entropy

MDPI

Article

Identifying Candidate Gene–Disease Associations via Graph Neural Networks

Pietro Cinaglia [1,*] and Mario Cannataro [2]

[1] Department of Health Sciences, Magna Graecia University of Catanzaro, 88100 Catanzaro, Italy
[2] Data Analytics Research Center, Department of Medical and Surgical Sciences, Magna Graecia University of Catanzaro, 88100 Catanzaro, Italy
* Correspondence: cinaglia@unicz.it

Abstract: Real-world objects are usually defined in terms of their own relationships or connections. A graph (or network) naturally expresses this model though nodes and edges. In biology, depending on what the nodes and edges represent, we may classify several types of networks, gene–disease associations (GDAs) included. In this paper, we presented a solution based on a graph neural network (GNN) for the identification of candidate GDAs. We trained our model with an initial set of well-known and curated inter- and intra-relationships between genes and diseases. It was based on graph convolutions, making use of multiple convolutional layers and a point-wise non-linearity function following each layer. The embeddings were computed for the input network built on a set of GDAs to map each node into a vector of real numbers in a multidimensional space. Results showed an AUC of 95% for training, validation, and testing, that in the real case translated into a positive response for 93% of the Top-15 (highest dot product) candidate GDAs identified by our solution. The experimentation was conducted on the DisGeNET dataset, while the DiseaseGene Association Miner (DG-AssocMiner) dataset by Stanford's BioSNAP was also processed for performance evaluation only.

Keywords: graph neural network; gene disease associations; link prediction; neural network; deep learning

Citation: Cinaglia, P.; Cannataro, M. Identifying Candidate Gene–Disease Associations via Graph Neural Network. *Entropy* **2023**, *25*, 909. https://doi.org/10.3390/e25060909

Academic Editor: Deniz Gençağa

Received: 3 May 2023
Revised: 1 June 2023
Accepted: 5 June 2023
Published: 7 June 2023

1. Introduction

In recent years, a large amount of genomic and biological data has been studied in clinical research trials to correlate human diseases with genomics data, e.g., to define novel treatments. Disease profiling may include data from omics data (e.g., genomics, transcriptomics, and metabolomics) that could be related to disease susceptibility, progress and manifestation. The genes are crucial to understand the key factors involved in a correlated disease (e.g., molecular basis and biological mechanisms), as well as to evaluate the treatments and diagnosis. For instance, a phenotype may be caused by mechanisms which can be divided into several groups, where each one is characterized by a set of specific patterns and pathways that handle its activation. The knowledge related to this one could be very limited or misleading when the molecular mechanisms are unknown.

In bioinformatics, this topic has widely been tackled through the development of solutions for pattern recognition and data mining, besides data integration, in order to gather gene–disease associations (GDAs). A GDA is an association between a genetic variant with a disease or trait. The related branch of study focuses on investigating the divergences for one or more genes that could predispose to disease onset as well as be potentially responsible for the development of a specific disease phenotype. The GDAs are studied to identify the biological meaning for a given association that goes beyond pure randomness. To give an example, recent studies [1] proved the responsibility of aberrant interactions between mutant proteins, transcription factors and transcriptional co-activators that are the cause of the onset of Huntington's disease. Authors based their own study on processing large genomic regions in a coordinated fashion to also evaluate

disease progression and altered chromosomal clusters, according to these. Results support the notion of a common genome-wide mechanism of disruption of RNA transcription in the brain and periphery of Huntington's disease patients. Similarly, computational models are also used to infer novel microRNA (miRNA)–disease associations by allowing the identification of promising miRNA–disease pairs for experimental validation [2–5].

Based on a network build by taking into account an exhaustive set of GDAs, the genotype of one or more polymorphisms will be seen more often in an individual carrying the trait, in relation to the probability that the latter is due to chance [6]. In this context, DisGeNET [7] integrates heterogeneous data sources to provide a set of curated and inferred GDAs. It identifies the diseases and the genes by using the Unified Medical Language System-Concept Unique Identifier (UMLS-CUI, or simply UMLS) and the Entrez Gene unique integer (i.e., gene identifier, or GeneID), respectively [8]. DisGeNET uses multiple data mining techniques on biological and literature data to build its own database [9].

