW is carried out on each similarity network, the walk sequences generated based on the given walk length for each entity (node) in each similarity network are considered as sentences; hence, a corpus is obtained. The walk sequences generated by the IERW produce better feature vector representations for each of the perspectives of the similarity networks since the nodes of the walk sequence for each entity (node) in the network are generated with the help of the IE of the neighbors of the current node, which uses the non-immediate neighbors during the calculation of transition probability.

(ii) Generation of feature vector representation

Once the corpus is created, the next step is to use the SG model [81,82] to generate the feature vector representation of each entity (node) in the similarity network. The feature vector representation of each node is learned from the walk sequences generated by the IERW on each similarity network. Finally, each entity in each similarity network is represented as a feature vector in the appropriate dimension. To generate the feature vector representation of each entity (drug or target) in the similarity networks, any kind of representation learning algorithm can be adopted.

(B) Feature vector representation fusion

Once the feature vector representations of multiple perspectives of the drugs and targets respectively are constructed, they need to be integrated or fused into a single feature vector representation. To get the single feature vector representation, we examined a few methods, such as taking the element-wise minimum, element-wise maximum or element-wise average of all feature vectors, but they all showed lower performance in DTI prediction. This is because the grade or importance of the feature vector representation of a node in each perspective is not the same and some of the feature vectors from the perspectives are missed. Therefore, to utilize the vector representation of all perspectives, an MVCNN is used in our work, which was proposed by Li et al. [83] to recognize 3D objects. In their work, the authors fused the images from multiple views into a single image by applying a CNN architecture [83]. We were inspired by this work and decided to use an MVCNN to get fused feature vector representations of a node available in multiple perspectives or views in a novel manner for the drug and target respectively.

To reduce the computational time of the MVCNN algorithm, in our work, the modified version of the architecture is used by applying a single-CNN architecture in which the feature vector representation of all perspectives is utilized. Sum-view pooling is used in the last layer by replacing the max-view pooling to get the fused feature vector representation of each node. Figure 7 illustrates the modified architecture of the MVCNN in the context of feature vector representation fusion to get the single DSN and single TSN.

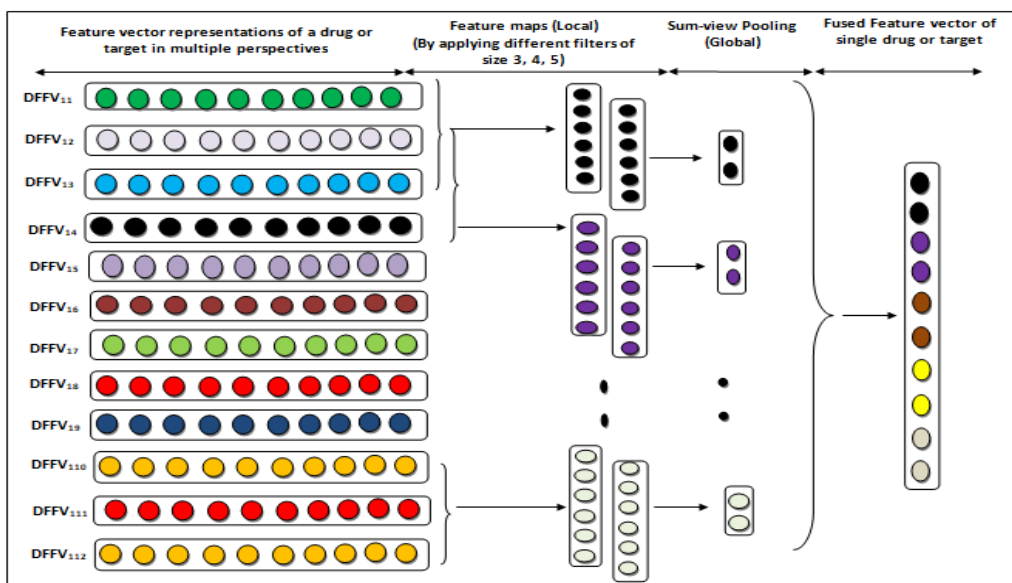


Figure 7. Architecture of MVCNN.

The input to the MVCNN model is the feature vectors of a drug or target in multiple views, and this is denoted as FVV_i . FVV_i is represented as the feature vector representation of $entity_i$, and it is determined by merging all perspectives of $entity_i \in R^{d \times m}$, where m is the number of views or perspectives. FVV_i is expressed as

$$FVV_i = \begin{bmatrix} FVV_{i,j,1,\dots} & \dots & FVV_{i,j,d} \\ \vdots & \ddots & \vdots \\ FVV_{i,j+1,1,\dots} & \dots & FVV_{i,m,d} \end{bmatrix} \quad 3.4$$

where $FVV_{i,j}$ is the feature vector of $entity_i$ in the j th view or perspective in dimension d (R^d).

To get the convolved feature maps, different windows of size 3, 4 and 5 are applied with various sizes of filters. A filter $W \in R^{v \times d}$ is applied to perform the convolutional operation on a window of v views to generate the new feature. For example, the new convolved feature maps c_i generated by the window of v views $FVV_{i,j:i,j+v-1}$ (from the i th entity and j th view to the i th entity and $j+v-1$ views) for the $entity_i$ is denoted by

$$C_i = f(W \cdot FVV_{i,j:i,j+v-1} + b) \quad 3.5$$

where W is the filter, f is the activation function and b is the bias term. The feature map c is obtained by applying this filter to each window of views in the feature matrix $\{FVV_{i,1:i,v}, FVV_{i,2:i,v+1}, \dots, FVV_{i,m-v+1:i,m}\}$:

$$C = [C_1, C_2, C_3 \dots \dots C_{m-v+1}] \quad 3.6$$

where $C \in R^{m-v+1}$. The obtained feature maps are called the local feature maps, i.e., the integration of a fixed number of perspectives based on window size v . In order to integrate all perspectives, or to capture the global features, sum-view pooling is applied to the convolved feature maps c_i . The sum pooling is calculated as follows:

$$C_i(\text{sum}) = \text{sum}(C) \quad 3.7$$

where C_i represents the feature vector of the particular filter i . An alternative method to this pooling is an element-wise max operation, which is not efficient for DTI prediction. Finally, the integrated feature vector of the particular $entity_i$ is the concatenation of the sum value of the different filters. The final integrated feature of the $entity_i$ is represented by

$$entity_i = [C_1(sum) \oplus C_2(sum) \oplus \dots C_k(sum)] \quad 3.8$$

where k is the number of filters applied. The feature vector representation of each entity fused by MVCNN achieves better improvement in DTI prediction because the fused vector contains the information about all perspectives without losing any information about any perspective. Once the fused feature vector representation is obtained, the next step is to calculate the similarity score. The similarity score between two drugs is calculated by using Eq (3.9) as given below.

$$\text{cosine similarity}(Drug_A, Drug_B) = \frac{Drug_A \cdot Drug_B}{\|Drug_A\| \cdot \|Drug_B\|} \quad 3.9$$

Likewise, the similarity score between target pairs is calculated. Finally, the fused single DSN and fused single TSN is constructed by using the calculated score.

3.2. Heterogeneous biological network construction

In general, an HBN is better suited to expressing the complex relationships between entities. Nowadays, HBNs are used in many data mining applications, like classification, recommendation, similarity search, clustering and prediction. To predict the novel DTI pairs, the HBN is constructed with the help of known DTI pairs, drug-DSNs and protein-protein similarity networks. In our proposed work, to increase the number of new DTI predictions, in addition to improving the strength of the interaction, the number of edges in the network between drugs and targets is increased. Therefore, our HBN consists of three types of nodes (drugs, proteins and diseases) and seven types of association among the nodes. The seven types of associations are fused drug-drug similarities, DDIs, drug-disease associations (DDiA), fused protein-protein similarities, protein-protein interactions, protein-disease associations and known drug-protein interactions. Figure 8 shows a sample HBN with different types of nodes and edges.

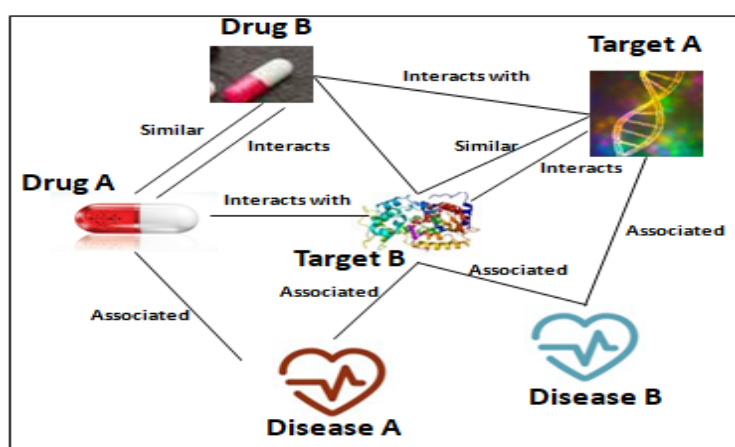


Figure 8. HBN.

Once the HBN is constructed, the next step is to mine the knowledge present in the HBN using the meta-graph. In our work, a meta-graph in the context of a drug-target pair with a hop length equal to 2 and 3 is considered based on the drug, target or disease entity nodes. These meta-graphs are associated with three different semantic relations, i.e., similarity, interaction or association, and the same is shown in Figure 9.

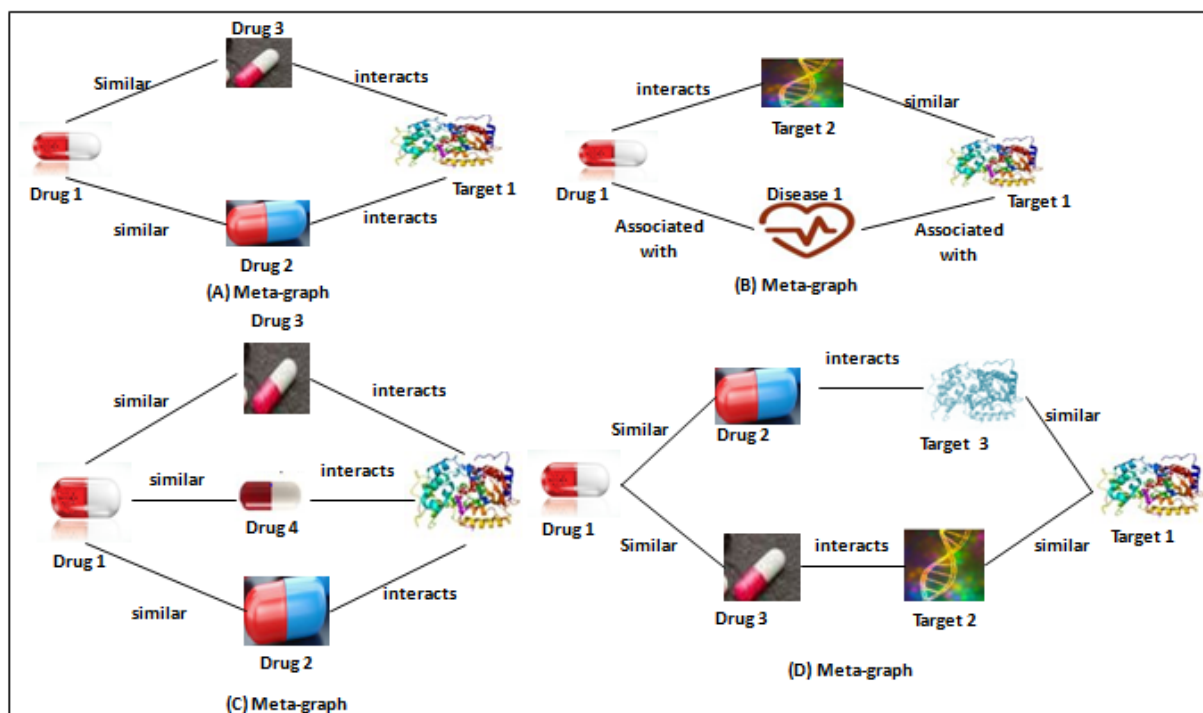


Figure 9. Multiple meta-graphs showing paths between the same entities.