The former was analyzed by integrating a set of existing databases (i.e., Universal Protein, Psychiatric Disorders Gene Association Network, Orphanet, Cancer Genome Interpreter, Comparative Toxicogenomics Database for Human, ClinGen, and Genomics England PanelApp), while the latter by performing the named entity recognition (NER) [10] of genetic variants on natural language texts retrieved both from the Single Nucleotide Polymorphism (SNP) database (i.e., dbSNP) [11] and the Universal Protein (UniProt) database [12]. Similarly, Stanford's BioSNAP [13] provides a network-based dataset (Disease–Gene Association Miner, or DG-AssocMiner), containing a set of human GDAs. However, the latter contains much less information and GDAs than DisGeNET.

The real-world objects are usually defined in terms of their relationships or connections, which can be naturally expressed by using a graph (or network) model. In this paper, we will use the terms "graph" and "network" interchangeably.

In biology, depending on what the nodes and edges represent, we may classify several types of networks [14], such as gene regulatory networks, signaling networks, human disease networks, or protein interaction network. A network is also used to model gene–disease networks consisting of GDAs. In the last decade, the application of novel end-to-end deep learning (DL) paradigms boosted the research on pattern recognition and data mining in many domains [15–17]. For instance, the graph neural network (GNN) allows the modeling of graph data via NN, basing the computation on techniques originally designed for imaging. Briefly, an image can be represented by a matrix that can be transported into a Euclidean space to extract latent representation; therefore, the connections between the adjacent pixels can be treated as a graph. This approach may be considered biunivocal by adapting well-known paradigms from imaging to graph (or network) analysis [18]. We propose the following non-exhaustive example. Assuming $G = (V, E)$ an undirected and unweighted graph, with V and E, two sets of n nodes and m edges, respectively. We can calculate a weight for each edge so that it is representative of the node's neighborhood information. The resulting weighted graph G' can be used to compute a fixed length vector representation for each entity (i.e., embedding) based on the edge weights. Similarly to imaging, we could analyze a graph consisting of the nodes of interest (e.g., genes), instead of pixels.

As discussed, DL approaches for imaging base their own analysis on the Euclidean data obtained from the image data source. Therefore, we could apply the principle at the basis of a convolutional NN (CNN) for feature extraction, connectivity evaluation, as well as to exploit the pattern recognition [19,20].

Ultimately, a graph convolution can be generalized from a 2-dimensional (2D) convolution, and if an image is analyzed as a special case of a graph, then the latter may also be analyzed by taking into account its adjacent matrix; the only detail is to be able to relate the nodes of the graph through a system of weights that are representative of the characteristics of each node.

According to this statement, we modeled a GNN based on graph convolutions making use of multiple convolutional layers, and a point-wise non-linearity function following each

layer; in detail, we used the rectified linear unit (*ReLU*) activation function. Fitting the input data source to the model required a preprocessing step to learn the weight matrix from its topology. The embeddings were computed for the input network built on a set of GDAs in order to map each node into a vector of real numbers in a multidimensional space, as well as to train our model. Subsequently, we used it to identify the existence of candidate links (i.e., GDAs) among the entities of the original network. The experimentation was conducted by using data in the DisGeNET database, while the DiseaseGene Association Miner (DG-AssocMiner) dataset by Stanford's BioSNAP was also processed for performance evaluation only. Briefly, our contribution can be summarized as follows:

- Designing a GNN able to infer candidate GDAs by training an initial set of well-known and curated relationships between genes and diseases, modeled as a network;
- Embedding a biological network consisting of GDAs, based on its own topological features;
- Identifying candidate GDAs by exploiting link prediction via GNN.

The rest of the paper is organized as follows. Section 3 presents the design and implementation of our solution. Section 4 describes the model evaluation, and the explanatory tests for link prediction related to GDAs. Section 5 discusses the results. Finally, Section 6 concludes the paper.

2. Related Works

In this section, we report a set of solutions (i.e., methodologies, models, software tools and prototypes) based on GNNs. As introduced, GNNs represent a significant approach for graph-based studies. These may be applied to predict biological objects and/or their interaction in several fields, such as protein–protein interaction (PPI), gene interaction, drug–target interaction (DTI), and chemical stability prediction, as well as to exploit the topological structures in biological networks [21–23].

Wan et al. [24] proposed an inductive graph aggregator-based framework able to predict potential compound–protein interactions. The authors built a homogeneous graph starting from a compound–protein heterogeneous graph by integrating the ligand-based protein representations and overall similarity associations. They based the training process on low-dimensional node embeddings.

Similarly, Li et al. [25] developed a GNN model for diagnosis prediction. Authors retrieved the information for training on a graph built on temporal electronic health record data. A graph convolutional network (GCN) was used for the mentioned solutions.