3.3. Meta-graph-based feature vector representation learning using a GNN

In general, the performance of the link prediction task heavily depends on the feature vector or embeddings learned from the representation learning model. One of the primary motivations for our work is to improve the representation learning of drug and target nodes that participate in the network by capturing structural and semantic information. A DL framework can be used to represent each node of the HBN in a vector representation with a lower dimension compared to the number of nodes in the HBN. DL-based representation learning performs better [84] than previously used RW-based methods like node2vec [85], metapath2vec [86], etc. The node embedding learned by the node2vec model achieves better performance for homogeneous networks [66]. However, it is not able to differentiate the types of nodes while learning the features.

On the other hand, the metapath2vec model is designed to learn the node embedding of a heterogeneous network in which the node's type is denoted by a meta-path. However, the meta-paths with different semantics, such as similarity, interaction or association (Figure 10(A)), as well as the meta-paths of varying hop length (Figure 10(B)), which convey different types of information, have been given equal importance. That is, the embedding of a drug or target node learned or generated by Meta-path 1 and Meta-path 2, though with different semantics, is considered equally important.

Similarly, the embedding learned by the meta-path of length 2 and the meta-path of length 3 (Figure 10(B)) is considered in an equal manner. In addition, the nodes with more than one edge among them have been considered separate linear semantic relation paths [16,31,48]. The prediction of DTI that is identified by multiple semantic relation paths between the same drug-target pairs, on the other hand, is more reliable than a single semantic relation path. In this work, a meta-graph is defined as the subgraph where the same source and target nodes (entities) are connected by more than one semantic linear path with different entities, semantics or hop lengths, as shown in Figure 10(C). Here, we defined the meta-graphs in such a way that each meta-graph has a single hop length. Therefore, in this work, in order to differentiate the embedding learned by various semantics, various lengths and multiple paths of the same meta-graph, a novel meta-graph guided representation learning model that, uses intra-meta-graph fusion, along with an attention mechanism, during the learning of the model to improve the performance of the DTI prediction.

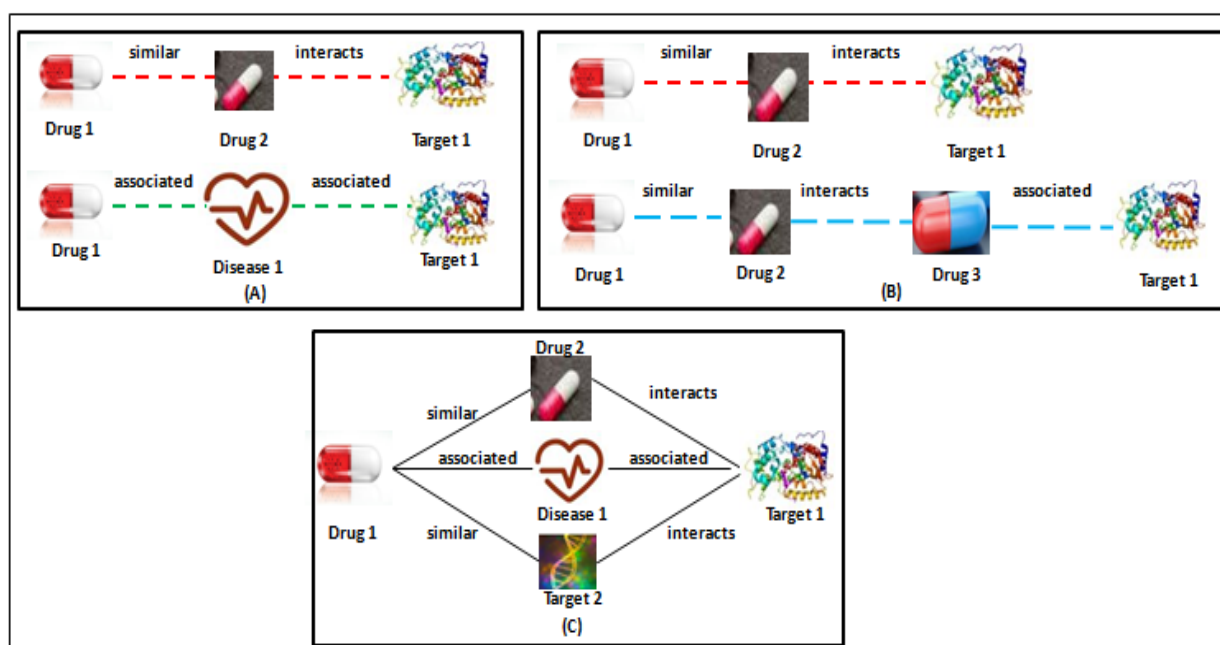


Figure 10. Meta-paths with different semantics, hop lengths and multiple paths.

Furthermore, as illustrated in Figure 9, multiple independent meta-graphs can define relation paths between the same entities. The novel meta-graph guided representation learning model discussed above tackles multiple meta-graph embeddings corresponding to different meta-graphs by using inter-meta-graph fusion by giving different attention to different meta-graphs to improve the strength of the association and the accuracy of DTI prediction. Unfortunately, when the meta-path guided model learns the embedding of a drug or target node, the relation information conveyed by the original source and destination of the DTI pair is lost. In order to overcome this problem, a vector enhancement layer has been added to the model to improve the feature vector representation, or embedding, of the drug-target pairs.

Our proposed representation learning model is shown in Figure 11, and it consists of four components: (1) node base embedding, (2) intra-meta-graph fusion, (3) inter-meta-graph fusion and (4) vector

enhancement. The sections that follow will detail the operation of each layer in our meta-graph guided GNN model for representational learning.

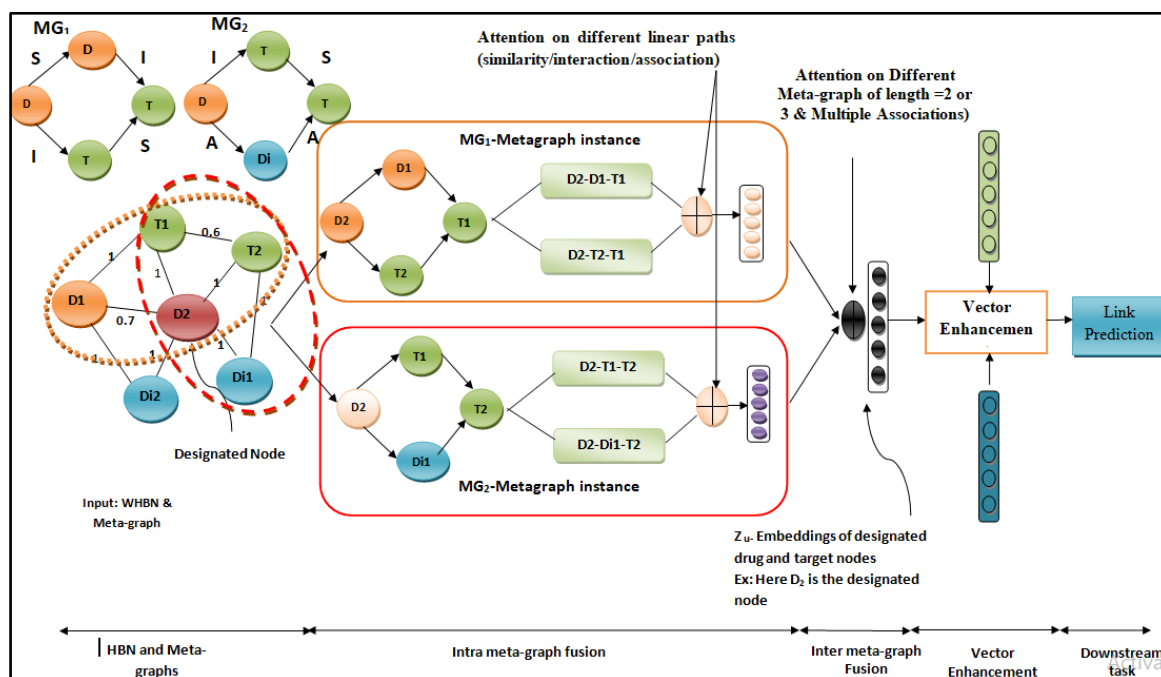


Figure 11. Architecture of meta-graph guided GNN for node embedding.

3.3.1. Node base embedding

In a GNN model, the node base embedding process is the initial step in which each node is represented as embedding vectors of equal dimension in order to avoid the dimension problem during the aggregation of neighbor information. This is adopted from MHN [46,47], but the entity nodes (drug, target and disease) and their corresponding features are different. To improve node base embedding in any large network, the node ID and attributes of the nodes have been considered. First, the embedding vector of the node is generated by the node ID. For example, the metapath2vec [78] and Deepwalk [87] models use node IDs to learn the embeddings of the node. Hence, the same mechanism is applied to get the embedding of the node, which is given by Eq (3.10):

$$h_v^{id} = W_e \cdot v \quad 3.10$$

where W_e is the weighted parametric matrix which is updated during training, v is the node ID and h_v^{id} is the latent vector generated for the given node ID in the corresponding layer. Second, the embedding vector of the node is constructed based on its attributes, just as any HBN is represented by the node's attributes. In our work, for the drug node, the ATC code, structure, target, disease, side effects, etc., have been considered. Likewise, the other types of nodes, such as those for proteins, are represented by their sequence, structure, biological functions, etc., and disease is represented by its own attributes. However, the dimension of the feature or attribute is not the same for all types of nodes. In order to map the feature vectors of different types of nodes, $v \in V_A$, obtained from the MVCNN into a common vector space based on attributes, the following function is used, as given by Eq (3.11):

$$h_v^{atrs} = W_A \cdot x_v \quad 3.11$$

where x_v denotes the feature vector representation of node v , W_A is the weighted parametric matrix and h_v^{atrs} is the transformed or projected latent vector of node v . To increase the performance of the DTI prediction, the feature vector representation of the node based on node ID and node attributes is combined into a single feature vector h_v by averaging these vectors, as given below.

$$h_v = \text{averagepooling}(h_v^{id}, h_v^{atrs}) \quad 3.12$$

Now, every node in the HBN is represented by a feature or latent vector in the same dimension. The vectors are then used to aggregate the meta-graph guided neighbors in the subsequent layers, as explained in the following section.

3.3.2. Intra-meta-graph fusion

To learn (i) the semantic and structural information buried in the designated node (drugs or targets) for the given meta-graph, (ii) the meta-graph-based neighbors and (iii) the context between the given source and destination nodes in the meta-graph, an intra-meta-graph fusion layer is used. Intra-meta-graph fusion is the aggregation of the information available in multiple instances of the same meta-graph with attention [88]. Take a look at Figure 11 to see the HBN and meta-graph #1. For example, assume node D_2 is the designated node to learn the embedding vector based on the given meta-graph #1 in the context of a drug-target pair. Meta-graph #1 has one instance in the HBN (Figure 11), and it is associated with both the similarity and interaction semantic relation paths.