In the past few years, GCNs have been widely used to develop several variants of GNNs [26]. They compute the input neurons with a set of weights (i.e., filters or kernels) that are exploited by using a sliding window to learn the features of interest for any neighboring nodes. This approach was originally designed for image analysis, in which a window slides across an image to acquire subsamples for object detection [27,28].

For instance, Li et al. [29] proposed a GCN for COVID-19 drug design. Authors based their own model on a multi-physical molecular graph representation by embedding various atom interactions in element-specific graph representations. They computed the embedding by only taking into account the distance-related node features, in order to overcome the issues related to feature extraction or generation. The model was trained by using data from several versions of the protein data bank binding (PDBbind) database.

A similar drug–side effects prediction was performed by Yu et al. [30] via a hybrid embedding GNN model, which integrates both graph-embedding and node-embedding modules. Therefore, the model was trained on a dual set of features, simultaneously.

Always resorting to using a GCN as a basis for their own model, Zhang et al. [31] proposed a *signed* GNN to predict deregulation types of miRNA–disease associations. The latter were processed as a signed bipartite network by taking into account both miRNA and disease nodes, as well as two types of edges representatives for a miRNA–disease association, in which the former was up- or down- regulated. The authors built their own model

by learning the topological features of the input graph, through a node labeling computed on miRNA–miRNA functional similarity and disease–disease semantic similarity.

This issue has also been studied by Gao et al. [32]. Also in this case, the similarity was computed via node embeddings; however, the authors implemented a link prediction task to overcome the insufficient number of labeled similar disease pairs.

Kang et al. [33] applied a link prediction task to four biological network data (i.e., lncRNA–disease association, miRNA–disease association, protein–protein interaction, and drug–drug interaction), by including a GCN–encoder layer to improve the multi-link discrimination capabilities.

Finally, we investigated several issues addressed through the use of GNN in biological networks. One detail that is immediately apparent is that the mentioned solutions are based on GCNs. According to Zhang et al. [34], the reason could lie in the fact that GCN is the simplest model with fewer instructions and floating-point operations per second (FLOPS), which makes it the most commonly used architecture in real-life applications.

3. Materials and Methods

We designed a solution based on a GNN in order to identify candidate GDAs by training our model on the basis of an initial set of well-known and curated relationships among genes and diseases modeled as a network (i.e., DisGeNET). The model was built making use of convolutional layers, and an activation function based on *ReLU*; the latter is a point-wise non-linearity function following each layer. Furthermore, we mapped each node into a vector of real numbers in a multidimensional space by performing the node embeddings of the input network built on a set of GDAs. Each embedding is symptomatic for the features of the related node, obtained by evaluating the connections with its neighbors.

Node embeddings are used to train our model, and subsequently to predict the existence of candidate links (i.e., GDAs). In addition, fitting the original data source to the model requires a preprocessing step to learn the weight matrix based on the original graph topology.

Summarizing, we mapped the input graph into a lower-dimensional space (i.e., embedding) by using the model's encoder, and a decoder to reconstruct the input graph from the embeddings via a dot product operator. Let us denote with A the adjacency matrix computed from the input graph, and with A' as the reconstructed adjacency matrix from the embedding. Our solution works by minimizing the loss function calculated on the difference between A and A'.

3.1. Datasets

We briefly describe the specifications related to the datasets used by our solution for the testing and the performance evaluation. Their own data sources (i.e., DisGeNET and Stanford's BioSNAP) are already presented in Section 1.

3.1.1. DisGeNET (v7.0, Update 2020)

The last release of DisGeNET (v7.0, https://www.disgenet.org/dbinfo, accessed on 19 May 2023) contains 1,135,045 GDAs. It is provided as a SQLite database, which also includes variant–disease associations (VDAs) and disease–disease associations, as well as additional information about the biological objects involved within the associations (i.e., genes, diseases and variants).

The data are grouped into several sets (or versions), of which we used the "Curated" and "All" versions; the former refers to well-known GDAs related to humans (Homo sapiens), while the latter also includes inferred associations computed by integrating data derived from the text mining of the scientific literature and no-human repositories (e.g., MGI-Mouse Genome Informatics Database, and Rat Genome Database).

We report below the statistics for the two graphs modeled starting from the DisGeNET database.

All-version:

- Total nodes: 37,444;
- Gene nodes: 17,074;
- Disease nodes: 20,370;
- Edges: 1,122,238.

Curated-version:

- Total nodes: 20,924;
- Gene nodes: 9738;
- Disease nodes: 11,186;
- Edges: 168,992.