In general, each meta-graph can have more than one instance for a designated node v (in our work, v can either be a drug or a target since our focus is on DTI prediction), where each instance can have different importance. However, the instances corresponding to a single meta-graph have the same hop length. In addition, in our work, each instance of the meta-graph can have more than one semantic path, each with a different importance associated with it. In this work, this multi-instance, multi-semantic view corresponding to a designated node in a single meta-graph is handled by intra-meta-graph fusion. A latent vector is first described for each semantic path, and then the importance of each semantic path is determined through the attention mechanism from which the aggregated single latent vector of each instance of the meta-graph is obtained. Then, the importance of each instance of the designated node v is determined through the second attention mechanism, through which the aggregated latent vector corresponding to the node v for a single meta-graph is obtained.

The neighbors of each semantic path are encoded into a latent vector by using a nonlinear sigmoid function, as given in Eq (3.13).

$$h_{v,SP_i}^{MG_j} = \sigma(W \cdot [h_v \ v \in N_v^{SP_i}]) \quad 3.13$$

where $W = w_1$ for similarity (drug or target) or $W = w_2$ for interaction (drug or target) or $W = w_3$ for association (disease). Here, W is denoted as the weight metric, which is learned during training, SP_i is denoted as semantic path i and $h_{v,SP_i}^{MG_j}$ is the latent vector of semantic path i in a meta-graph instance j (MG_j for the designated node v with the dimension d'). Once the semantic paths of a given meta-graph are encoded as embedding vectors, an attention mechanism is used to learn the importance of each semantic path. The basic idea of this approach is that the different paths or semantics would contribute

differently to the designated node's vector representation. The importance of each semantic path e_{SP_i} is determined by using Eq (3.14).

$$e_{SP_i} = q^T \cdot h_{v,SP_i}^{MG_j} \quad 3.14$$

where q^T is denoted as the weight metric, and it is used to decide the importance of similarity, interaction or association semantic relations associated in each semantic path. Then, the normalized attention weight α_{SP_i} is determined by using the softmax function as given in Eq (3.15)

$$\alpha_{SP_i} = \frac{\exp(e_{SP_i})}{\sum_{SP_i \in SP} \exp(e_{SP_i})} \quad 3.15$$

where α_{SP_i} is the normalized attention weight of semantic path i in meta-graph instance j (MG_j). The combined latent vectors of all semantic paths of a meta-graph instance j with associated attention are determined by using Eq (3.16).

$$h_v^{MG_j} = \sum_{SP_i \in SP} \alpha_{SP_i} \cdot h_{v,SP_i}^{MG_j} \quad 3.16$$

The feature vector corresponding to one instance of the meta-graph for designated node v is given by Eq (3.16). However, there may be more than one instance for node v . Therefore, the final feature vector of designated node v , considering all of the meta-graph instances, is represented as given in Eq (3.17).

$$h_v^{MG_k} \in R^{d'} \forall v \in V \quad 3.17$$

Once the instances of a given meta-graph is encoded as an embedding vector representation, to learn the appropriate attention or weightage of the instances, we utilize the graph attention layer [78]. The basic idea of this approach is to tackle the different instances that contribute differently to the designated node's vector representation. The contribution or influence of each meta-graph instance is determined by

$$e_v^{MG_j} = a_{MG}^T \cdot h_v^{MG_j} \quad 3.18$$

where $a_{MG} \in R^{2d'}$ denotes the parameterized attention vector for meta-graph instance MG_j . Additionally, $e_v^{MG_j}$ is the attention or importance of each meta-graph instance MG_j for the designated node v , and it is normalized by using the softmax function across all meta-graph instances $MG_i \in M$ to determine the normalized attention weight $\beta_v^{MG_j}$ as given in Eq (3.19).

$$\beta_v^{MG_j} = \frac{\exp(e_v^{MG_j})}{\sum_{MG_i \in MG} \exp(e_{v_s}^{MG_j})} \quad 3.19$$

Once the normalized attention weights $\beta_v^{MG_j}$ are determined for all instances $MG_j \in MG_k$, then the vector representation of all instances of single meta-graph m for the designated node v is denoted by $h_v^{MG_m}$ and computed by passing output into the sigmoid function, as follows:

$$h_v^{MG_m} = \sigma \left(\sum_{MG_j \in MG_m} \beta_v^{MG_j} \cdot h_v^{MG_j} \right) \quad 3.20$$

In order to sustain the learning process, the attention mechanism is enhanced to multiple heads to

avoid variance occurring due to heterogeneity of the networks. Therefore, an attention mechanism is applied K times autonomously and the results are concatenated by using Eq (3.21) as given below.

$$h_v^{MG_m} = \parallel_{k=1}^K \sigma(\sum_{MG_j \in MG_m} [\beta_v^{MG_j}]_k \cdot h_v^{MG_j}) \quad 3.21$$

where $[\alpha_v^{MG_j}]_k$ indicates the normalized attention of the meta-graph instance $MG(v,u)$ to the designated node v at the k th head.

The feature representation of ‘g’ meta-graph instances for the designated node v is given as $\{h_v^{MG_{m_1}}, h_v^{MG_{m_2}}, \dots, h_v^{MG_{m_g}}\}$. Here, $h_v^{MG_m} \in R^{d'}$ illustrates that the node v encompasses the information from every instance in each aspect of semantics in the context of drug-target pair. Once the latent vector of each meta-graph is learned, the vector does not hold the information about the hop length (hl), semantics (st) or semantic path count (sc) associated with that meta-graph. Therefore, the above information about each meta-graph is appended into the latent vector of the corresponding meta-graph, which is given as $\{h_v^{MG_{m_1}}[hl_{m_1}, st_{m_1}, sc_{m_1}], h_v^{MG_{m_2}}[hl_{m_2}, st_{m_2}, sc_{m_2}], \dots, h_v^{MG_{m_g}}[hl_{m_g}, st_{m_g}, sc_{m_g}]\}$.

3.3.3. Inter-meta-graph fusion

Once the aggregation of the information present in each meta-graph is carried out by using intra-meta-graph fusion, the next step is to aggregate the multi-instance, multi-semantic information available across all meta-graphs for the designated node v by using an inter-meta-graph fusion layer. According to our proposed work, every node (drug, target), has different kinds of latent vectors for each meta-graph: $\{h_v^{MG_1}, h_v^{MG_2}, h_v^{MG_3}, \dots, h_v^{MG_M}\}$ for the node $v \in V$, where M is the number of meta-graphs. According to the sample meta-graphs shown in Figure 11, the node D_2 has two different latent vector embeddings based on the given two meta-graphs. Meta-graph #1 is defined based on the similarity and interaction semantics with hop length 2, and Meta-graph #2 is defined based on the similarity, interaction and association semantics with hop length 2. Similarly, 15 meta-graphs were considered in the context of a drug-target pair, and each drug and target node will learn 15 different embedding vectors corresponding to each meta-graph. According to our findings, meta-graphs with multiple different semantic paths with hop length 2 require more attention than meta-graphs with multiple same semantic paths with hop length 2 or 3.

In general, the designated node v can be associated with multiple meta-graphs. Each meta-graph is considered from the point of view of the graph rather than as an instance. And, no meta-graphs are not identical to each other. They may have different hop lengths, different or the same semantic paths and multiple semantic paths. To get the single embedding vector for the designated node v , we examined a few methods, such as taking element-wise minimum, maximum or average value among all of the embedding vectors [39, 40]. However, the vector learned by the designated node v based on different meta-graphs during the intra-meta-graph fusion layer is not the same, as the information is different among the meta-graphs in terms of semantics, hop length and multiple paths in the meta-graph. Therefore, to fuse the embedding vectors of each node, our proposed work uses an attention or weight mechanism on the latent vectors of each meta-graph. The fusion or merging of different latent vectors of different meta-graphs with attention is determined as follows:

$$e_v^{MG_i} = W^T \cdot h_v^{MG_i} [hl_i, st_i, sc_i] \quad 3.22$$

where W^T is the learnable parameter based on the hop length, semantics and semantic path count. We assigned larger initial weights for the meta-graph which has a lesser hop length (e.g., 2) and more semantic paths (three paths) based on different semantics (drug, target, disease). Similarly, the meta-graph is given less weight because it has more hops (e.g., three), fewer semantic paths (two paths) and the same semantics (drug, target or disease). As a result, each meta-graph's weight or attention is denoted as γ_{MG_i} , which is determined by using Eq (3.23) given below.

$$\gamma_{MG_i} = \frac{\exp(\exp(e_v^{MG_i}))}{\sum_{\substack{MG_i \in MG, \\ sc_i \in SC, \\ st_i \in ST, \\ hl_i \in HL}} \exp(e_v^{MG_i})} \quad 3.23$$

Then, the fused embedding vector h_v^{MG} of all of the considered meta-graphs of each node is determined by using the sigmoid activation function as given by Eq (3.24).

$$h_v^{MG} = \sigma(\sum_{MG_i \in MG} \gamma_{MG_i} \cdot h_v^{MG_i}) \quad 3.24$$

where γ_{MG_i} is the attention weight assigned to each meta-graph for the designated node v and h_v^{MG} is the aggregated latent vector of multiple meta-graphs of the designated node v .

The final step is to show the node embedding vector in vector space in the desirable output dimension; an additional transformation step is needed, and it is carried out by using the nonlinear sigmoid activation function:

$$Z_v = \sigma(W_o * h_v^{MG}) \quad 3.25$$

where W_o is the weight matrix at the output layer. Z_v is the embedding vector of each drug or target node in the vector space in a desirable dimension through which any kind of downstream task like node classification, clustering, link prediction, etc., can be carried out.

3.3.4. Vector enhancement layer

Each node in the inter-meta-graph fusion layer is learning the vector embedding in the context of a drug-target pair with varying levels of attention. Though the model learns and generates the embedding vector in the context of drug-target pairs for each drug or target node in a better way, the specific drug-target pair information that exists for that particular drug or target node on the HBN is missed. Hence, the embedding vector learned in the previous layer needs to be enhanced. To enhance the embedding, one more layer, called the vector enhancement layer, has been added, in which the specific drug-target information is utilized in the corresponding embeddings of the drug or target node to improve the performance of the DTI prediction of that particular pair. Accordingly, the embedding vector of the drug or target is enhanced by Eq (3.26) or (3.27), respectively.

$$Z_D = \{\sum_{p \in P} h_V^D \odot h_V^T\} \oplus Z_V^D \quad 3.26$$

$$Z_T = \{\sum_{p \in P} h_V^T \odot h_V^D\} \oplus Z_V^T \quad 3.27$$

where Z_D or Z_T are the enhanced embedding vectors of the drug (D) or target (T), P denotes the number of drug-target pairs belonging to that particular drug or target node, \oplus indicates element-wise addition operation with Z_V^D or Z_V^T , which is the embedding vector of the drug or target that is learned in the inter-meta-graph fusion layer and sum of the product of drug-target pairs belonging to that particular D or T .

Each drug node and target node in the HBN learned the enhanced embedding vector by using multiple meta-graphs and its instances. The performance of the proposed meta-graph guided representation learning model is discussed later in this paper.

3.4. DTI prediction using a CNN

To predict the unknown DTI pairs, the feature vector representation of each drug and target node learned using a meta-graph guided GNN model is concatenated as drug-target pairs via element-wise multiplication. The ratio of known to unknown DTI pairs is 1:69. To tackle the balanced drug-target pairs, oversampling techniques such as SMOTE [89], k-means SMOTE [89] or a generative adversarial network (GAN) [90] is carried out on the minority class samples of the training data before being fed into any downstream task model. A CNN model is used [91] to predict the probability score of drug-target pairs. Most biomedical applications use a CNN model to detect DNA methylation [92] and multi-label protein lysine PTM sites [93]. In our work, a CNN is applied as the supervised training model. The architecture of the CNN is composed of an input layer, a convolutional layer, a max pooling layer, a fully connected layer (FCL) and an output layer. The length of the input vector is l , and the weight matrix size is $4 \times l$. In this work, we used four kernels, or filters. The convolutional layer learns only the local features from the input vector. To learn the global features and reduce the dimensionality of the feature map, a max pooling layer is used. In the max pooling layer, the filter moves by a certain stride size (here, $s = 2$). The output vector extracted from all of the kernels is represented as a one-dimensional vector. This vector is given as an input to the FCL, which uses the sigmoid function to map the features into the output values (probability scores) that lie between the ranges of 0 and 1. These probability scores interpret the strength of the interaction between the drug-target pairs. When the drug's strength or probability of interaction is equal to or greater than 0.5, the drug is interacting with the target. The following section evaluates and analyzes the results of each contribution described in Section 3 to ensure the performance of our proposed system.

4. Results

4.1. Datasets

To validate the performance of our proposed DTiGNN framework, we used two types of datasets: categorical and non-categorical. Yamanishi_08 [94] is one of the popular benchmark datasets used in [31,35], and it comes under the category of a categorical dataset, as shown in Table 1. Kuang et al. [95], Luo et al. [15] and a modified Luo method with the inclusion of additional edges are categorized under the non-categorical dataset, as shown in Table 2. Here, the Yamanishi_08 dataset divides the drugs and targets into four categories, i.e., enzymes, ICs, G-protein channel receptors (GPCRs) and nuclear receptors (NRs).

In our proposed work, similarity and interaction edges are treated in a separate manner. The reason for

adding additional edges is to improve the strength of the DTI, as well as the number of DTI predictions.

Table 1. Numbers of drugs, targets and interactions for four gold standard datasets (categorical).

Dataset		Enzyme	IC	GPCR	NR	TOTAL
Yamanishi_08 [94]	Drugs	445	210	223	54	932
	Targets	664	204	95	26	989
	DTI (known)	2926	1476	635	90	5127
	DTI (Unknown)	292554	41364	20550	1314	355782
	Ratio	0.010	0.036	0.031	0.068	0.014

Table 2. Numbers of drugs, targets and interactions for four gold standard datasets (non-categorical).

Dataset	Nodes	Total
Kuang [95]	Drugs	809
	Targets	786
	DTI (known)	3681
Luo [15]	Drugs	708
	Targets	1512
	DTI (known)	1923
	Disease	5603
	Edges	1,895,445
Modified Luo	Drugs	708
	Targets	1512
	DTI (known)	1923
	Disease	5603
	Edges (DDI, PPI)	3,545,374

4.2. Evaluation metrics

Herein, to verify the robustness and performance of the DTiGNN system, we adopted a 10-fold cross-validation method for the prepared dataset. The performance is analyzed by calculating the area under receiver operating characteristic curve (AUC) [96] and the area under the precision-recall curve (AUPR) [96] for the test data. The AUC is calculated by drawing the curves between the true positive rate (TPR) and false positive rate (FPR) at different threshold levels. Similarly, the AUPR curve is constructed between different precision and recall values through which the performance of the system is validated.

$$Precision = \frac{TruePositive}{TruePositive+FalsePositive} \quad 4.1$$

$$\text{Recall or TPR or Sensitivity} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \quad 4.2$$

$$F - \text{Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad 4.3$$

$$FPR = \frac{\text{FalsePositive}}{\text{FalsePositive} + \text{FalseNegative}} \quad 4.4$$

$$\text{Specificity} = \frac{\text{TrueNegative}}{\text{TrueNegative} + \text{FalsePositive}} \quad 4.5$$

In the following sections, our proposed system is compared with other state-of-the-art methods to analyze the performance of the DTI prediction. The newly predicted DTIs are validated by using docking analysis and several databases. The docking analysis is done with the help of the AutoDoc Vina [97] tool. The databases include DrugBank [55], CTD [62], PubMed [98], ChEMBL [99], etc., for validation. The distinctive characteristics of our proposed DTiGNN framework to boost or improve the accuracy of the DTI prediction are also highlighted by comparing it with other methods.

4.3. Discussion

The following section deals with the results and discussion of each contribution of the system, as well as performance comparisons with state-of-the-art methods. To validate the performance of the DTiGNN system, it is examined on various datasets (Yamanishi_08 [94], Kuang [95], Luo [15] and modified Luo) by using the following criteria: analysis of the impact of additional similarity perspectives for the drug and target, respectively, for DTI prediction; the impact of the novel IERW for link prediction; the adaptation of the MVCNN for SNI for DTI prediction; the impact of novel meta-graph-based representation learning; the impact of the HBN; analysis of the performance with state-of-the-art methods; statistical test analysis; and evidence analysis for the new DTI pairs with docking analysis.

4.3.1. Impact of different drug similarity perspectives on DTI prediction targets

Table 3 shows the impact of each similarity perspective of the drug and target in DTI prediction for the dataset described in Section 4.1. In addition to the various similarity measures that are described in Section 3.2-A, the pathway-based similarity of drugs and Pfam-based similarity of proteins are included to improve DTI prediction.

Table 3 shows that the addition of each similarity perspective at a time showed better performance for the drugs and the targets on all of the considered datasets. Specifically, the pathway-based similarity improved the score of AUC by 2, 4, 2 and 2% for Yamanishi_08, Kuang, Luo, and the modified Luo dataset, respectively. Similarly, Pfam-based similarity increased the value of the AUC by 3, 2, 3 and 1% for the Yamanishi_08, Kuang, Luo and modified Luo datasets, respectively. At times, drugs that do not have common targets are found to be similar via pathway-based similarity. Similarly, Pfam reduced the false negatives and false positives that occurred due to protein sequence-based similarities. Due to this, the sensitivity and specificity scores of the DTI prediction are also improved by 3 and 4%, respectively. When the sensitivity score increases, the AUC score also increases accordingly.

Table 3. Comparison of different similarity perspectives of the drug or target for DTI prediction.

Datasets		Yamanishi_08 [94]	Kuang [95]	Luo [15]	Modified Luo	
	Similarity Methods	AUC				
Drug	Chemical Structure	0.58	0.59	0.62	0.63	
	Chemical Structure+ ATC	0.62	0.61	0.69	0.67	
	Chemical Structure+ ATC+ Target	0.68	0.65	0.74	0.71	
	Chemical Structure+ ATC+ Target + Therapeutic	0.74	0.72	0.78	0.75	
	Chemical Structure+ ATC+ Target+ Therapeutic+ Side Effects	0.77	0.78	0.81	0.8	
	Chemical Structure+ ATC+ Target+ Therapeutic+ Side Effects+ MF	0.8	0.81	0.84	0.85	
	Chemical Structure+ ATC+ Target+ Therapeutic+ Side Effects+ MF+CC	0.83	0.85	0.85	0.84	
	Chemical Structure+ ATC+ Target+ Therapeutic+ Side Effects+ MF+ CC +BF	0.85	0.84	0.86	0.87	
	Chemical Structure+ ATC+ Target+ Therapeutic+ Side Effects+ MF+ CC+ BF + DDI	0.87	0.86	0.88	0.89	
	Chemical Structure+ ATC+ Target +Therapeutic+ Side Effects+ MF+ CC+ BF+ DDI+ Drug-Disease Association	0.90	0.89	0.91	0.92	
	Chemical Structure+ ATC+ Target +Therapeutic+ Side Effects+ MF+ CC+ BF+ DDI+ Drug-Disease Association+ Drug-Target (GIP-based)	0.93	0.92	0.96	0.95	
	Chemical Structure+ ATC+ Target +Therapeutic+ Side Effects+ MF+ CC+ BF+ DDI+ Drug-Disease Association+ Drug-Target (GIP-based)+ Pathway	0.95	0.96	0.98	0.97	
	Target	Protein-Sequence	0.67	0.65	0.78	0.75
		Protein-Sequence + Protein Structure	0.74	0.71	0.81	0.8
Protein-Sequence+ Protein Structure+ MF		0.78	0.75	0.84	0.85	
Protein-Sequence+ Protein Structure+ MF+ CC		0.83	0.82	0.85	0.84	
Protein-Sequence+ Protein Structure+ MF+ CC+ BF		0.84	0.85	0.86	0.87	
Protein-Sequence+ Protein Structure+ MF+ CC+ BF + PPI		0.85	0.87	0.88	0.89	
Protein-Sequence+ Protein Structure+ MF+ CC+BF+ PPI+ Protein- Disease Association		0.89	0.86	0.91	0.92	
Protein-Sequence+ Protein Structure+ MF+ CC+ BF + PPI + Drug-Target (GIP based)		0.91	0.93	0.94	0.95	
Protein-Sequence+ Protein Structure+ MF+ CC+ BF + PPI + Drug-Target (GIP-based)+ Pfam		0.94	0.95	0.97	0.96	

4.3.2. Use of a novel IERW for DTI prediction

Our proposed system represents the DSNs (12) and protein similarity networks (9) as homogeneous networks. In order to illustrate the walk sequences generated by our proposed IERW method, two existing methods, namely, the RW [33] and RWR [34] methods, have been applied to four different datasets for DTI (link) prediction. Also, the same methods have been applied to two additional tasks (applications), i.e., DDI [85] and DDiA prediction [86], as these tasks also use the similarity networks of drugs and diseases from multiple sources. Table 4 shows a comparison of the proposed IERW method with the RW and RWR methods on the DDI, DDiA and DTI prediction tasks. Table 4 interprets that the feature vector representation generated by the IERW method is very effective, with an increase in the AUC score of 14, 14 and 15% (average of all datasets) for the application of DDI, DDiA and DTI (proposed), respectively.

Table 4. Comparing the effects of feature vector representations for DTI prediction.

Dataset	Feature vector representation method	AUC	AUPR
Enzyme	RW	0.81	0.8
	RWR	0.84	0.85
	IERW	0.95	0.96
IC	RW	0.79	0.81
	RWR	0.86	0.83
	IERW	0.97	0.95
GPCR	RW	0.82	0.81
	RWR	0.83	0.86
	IERW	0.98	0.96
NR	RW	0.83	0.81
	RWR	0.87	0.86
	IERW	0.98	0.97
Kuang et al. [95]	RW	0.82	0.79
	RWR	0.87	0.85
	IERW	0.96	0.95
Luo [15]	RW	0.81	0.8
	RWR	0.84	0.85
	IERW	0.97	0.95
Modified Luo	RW	0.83	0.81
	RWR	0.87	0.88
	IERW	0.99	0.98
Rohani and Eslahchi et al. (DDI) [100]	RW	0.81	0.79
	RWR	0.87	0.85
	IERW	0.95	0.96
Zhou et al. [101] (DDiA)	RW	0.82	0.84
	RWR	0.85	0.86
	IERW	0.96	0.97

The IERW outperforms the other two methods on DTI prediction since the walk sequences consider both immediate (local) and non-immediate (global) neighbors during the selection of the best

context neighbor node, thereby reducing the false negatives that occur during link prediction. Therefore, our proposed IERW can be used for any kind of homogeneous network to learn the vector representation of each node in the network, which can be used for any downstream tasks.