3.1.2. DG-AssocMiner

In addition, we used a secondary dataset to evaluate our model on a different source. The latter is also related to the context concerning the GDAs. In detail, we retrieved the DG-AssocMiner provided by Stanford's BioSNAP; this is available at https://snap.stanford.edu/biodata/datasets/10012/10012-DG-AssocMiner.html, accessed on 19 May 2023).

- Total nodes: 7813;
- Gene nodes: 7294;
- Disease nodes: 519;
- Edges: 21,357.

In this work, we strictly focused on DisGeNET, being composed of a larger dataset; we used DG-AssocMiner to provide a secondary feedback on the performance evaluated on the former. Results are reported in Section 4.

3.2. Preprocessing

We extracted all GDAs from DisGeNET by excluding additional information (e.g., aliases for genes and diseases, source used to identify an association, and other identifiers for external databases); the latter will be able to be integrated in post-processing, eventually.

Furthermore, we encoded the gene and disease identifiers in 32-bit floating point, according to the requirements of tensors (i.e., data structures used in linear algebra). Subsequently, we modeled an undirected, bipartite graph, where the nodes represent both genes and diseases, while the edges represent the GDAs. We repeated this step both for the "Curated" and "All" versions, in order to produce two independent graphs. The former was used for model training, validation and testing, while the latter as a reference for the identified GDAs.

We treated the GDAs as pairs $(gene, disease)$. Let G be a set of genes and D be a set of diseases, such that $G = [g_1, g_2, \ldots, g_n]$ and $D = [d_1, d_2, \ldots, d_m]$, with n and m, respectively, the size of G and D. Formally, the cross-referencing may be computed to build the following domain:

$$\forall g \in G \; \exists d \in D : f(g, d) \rightarrow GDA$$

In the proposed solution, the pre-processing is performed automatically by a dedicated module which receives the following inputs: an edge list (i.e., GDAs), and two sets, from which to extract node information for gene and disease, respectively.

3.3. Graph Neural Network Architecture

We designed a GNN architecture based on graph convolution. The latter is a well-studied mathematical operator behind most GNN architectures [35]. As discussed before in Sections 1 and 2, this approach considers the graph similar to an image, taking into account a generalization of this one, where each node is evaluated in reference to a set of adjacent neighbors, such as the pixels in an image. Therefore, the nodes in a specific layer are evaluated based on the neighbors' features computed in the previous one, and these are transformed in a set of embedding vectors representing the related latent space.

We applied our model to link classification. Let us denote with $G = (V, E)$ a given undirected graph, where $V = [v_1, v_2, v_{...}, v_n]$ is the set of nodes of size n, and $E = [e_1, e_2, e_{...}, e_m]$ the set of edges of size m. For a node v_i (with $0 \leq i \leq n$), we can denote its features in the latent space as the vector h_i such that the latter is computed by a function f; formally:

$$f : v_i \rightarrow h_i$$

Therefore, the whole set of neighbor's features for v_i can be represented through a tensor, where each row is the vector resulting from the adjacent neighbor's feature. Formally, we can describe the function g for the evaluation of a link between two nodes v_i and v_j (with $0 \leq j \leq n$) as follows:

$$\begin{gathered} h_i = f(v_i) \\ h_j = f(v_j) \\ s_{ij} = fe(h_i, h_j, w_{ij}) \end{gathered}$$

where w_{ij} is the weight (normalized to $[0 - 1]$ range) on the link between v_i and v_j; note that w_{ij} will be 1 for all edges, if the graph is unweighted.

We used *ReLU* as the activation function following each layer, and *Adam* [36] for the first-order gradient-based optimization of the stochastic objective function. Formally, *ReLU* can be defined as follows:

$$ReLU(x) = max(0, x)$$

Furthermore, we integrated an *encoder* within our model (i.e., *auto-encoder*), in order to compute the node embeddings by processing the input graph via multiple convolutional layers, based on the graph convolutional operator proposed by Kipf et al. [37].

We report the main specifications of the proposed model as follows:

- Layers: 2 convolutional layers;
- Activation function: *ReLU*;
- Dropout: $p = 0.5$;
- Feature Size: 50;
- Decoder: *inner product decoder*.

Briefly, *dropout* refers to a simple way to prevent a NN from overfitting [38], by dropping out the nodes, both in the input and hidden layers; p is the probability of an element to be zeroed. The *dropout* randomly zeroes a set of elements within the input tensor; in detail, during the training step, it scales the outputs by a factor $1/(1 - p)$, computing the identity function. According to Srivastava et al. [39], dropping a neuron with $p = 0.5$ obtains the highest variance for the probability distribution of a NN architecture, for a wide variety of use cases (including ours).