4.3.3. Impact of MVCNN-based fusion for DTI prediction

To analyze the performance of our proposed MVCNN method, SNF and SKF methods have been considered for the same task as described in Section 4.3.2. These two applications also use the fusion techniques to combine the similarity networks of drugs and diseases.

Table 5. Performance of different SNI methods.

Dataset	Integration Method	AUC	AUPR
Enzyme	SNF	0.83	0.81
	SKF	0.87	0.86
	MVCNN	0.98	0.97
IC	SNF	0.81	0.79
	SKF	0.87	0.85
	MVCNN	0.95	0.96
GPCR	SNF	0.82	0.81
	SKF	0.83	0.86
	MVCNN	0.98	0.96
NR	SNF	0.79	0.81
	SKF	0.86	0.83
	MVCNN	0.97	0.95
Kuang et al. [95]	SNF	0.81	0.8
	SKF	0.84	0.85
	MVCNN	0.95	0.96
Luo [15]	SNF	0.83	0.81
	SKF	0.87	0.88
	MVCNN	0.99	0.98
Modified Luo	SNF	0.82	0.79
	SKF	0.87	0.85
	MVCNN	0.96	0.95
Rohani and Eslahchi et al. (DDI) [100]	SNF	0.81	0.8
	SKF	0.84	0.85
	MVCNN	0.97	0.95
Zhou et al. [101] (DDiA)	SNF	0.81	0.79
	SKF	0.87	0.85
	MVCNN	0.95	0.96

From the experimental analysis shown in Table 5, the MVCNN surpasses the other two methods in terms of the ability to identify different kinds of associations since SNF and SKF methods update the fused similarity matrix score by selecting only the edges that have a high score; alternatively, the MVCNN also considers the edges whose score is lower but can nevertheless contribute to the prediction of some DTI pairs. The performance improvement of the AUPR score in terms of the prediction of DDI, DDiA and DTI as a result of using the MVCNN is 14, 14 and 15%, respectively.

Example: According to our HBN, the drug trovafloxacin does not have any interaction with the target gene or protein ACE. The actual similarity score of this drug to other drugs and targets is lower. Based on the SNF and SKF methods, the edges of this drug and target are removed, whereas, in the case of MVCNN-based SNI, the drug trovafloxacin is interacting with the target ACE with a probability score of 0.687 because the edges are not removed from the network. As a result, the edges with lower similarity scores also contributes to some DTI pairs.

4.3.4. Novel meta-graph-based representation learning for DTI prediction

Meta-graph-based representation learning is used to represent each drug and target node respectively in a d -dimensional vector. This plays an important role in DTI prediction since the feature vector generated for similar nodes is represented as being close to each other in the vector or embedding space. According to our task, the closer the vector, the higher the probability of interaction. To prove the performance of our novel meta-graph-based representation learning, each type of meta-graph is analyzed. In order to make the model effectively learn the features, various meta-graphs with the same or different semantics and different hop lengths, i.e., 2 and 3, are used, and the result is illustrated in Table 6. When these embeddings are integrated into a single embedding representation, the CNN model exhibits better performance in terms of the prediction of drug-target pairs. Table 6 interprets that the system achieves better performance for the aggregation of a meta-graph with different semantics (a combination of similarity, interaction and association) for hop length = 2, and that the AUC score is increased by 11% on the modified Luo dataset. The other datasets, i.e., Enzyme, IC, GPCR and NR, perform better for the aggregation of different meta-graphs with similarity and interaction semantics of hop length = 2.

From the aggregated meta-graph-based representation learning taken from Table 6, the model has been compared with three other representation learning methods, namely, node2vec [85], metapath2vec [86] and a meta-path guided GNN [46], in terms of DTI prediction performance, as shown in Figure 12. Using a meta-graph guided GNN model and a CNN model for link prediction produces better performance, with an increase in the score of 8% and 6% for the AUC and AUPR, respectively as shown in Table 7.

The significant improvement of our representation learning algorithm is attributed to its ability to learn the interactions between drugs and targets when there is more than one semantic linear path between the same entities. However, the novel meta-graph guided GNN loses the actual DTI information to the specific DTI pairs, as explained in Section 3.3.4. Therefore, the system is examined with an enhanced node embedding vector for DTI prediction. The improvement in the performance is depicted in Figure 12, and the AUPR and AUC score are near 1 for the meta-graph-based GNN with a vector enhancement (VE) approach.

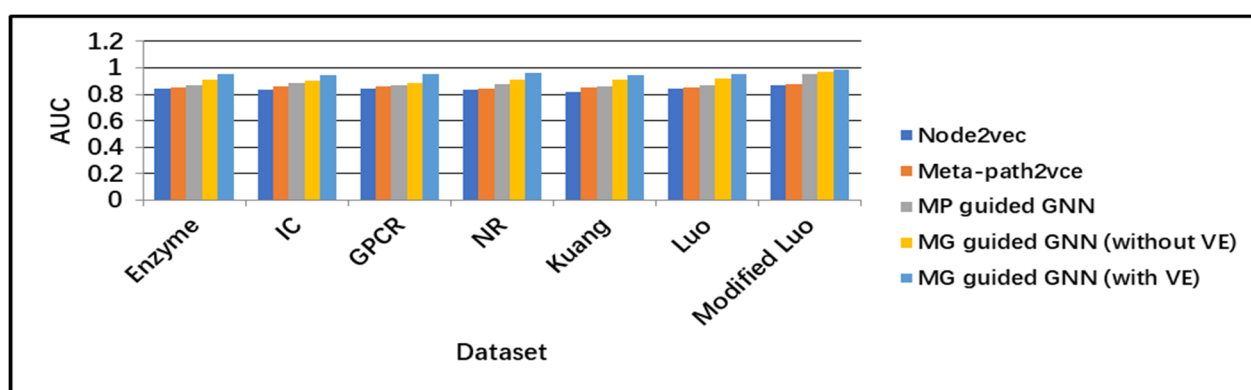
Table 6. Comparison of different meta-graphs with hop length = 2, 3 for four gold standard datasets.

Hop length	Meta-graphs	Enzyme	IC	GPCR	NR	Kuang [95]	Luo [15]	Modified Luo
AUC								
2	Aggregation (S)	0.879	0.861	0.867	0.879	0.868	0.858	0.891
	Aggregation (I)	0.90	0.908	0.911	0.916	0.899	0.897	0.924
	Aggregation (S+I)	0.928	0.923	0.917	0.921	0.929	0.921	0.952
	Aggregation (S+A)	N/A	N/A	N/A	N/A	N/A	N/A	0.967
	Aggregation (I+A)	N/A	N/A	N/A	N/A	N/A	N/A	0.948
	Aggregation (S, I, S+I, S+I+A)	N/A	N/A	N/A	N/A	N/A	N/A	0.992
3	Aggregation (S)	0.869	0.851	0.857	0.869	0.858	0.848	0.871
	Aggregation (I)	0.891	0.918	0.889	0.876	0.89	0.874	0.91
	Aggregation (S+I)	0.918	0.923	0.915	0.91	0.927	0.911	0.932
	Aggregation (S+A)	N/A	N/A	N/A	N/A	N/A	N/A	0.925
	Aggregation (I+A)	N/A	N/A	N/A	N/A	N/A	N/A	0.916
	Aggregation (S, I, S+I, S+I+A)	N/A	N/A	N/A	N/A	N/A	N/A	0.971

**N/A not applicable, S- Similarity, I- Interaction, A- Association

Table 7. Comparison of different representation learning models for DTI prediction.

Method	Enzyme	IC	GPCR	NR	Kuang [95]	Luo [15]	Modified Luo
AUC							
Node2vec	0.843	0.831	0.84	0.838	0.821	0.842	0.871
Meta-path2vce	0.851	0.857	0.856	0.843	0.85	0.849	0.876
MP guided GNN	0.871	0.887	0.867	0.879	0.861	0.869	0.949
MG guided GNN (without VE)	0.91	0.904	0.889	0.911	0.908	0.916	0.972
MG guided GNN (with VE)	0.951	0.945	0.956	0.958	0.942	0.949	0.990

**Figure 12.** Comparison of different representation learning models for DTI prediction.

4.3.5. Comparison of performance with state-of-the-art methods

The strength of our proposed method is investigated by comparing the performance of DTiGNN with six state-of-the-art methods by using the dataset described in Section 3.1. The state-of-the-art methods are DTINet [15], DRR [16], DTICNN [91], HNEDTI [48], metapath2vec [49] and DTiGEMS+ [31]. These methods were chosen to compare the overall performance of the DTiGNN framework to various embedding-based and matrix-factorization-based methods. The above methods use various similarity measures for drugs and targets from multiple sources. The proposed system (DTiGNN with VE) surpasses the other methods with an increase in the number of novel predictions, and the strength (probability score of interaction) of the DTI is also improved compared to the existing methods. Our proposed work has more than 100 pairs compared to the DTiGEMS+ method.

Furthermore, the results of each method are classified into two groups: with and without balancing the drug-target samples prior to the DTI prediction model. Here, the performance of the proposed framework outperforms all other methods with balancing techniques, as shown in Table 8. The system achieves better AUC performance, with a score of 0.99, which is 10% higher than the metapath2vec method. Finally, the DTI pairs identified by our proposed DTiGNN system are compared to available benchmark databases, such as DrugBank [55], CTD [62], PubMed [97], ChEMBL [98] and others.

Figure 13(a) and (b) shows the receiver operating characteristics curves and precision-recall curves for DRR, DTICNN, HNEDTI, metapath2vec, DTiGEMS+ and the proposed DTiGNN model on the DTI prediction task. Our proposed model achieves higher AUC and AUPR scores than various state-of-the-art methods, illustrating that the DTiGNN model has been designed appropriately.

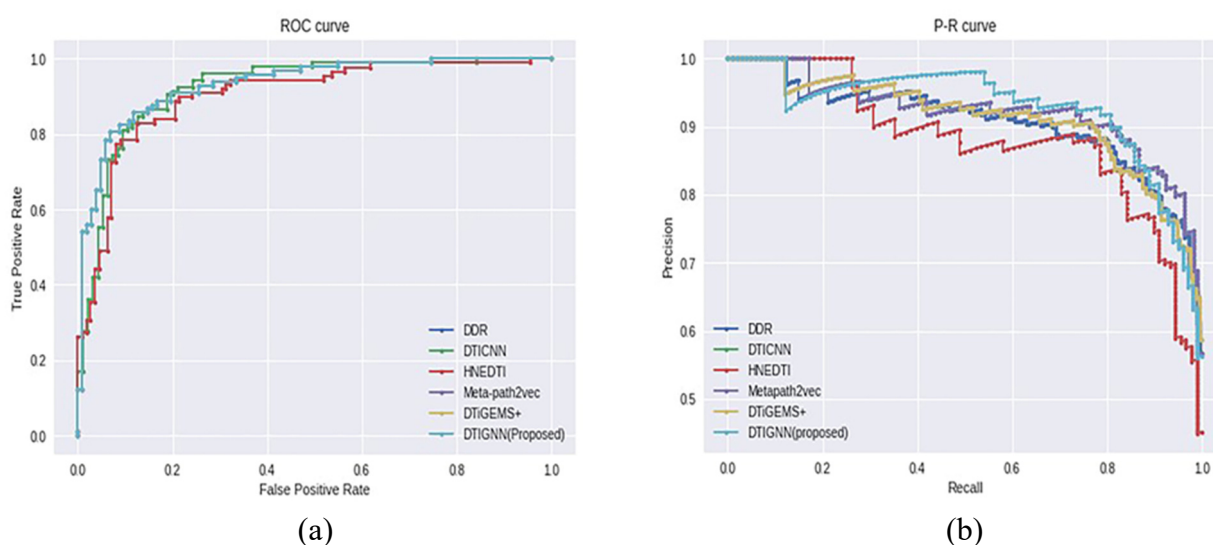


Figure 13. Receiver operating characteristic (ROC) curve (a) and precision-recall (P-R) curve (b).

Table 8. Comparison of the performance of proposed framework with state-of-the-art methods.

Balancing	Method	Dataset						
		Enzyme	IC	GPCR	NR	Kuang [95]	Luo [15]	Modified Luo
Without balancing	DRR	0.821	0.832	0.831	0.834	0.847	0.849	0.859
	DTICNN	0.828	0.824	0.830	0.832	0.833	0.849	0.853
	HNEDTI	0.79	0.812	0.797	0.808	0.812	0.817	0.823
	Metapath2vec	0.828	0.832	0.831	0.834	0.847	0.849	0.859
	DTiGEMS+	0.836	0.827	0.826	0.831	0.849	0.852	0.864
	DTiGNN (without VE)	0.861	0.841	0.862	0.851	0.87	0.882	0.897
	DTiGNN (with VE)	0.894	0.882	0.895	0.908	0.912	0.913	0.920
With Balancing	DRR	0.812	0.824	0.811	0.817	0.823	0.821	0.83
	DTICNN	0.828	0.832	0.831	0.834	0.847	0.849	0.859
	HNEDTI	0.816	0.827	0.836	0.831	0.849	0.852	0.864
	Metapath2vec	0.861	0.841	0.862	0.851	0.87	0.882	0.897
	DTiGEMS+	0.894	0.882	0.895	0.908	0.912	0.913	0.920
	DTiGNN (without VE)	0.925	0.928	0.917	0.928	0.931	0.943	0.951
	DTiGNN (with VE)	0.952	0.938	0.949	0.957	0.971	0.97	0.99

4.3.6. Impact of HBN

The system is able to predict the new DTIs with the DSN and known DTIs alone. The number of networks contributing to the DTI prediction is increased to improve prediction accuracy, as shown in Table 9.

Table 9. Comparison of the performance after including additional edges to the HBN.

Method	Dataset	Luo		Modified Luo	
		AUC	AUPR	AUC	AUPR
Drug	Drug similarities	0.849	0.839	0.869	0.861
	DDI	0.862	0.872	0.90	0.908
	Drug-disease	0.883	0.898	0.928	0.923
Protein (target)	Protein similarities	0.931	0.927	0.948	0.961
	Protein-protein interaction	0.947	0.941	0.967	0.951
	Protein-disease	0.964	0.972	0.992	0.981

Table 9 shows that the performance of the model is improved when additional edges are added to the network. The performance of AUC and AUPR increased by 12 and 13% on the Luo dataset, respectively, and by 13 and 12% on the modified Luo dataset. From these results, we inferred that the

model learns the vector representation of each node in a better way when the additional edges and information are added to the network.

4.3.7. Evidence for the prediction of DTI pairs

The number of DTI pairs predicted by our proposed system is higher than that for the state-of-the-art methods. Among all of the newly predicted DTI pairs, Table 10 shows the top 15 pairs. Here, the listed DTI pairs are unknown, that is, they do not have the edge in between as in the original HBN. This is validated with the help of already-approved drugs from different data sources like DrugBank [55], CTD [62], ChEMBL [99], etc. Most of the drug-target pairs identified by our proposed system can be found in the above-mentioned data sources. Approximately 70 out of 100 pairs are supported by different data sources.

These top interactions obtained from the proposed model are validated with the help of docking analysis. For the docking analysis, we have selected the AutoDoc Vina [97] tool, and the same is shown in Table 10. The affinity score that we obtained from the AutoDoc tool is also good for the top-ranked DTI pairs. From this, we can conclude that the DTIs identified from our proposed model are good compared to the state-of-the-art methods.

Table 10. Top 15 new DTI pairs with evidence.

Drug Id	Drug Name	Uniprot ID	Gene Name	Score (Proposed)	Evidence	Affinity score using AutoDoc Vina tool
DB00626	Bacitracin	P11229	CHRM1	0.998	1	-9.7
DB00829	Diazepam	P04150	NR3C1	0.996	1	-8.8
DB00945	Aspirin	Q12809	KCNH2	0.994	1	-9.3
DB06694	Xylometazoline	Q99460	PSMD1	0.991	1	-9.2
DB00509	Dextrothyroxine	P43088	PTGFR	0.990	1	-9.1
DB00768	Olopatadine	Q99519	NEU1	0.988	0	-9.0
DB06725	Lornoxicam	Q99519	NEU1	0.987	1	-8.9
DB00404	Alprazolam	P10745	RBP3	0.987	1	-8.9
DB01392	Yohimbine	P30968	GNRHR	0.985	1	-8.7
DB00252	Phenytoin	P04150	NR3C1	0.984	1	-9.6
DB01588	Prazepam	P31645	SLC6A4	0.983	0	-8.4
DB04552	Niflumic acid	P31645	SLC6A4	0.983	1	-7.7
DB00363	Clozapine	P10745	RBP3	0.982	1	-8.6
DB06288	Amisulpride	P30968	GNRHR	0.981	1	9.2
DB00546	Adinazolam	P22888	LHCGR	0.98	1	-7.9

4.3.8. Statistical testing analysis of various models

To prove our improved DTI prediction task, statistical testing analysis was carried out on an IERW and an enhanced representation learning model. We adopted 10-fold cross-validation, as discussed earlier in Section 4.2. To see the statistical difference between the models, a paired t-test was chosen,

and it was done by comparing each of the existing models with our proposed model. Assume a p-value of 0.05, three degrees of freedom (n-1), a null hypothesis of “no significant difference between the models” and the alternate hypothesis of “significant difference between the models”. Then, the t-statistic value is determined by using Eq (4.6) as follows:

$$t - \text{statistic} = t = \frac{\sqrt{N} * m}{sd} \quad 4.6$$

$$sd = \sqrt{\frac{\sum_{i=1}^N (diff_i - m)^2}{m}}$$

$$m = \frac{1}{N} (diff_i)$$

$$diff_i = ACC(\text{existing model}_{j=1..t}) - ACC(\text{model}_{proposed})$$

The t-statistic value does not fall within the range of $[-3.182 \text{ and } +3.182]$ for any of the pairs for the above-mentioned algorithms. Hence, the null hypothesis is rejected, proving that the models are statistically different and not the same.

4.3.9. Model interpretation

Figure 14 shows the vector representation of drug and protein nodes learned by a novel meta-graph guided GNN model in vector space by using a t-distributed stochastic neighbor embedding (t-SNE) visualization algorithm. The figure shows that nodes in the heterogeneous network that are structurally closer are also closer in vector space. The red dots represent proteins, while the yellow dots represent drugs. The dots that are closer together have a greater chance of interacting with one another.

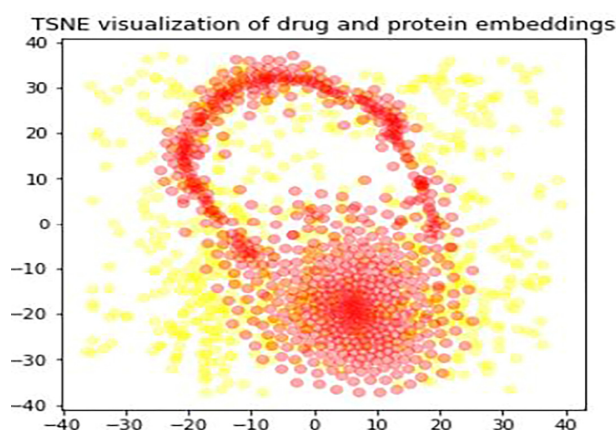


Figure 14. Embedding vector of drug and protein in vector space.

4.3.10. Case study using three drugs

To analyze the performance of the proposed model, we downloaded the known DTIs from DrugBank. We chose the drugs with the most interactions with known DTIs for this study. For example,

we considered three drugs, i.e., DB00334, DB00371 and DB01224, and the numbers of interactions were 24, 24 and 23, respectively. During the training, the features of the considered drugs and targets were excluded, and the same was given as an input to test the model. Our model found 23 known interactions out of 24 in the DB00334 set, and 24 known interactions out of 24 in the DB00371 set. Finally, in the DB01224 set, 23 out of 23 known interactions were identified. These results indicate that the proposed DTiGNN model performs well in terms of DTI prediction.

5. Conclusions

In this work, a new computational DTiGNN framework was developed to predict the unknown interaction between drug-target pairs. This system takes advantage of multiple similarities between the drug and target, including pathway- and protein domain-based similarities. The work on the IERW is used to generate the walk sequences of each node in each similarity, from which the feature vector representation of all drug and target nodes are individually learned by using the SG model. Next, the feature vector representations of all perspectives of each drug and target node are fused into a single vector representation by using a MVCNN. To learn the embedding vector of the drug and target nodes from the HBN, a meta-graph guided GNN is used. The feature vector representation of the drug-target pair is balanced by using oversampling techniques such as SMOTE, k-means SMOTE and a GAN. With the balancing samples, the feature vector of the drug-target pair is fed into the CNN model to predict the probability score. Here, the performance of the proposed system is evaluated with the help of AUC and AUPR metrics. The proposed system showed good performance compared to six existing frameworks for DTI prediction. These DTI pairs can be used by researchers or clinicians to determine the repurposing of drugs, personalized medicine, etc. The same framework can be used to identify the new association between any pair of biological entities. In future work, the DTI prediction system can be improved by considering an additional number of nodes and networks, such as disease similarity networks, and the incorporation of those kinds of meta-graphs can improve the reliability of the DTI pairs.

Acknowledgments

We thank all of the biomedical data sources curated by various authors for providing data for the prediction. This research did not receive a specific grant from funding agencies in the public, commercial or not-for-profit sectors.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. J. A. DiMasi, R. W. Hansen, H. G. Grabowski, The price of innovation: new estimates of drug development costs, *J Health Econ.*, **22** (2003), 151–185. [https://doi.org/10.1016/s0167-6296\(02\)00126-1](https://doi.org/10.1016/s0167-6296(02)00126-1)

2. M. A. Yildirim, K. Goh, M. E. Cusick, A. Barabas, M. Vidal, Drug-target network, *Nat. Biotechnol.*, **25** (2007), 1119–1126. <https://doi.org/10.1038/nbt1338>
3. T. T. Ashburn, K. B. Thor, Drug repositioning: identifying and developing new uses for existing drugs, *Nat. Rev. Drug Discov.*, **3** (2004), 673–683. <https://doi.org/10.1038/nrd1468>
4. F. Cheng, C. Liu, J. Jiang, W. Lu, W. Li, G. Liu, et al., Prediction of drug-target interactions and drug repositioning via network-based inference, *PLoS Comput. Biol.*, **8** (2012), 1002503. <https://doi.org/10.1371/journal.pcbi.1002503>
5. Y. Yamanishi, M. Kotera, M. Kanehisa, S. Goto, Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework, *Bioinformatics*, **26** (2010), 246–254. <https://doi.org/10.1093/bioinformatics/btq176>
6. A. C. Cheng, R. G. Coleman, K. T. Smyth, Q. Cao, P. Soulard, D. R. Caffrey, et al., Structure-based maximal affinity model predicts small-molecule druggability, *Nat. Biotechnol.*, **25** (2007), 71–75. <https://doi.org/10.1038/nbt1273>
7. H. Alonso, A. A. Bliznyuk, J. E. Gready, Combining docking and molecular dynamic simulations in drug design, *Med. Res. Rev.*, **26** (2006), 531–568. <https://doi.org/10.1002/med.20067>
8. K. Wang, J. Sun, S. Zhou, C. Wan, S. Qin, C. Li, et al., Prediction of drug–target interactions for drug repositioning only based on genomic expression similarity, *PLoS Comput. Biol.*, **9** (2013), e1003315. <https://doi.org/10.1371/journal.pcbi.1003315>
9. A. Tropsha, Best practices for QSAR model development, validation, and exploitation, *Mol. Inform.*, **29**(2010), 476–488. <http://doi.org/10.1002/minf.201000061>
10. D. Rognan, Chemogenomic approaches to rational drug design, *Br. J. Pharmacol.*, **152** (2007), 38–52. <https://doi.org/10.1038/sj.bjp.0707307>
11. A. C. A. Nascimento, R. B. C. Prudêncio, I. G. Cost, A multiple kernel learning algorithm for drug-target interaction prediction, *BMC Bioinformatics*, **17** (2016), 1–16, <https://doi.org/10.1186/s12859-016-0890-3>
12. T. He, M. Heidemeyer, F. Ban, A. Cherkasov, M. Ester, SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines, *J. Cheminform.*, **9** (2017), 1–14. <https://doi.org/10.1186/s13321-017-0209-z>
13. K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, P. Zhang, Interpretable drug target prediction using deep neural representation, in *IJCAI International Joint Conferences on Artificial Intelligence Organization*, (2018), 3371–3377. <https://doi.org/10.24963/ijcai.2018/468>
14. M. Tsubaki, K. Tomii, J. Sese, Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences, *Bioinformatics* **35** (2019), 309–318. <https://doi.org/10.1093/bioinformatics/bty535>
15. Y. Luo, X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, et al., A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information, *Nat. Commun.*, **8** (2017), 573. <https://doi.org/10.1038/s41467-017-00680-8>
16. R. S. Olayan, H. Ashoor, V. B. Bajic, DDR: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches, *Bioinformatics*, **34** (2018), 1164–1173. <https://doi.org/10.1093/bioinformatics/btx731>
17. W. Zhang, W. Lin, D. Zhang, S. Wang, J. Shi, Y. Niu, Recent advances in the machine learning-based drug-target interaction prediction, *Curr. Drug Metab.*, **20** (2019), 194–202. <https://doi.org/10.2174/1389200219666180821094047>

18. M. Thafar, A. B. Raies, S. Albaradei, M. Essack, V. B. Bajic, Comparison study of computational prediction tools for drug–target binding affinities, *Front. Chem.*, **7** (2019), 782. <https://doi.org/10.3389/fchem.2019.00782>
19. S. Anusuya, M. Keshewani, K. V. Priya, A. Vimala, G. Shanmugam, D. Velmurugan, et al., Drug–target interactions: prediction methods and applications, *Curr. Protein Pept. Sci.*, **19** (2018), 537–561. <https://doi.org/10.2174/1389203718666161108091609>
20. K. Sachdev, M. K. Gupta, A comprehensive review of feature based methods for drug target interaction prediction, *J Biomed Inform.*, **93** (2019), 103159. <https://doi.org/10.1016/j.jbi.2019.103159>
21. W. Chen, L. Kurgan, Survey of similarity-based prediction of drug–protein interactions, *Curr. Med. Chem.*, **27** (2020), 5856–5886. <https://doi.org/10.2174/0929867326666190808154841>
22. K. Abbasi, P. Razzaghi, A. Poso, S. Ghanbari-Ara, A. Masoudi-Nejad, Deep learning in drug target interaction prediction, *Curr. Med. Chem.*, **28** (2021), 2100–2113. <https://doi.org/10.2174/0929867327666200907141016>
23. Q. Bai, J. Ma, S. Liu, T. Xu, A. J. Banegas-Luna, H. Pérez-Sánchez, et al., WADDAICA: a webserver for aiding protein drug design by artificial intelligence and classical algorithm, *Comput. Struct. Biotechnol. J.*, **19** (2021), 3573–3579. <https://doi.org/10.1016/j.csbj.2021.06.017>
24. Q. Bai, S. Tan, T. Xu, H. Liu, J. Huang, X. Yao, MolAICal: a soft tool for 3D drug design of protein targets by artificial intelligence and classical algorithm, *Brief. Bioinform.*, **22** (2021), 161. <https://doi.org/10.1093/bib/bbaa161>
25. Q. Bai, S. Liu, Y. Tian, T. Xu, A. J. Banegas-Luna, H. Pérez-Sánchez, et al., Application advances of deep learning methods for de novo drug design and molecular dynamics simulation, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, **12** (2022), 1581. <https://doi.org/10.1002/wcms.1581>
26. M. Koutrouli, E. Karatzas, D. Paez-Espino, G. A. Pavlopoulos, A guide to conquer the biological network era using graph theory, *Front. Bioeng. Biotechnol.*, **8** (2020), 34. <http://doi.org/10.3389/fbioe.2020.00034>
27. Y. Tabei, M. Kotera, R. Sawada, Y. Yamanishi, Network-based characterization of drug–protein interaction signatures with a space-efficient approach, *BMC Syst. Biol.*, **39** (2019). <https://doi.org/10.1186/s12918-019-0691-1>
28. W. Wang, H. Lv, Y. Zhao, D. Liu, Y. Wang, Y. Zhang, DLS: A link prediction method based on network local structure for predicting drug–protein interactions, *Front. Bioeng. Biotechnol.*, **8** (2020). <https://doi.org/10.3389/fbioe.2020.00330>
29. Z. Wu, W. Li, G. Liu, Y. Tang, Network-based methods for prediction of drug–target interactions, *Front. Pharmacol.*, **9** (2018), 1134. <https://doi.org/10.3389/fphar.2018.01134>
30. K. Abbas, A. Abbasi, S. Dong, L. Niu, L. Yu, B. Chen, et al., Application of network link prediction in drug discovery, *BMC Bioinformatics*, **187** (2021). <https://doi.org/10.1186/s12859-021-04082-y>
31. M. A. Thafar, R. S. Olayan, H. Ashoor, S. Albaradei, V. B. Bajic, X. Gao, et al., DTiGEMS+: drug–target interaction prediction using graph embedding, graph mining, and similarity-based techniques, *J. Cheminform.*, **12** (2020), 1–17. <https://doi.org/10.1186/s13321-020-00447-2>
32. L. Wang, Z. You, X. Chen, S. Xia, F. Liu, X. Yan, et al., Computational methods for the prediction of drug–target interactions from drug fingerprints and protein sequences by stacked auto-encoder deep neural network, in *International Symposium on Bioinformatics Research and Applications*, 46–58. https://doi.org/10.1007/978-3-319-59575-7_5

33. B. Agyemang, W. P. Wu, M. Y. Kpiebaareh, Z. Lei, E. Nanor, L. Chen, Multi-view self-attention for interpretable drug–target interaction prediction, *J. Biomed. Inform.*, **110** (2020), 103547. <https://doi.org/10.1016/j.jbi.2020.103547>
34. P. Zhang, Z. Wei, C. Che, B. Jin, DeepMGT-DTI: Transformer network incorporating multilayer graph information for Drug-Target interaction prediction, *Comput. Biol. Med.*, **142** (2022), 105214. <https://doi.org/10.1016/j.compbiomed.2022.105214>
35. C. Chen, H. Shi, Z. Jiang, A. Salhi, R. Chen, X. Cui, et al., DNN-DTIs: Improved drug-target interactions prediction using XGBoost feature selection and deep neural network, *Comput. Biol. Med.*, **136** (2021), 104676. <https://doi.org/10.1016/j.compbiomed.2021.104676>
36. X. Zeng, S. Zhu, W. Lu, Z. Liu, J. Huang, Y. Zhou, et al., Target identification among known drugs by deep learning from heterogeneous networks, *Chem. Sci.*, **11** (2020), 1775–1797. <https://doi.org/10.1039/C9SC04336E>
37. B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, et al., Similarity network fusion for aggregating data types on a genomic scale, *Nat. Methods*, **11** (2014), 333–337. <https://doi.org/10.1038/nmeth.2810>
38. F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, X. Kong, Random walks: A review of algorithms and applications, *IEEE Trans. Emerging Top. Comput. Intell.*, **4** (2019), 95–107. <https://doi.org/10.1109/TETCI.2019.2952908>
39. A. Ghulam, X. Lei, M. Guo, C. Bian, Disease-pathway association prediction based on random walks with restart and PageRank, *IEEE Access*, **8** (2020), 72021–72038. <https://doi.org/10.1109/ACCESS.2020.2987071>
40. S. Li, L. Jiang, J. Tang, N. Gao, F. Guo, Kernel fusion method for detecting cancer subtypes via selecting relevant expression data, *Front. Genet.*, **11** (2020), 979. <https://doi.org/10.3389/fgene.2020.00979>
41. R. Qin, L. Duan, H. Zheng, J. Li-Ling, K. Song, Y. Zhang, An ontology-independent representation learning for similar disease detection based on multi-layer similarity network, *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **18** (2021), 183–193. <https://doi.org/10.1109/tcbb.2019.2941475>
42. H. I. Lee, H. Nam, Identification of drug-target interaction by a random walk with restart method on an interactome network, *BMC Bioinformatics*, **208** (2018), 9–18. <https://doi.org/10.1186/s12859-018-2199-x>
43. T. N. Jarada, J. G. Rokne, R. Alhajj, SNF-NN: computational method to predict drug-disease interactions using similarity network fusion and neural networks, *BMC Bioinformatics*, **28** (2021), 1–20. <https://doi.org/10.1186/s12859-020-03950-3>
44. D. V. Anand, Q. Xu, J. Wee, K. Xia, T. C. Sum, Topological feature engineering for machine learning based halide perovskite materials design, *Npj Comput. Mater.*, **203** (2022), 1–8. <https://doi.org/10.1038/s41524-022-00883-8>
45. M. Wang, Z. Cang, G. W. Wei, A Topology-based network tree for the prediction of protein-protein binding affinity changes following mutation, *Nat. Mach. Intell.*, **2** (2020), 116–123. <https://doi.org/10.1038/s42256-020-0149-6>
46. S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, et al., Metapath-guided heterogeneous graph neural network for intent recommendation, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (2019), 2478–2486. <https://doi.org/10.1145/3292500.3330673>

47. X. Fu, J. Zhang, Z. Meng, I. King, MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding, in *Proceedings of The Web Conference*, (2020), 2331–234. <https://doi.org/10.1145/3366423.3380297>
48. Z. Lu, Y. Wang, M. Zeng, M. Li, HNEDTI: Prediction of drug-target interaction based on heterogeneous network embedding, in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, 211–214. <https://doi.org/10.1109/BIBM47256.2019.8983181>
49. M. Samizadeh, B. Minaei-Bidgoli, Drug-target interaction prediction by Metapath2vec node embedding in heterogeneous network of interactions, *Int. J. Artif. Intell. Tools*, **29** (2020), 2050001. <https://doi.org/10.1142/S0218213020500013>
50. X. Liu, H. Feng, J. Wu, K. Xia, Persistent spectral hypergraph-based machine learning (PSH-ML) for protein-ligand binding affinity prediction, *Brief. Bioinform.*, **22** (2021), 127. <https://doi.org/10.1093/bib/bbab127>
51. X. Liu, X. Wang, J. Wu, K. Xia, Hypergraph-based persistent cohomology (HPC) for molecular representations in drug design, *Brief. Bioinform.*, **22** (2021), 411. <https://doi.org/10.1093/bib/bbaa411>
52. G. A. Petsko, Guilt by association, *Genome Biol.*, **104** (2009), 1–3. <https://doi.org/10.1186/gb-2009-10-4-104>
53. H. Wang, H. Zheng, J. Wang, C. Wang, F. Wu, Integrating omics data with a multiplex network-based approach for the identification of cancer subtypes, *IEEE Trans. Nanobioscience*, **15** (2016), 335–342. <https://doi.org/10.1109/tnb.2016.2556640>
54. D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, et al., DrugBank 5.0: a major update to the DrugBank database for 2018, *Nucleic Acids Res.*, **46** (2018), 1074–1082. <https://doi.org/10.1093/nar/gkx1037>
55. D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, *J. Chem. Inf. Comput. Sci.*, **28** (1988), 31–36. <https://doi.org/10.1021/ci00057a005>
56. C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, E. Willighagen, Recent developments of the chemistry development kit (CDK)-an opensource java library for chemo-and bioinformatics, *Curr. Pharm. Des.*, **12** (2006), 2111–2120. <https://doi.org/10.1021/ci025584y>
57. A. Rácz, D. Bajusz, K. Héberger, Life beyond the Tanimoto coefficient: similarity measures for interaction fingerprints, *J. Cheminform.*, **10** (2018), 1–12. <https://doi.org/10.1186/s13321-018-0302-y>
58. M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, P. Bork, A side effect resource to capture phenotypic effects of drugs, *Mol. Syst. Biol.*, **6** (2010), 343. <https://doi.org/10.1038/msb.2009.98>
59. F. Zhu, TTD, 2009. Available from: <https://bio.tools/ttd>
60. UniProt consortium, Uniprot, 2023. Available from: <http://www.uniprot.org/>
61. T. F. Smith, M. S. Waterman, Identification of common molecular subsequences, *J Mol. Biol.*, **147** (1981), 195–197. [https://doi.org/10.1016/0022-2836\(81\)90087-5](https://doi.org/10.1016/0022-2836(81)90087-5)
62. A. P. Davis, C. J. Grondin, R. J. Johnson, D. Sciaky, B. L. King, R. McMorran, et al., The Comparative toxicogenomics database: update 2017, *Nucleic Acids Res.*, **45** (2017), 972–978. <https://doi.org/10.1093/nar/gkw838>
63. Gene Ontology, The Gene Ontology Resource, 2023. Available from: <http://www.geneontology.org/>

64. J. A. Blake, M. Dolan, H. Drabkin, D. P. Hill, L. Ni, D. Sitnikov, et al., Gene Ontology annotations and resources, *Nucleic Acids Res.*, **41** (2013), 530–535. <https://doi.org/10.1093/nar/gks1050>
65. G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, S. Wang, GOSemSim: an R package for measuring semantic similarity among GO terms and gene products, *Bioinformatics*, **26** (2010), 976–978. <https://doi.org/10.1093/bioinformatics/btq064>
66. S. S. Deepika, T. V. Geetha, A meta-learning framework using representation learning to predict drug-drug interaction, *J. Biomed. Inform.*, **84** (2018), 136–147. <https://doi.org/10.1016/j.jbi.2018.06.015>
67. L. Huang, H. Luo, S. Li, F. Wu, J. Wang, Drug-drug similarity measure and its applications, *Brief. Bioinform.*, **22** (2021), 265. <https://doi.org/10.1093/bib/bbaa265>
68. T. van Laarhoven, S. B. Nabuurs, E. Marchiori, Gaussian interaction profile kernels for predicting drug-target interaction, *Bioinformatics*, **27** (2011), 3036–3043. <https://doi.org/10.1093/bioinformatics/btr500>
69. H. Ye, K. Tang, L. Yang, Z. Cao, Y. Li, Study of drug function based on similarity of pathway fingerprint, *Protein Cell*, **3** (2012), 132–139. <https://doi.org/10.1007/s13238-012-2011-z>
70. M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, K. Morishima, KEGG: new perspectives on genomes, pathways, diseases and drugs, *Nucleic Acids Res.*, **45** (2017), 353–361. <https://doi.org/10.1093/nar/gkw1092>
71. L. R. Sancho, BioCor: Functional similarities. Available from: <https://llrs.github.io/BioCor/>
72. A. Herbert, MaxCluster, 2023. Available from: <http://www.sbg.bio.ic.ac.uk/~maxcluster/>
73. National Science Foundation, US Department of Energy, National Cancer Institute, National Institute of Allergy and Infectious Diseases, National Institute of General Medical Sciences of the National Institutes of Health, RCSB Protein Data Bank (RCSB PDB), 2023. Available from: <https://www.rcsb.org/>
74. I. Kufareva, R. Abagyan, Methods of protein structure comparison, *Homol. Model.*, **857** (2012), 231–257. https://doi.org/10.1007/978-1-61779-588-6_10
75. X. Wu, E. Pang, K. Lin, Z. M. Pei. Improving the measurement of semantic similarity between gene ontology terms and gene products: Insights from an edge- and IC-Based hybrid method, *PLoS ONE*, **8** (2013), 66745. <https://doi.org/10.1371/journal.pone.0066745>
76. M. Shatnawi, Review of recent protein-protein interaction techniques, in *Emerging Trends in Computational Biology, Bioinformatics and Systems Biology*, (2015), 99–121. <http://doi.org/10.1016/B978-0-12-802508-6.00006-5>
77. C. Menichelli, O. Gascuel, L. Bréhélin, Improving pairwise comparison of protein sequences with domain co-occurrence, *PLoS Comput. Biol.*, **14** (2018), 1005889. <https://doi.org/10.1371/journal.pcbi.1005889>
78. R. D. Finn, A. Bateman, J. Clements, P. Coggil, R. Y. Eberhardt, S. R. Eddy, et al., Pfam: the protein families database, *Nucleic Acids Res.*, **42** (2014), 222–230. <https://doi.org/10.1093/nar/gkt1223>
79. Y. M. Omar, P. Plapper, A survey of information entropy metrics for complex networks, *Entropy*, **22** (2020), 1417. <https://doi.org/10.3390/e22121417>
80. A. P. Riascos, J. L. Mateos, Random walks on weighted networks: Exploring local and non-local navigation strategies, 2019, preprint, arXiv:1901.05609.
81. Chris McCormick, Word2vec tutorial-the skip-gram model, 2016. Available from: <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>

82. B. Wang, A. Wang, F. Chen, Y. Wang, C. C. J. Kuo, Evaluating word embedding models: methods and experimental results, *APSIPA Trans. Signal Inf. Process.*, **8** (2019). <https://doi.org/10.1017/ATSIP.2019.12>
83. Y. Li, M. Yang, Z. Zhang, A survey of multi-view representation learning, *IEEE Trans. Knowl. Data Eng.*, **31** (2018), 1863–1883. <https://doi.org/10.1109/TKDE.2018.2872063>
84. K. Sun, L. Wang, B. Xu, W. Zhao, S. W. Teng, F. Xia, Network representation learning: From traditional feature learning to deep learning, *IEEE Access*, **8** (2020), 205600–205617. <https://doi.org/10.1016/j.jfds.2017.05.001>
85. A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2016), 855–864. <https://doi.org/10.1145/2939672.2939754>
86. Y. Dong, N. V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2017), 135–144. <https://doi.org/10.1145/3097983.3098036>
87. B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, (2014), 701–710. <https://doi.org/10.1145/2623330.2623732>
88. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in *ICLR*, (2018).
89. G. Douzas, F. Bacao, F. Last, Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE, *Inf. Sci.*, **465** (2018), 1–20. <https://doi.org/10.1016/j.ins.2018.06.056>
90. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, Generative adversarial networks: An overview, *IEEE Signal Proc. Mag.*, **35** (2018), 53–65. <https://doi.org/10.1109/MSP.2017.2765202>
91. J. Peng, J. Li, X. Shang, A learning-based method for drug-target interaction prediction based on feature representation learning and deep neural network, *BMC Bioinformatics*, **21** (2020), 1–13. <https://doi.org/10.1186/s12859-020-03677-1>
92. Z. Zheng, N. Q. K. Le, M. C. H. Chua, MaskDNA-PGD: An innovative deep learning model for detecting DNA methylation by integrating mask sequences and adversarial PGD training as a data augmentation method, *Chemom. Intell. Lab. Syst.*, **232** (2023), 104715. <https://doi.org/10.1016/j.chemolab.2022.104715>
93. J. N. Sua, S. Y. Lim, M. H. Yulius, X. Su, E. K. Y. Yapp, N. Q. K. Le, et al., Incorporating convolutional neural networks and sequence graph transform for identifying multilabel protein Lysine PTM sites, *Chemom. Intell. Lab. Syst.*, **206** (2020), 104171. <https://doi.org/10.1016/j.chemolab.2020.104171>
94. Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, M. Kanehisa, Prediction of drug-target interaction networks from the integration of chemical and genomic spaces, *Bioinformatics*, **24** (2008), 232–240. <https://doi.org/10.1093/bioinformatics/btn162>
95. Q. Kuang, X. Xu, R. Li, Y. Dong, Y. Li, Z. Huang, et al., An eigenvalue transformation technique for predicting drug-target interaction, *Sci. Rep.*, **5** (2015), 13867. <https://doi.org/10.1038/srep13867>

96. J. Davis, M. Goadrich, The relationship between precision-recall and ROC Curves, in *Proceedings of the 23rd international conference on machine learning*, (2006), 233–240. <https://doi.org/10.1145/1143844.1143874>
97. T. Nguyen, T. H. Nguyen, T. N. H. Pham, N. T. Huy, M. V. Bay, M. Q. Pham, et al., Autodockvina adopts more accurate binding poses but autodock4 forms better binding affinity, *J. Chem. Inf. Model.*, **60** (2019), 204–211. <https://doi.org/10.1021/acs.jcim.9b00778>
98. PubMed, *National Library of Medicine National Center for Biotechnology Information*, 2023. Available from: <https://pubmed.ncbi.nlm.nih.gov/>
99. European Molecular Biology Laboratory, ChEMBL, 2023. Available from: <https://www.ebi.ac.uk/chembl/>
100. N. Rohani, C. Eslahchi, Drug-drug interaction predicting by neural network using integrated similarity, *Sci. Rep.*, **9** (2019), 1–11. <https://doi.org/10.1038/s41598-019-50121-3>
101. R. Zhou, Z. Lu, H. Luo, J. Xiang, M. Zeng, M. Li, NEDD: a network embedding based method for predicting drug-disease associations, *BMC Bioinformatics*, **387** (2020), 1–12. <https://doi.org/10.1186/s12859-020-03682-4>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)