3.4. Link (GDA) Prediction

The proposed solution uses a *Decoder*, to perform the link predictions via binary classification. It needs a set of negative links that are randomly included into the validation and testing sets; these are excluded from the training set to evaluate only the original graph structure.

The link prediction is performed by computing the dot product (or scalar product) of the node embeddings for all pair of nodes, and by evaluating the probability of edge existence as the resulting score from the aggregation of all embeddings for each pair of nodes. Therefore, a candidate link between a pair of nodes (v_i, v_j) is evaluated in reference to the calculated dot product between their respective embeddings. We calculated the dot products via the Einstein summation convention on the operands, for tensor contractions.

Resulting candidate links can be discriminated by applying a threshold, or by extracting a defined number among those with the highest value. Since a threshold is not defined in the literature, we opted for an empirical evaluation.

Figure 1 shows a non-exhaustive workflow of the proposed solution.

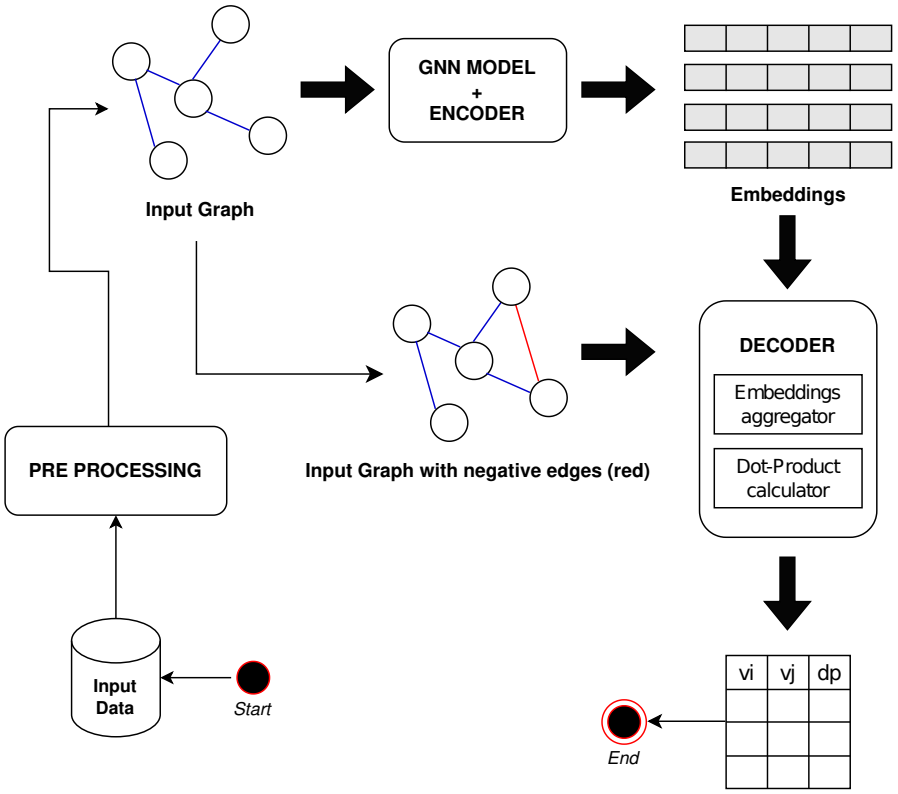


Figure 1. A non-exhaustive workflow of the proposed solution is reported.

3.5. Evaluation Criteria

We assessed the overall effectiveness of the proposed model, by employing two well-known key performance indicators (KPI): average precision (AP), and area under the receiver operating characteristic (ROC) curve (i.e., AUC or AUROC) [40].

The AP represents a precision–recall curve through a weighted average value of precision. It is defined as follows:

$$AP = \sum_n (Recall_n - Recall_{n-1}) \times Precision_n$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}, \text{ and}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

The ROC curve shows the relationship between sensitivity and specificity, for every possible cut-off. It takes into account the true positive rate (TPR) and false positive rate (FPR). It is possible to detect a unique threshold value for each local maximum in the difference curve. It will be the sensitivity parameter useful to measure how many flat points we should expect before producing a knee/elbow, into the plot [41–43]. A larger sensitivity value will detect a more conservative threshold. We evaluated the best threshold (or *Best*, in the figures) via Youden's index (or *J*). It will be the maximum value of *J*, among all ones calculated for each point of the ROC curve [44]. Formally, *J* is defined as follows:

$$J = sensitivity + specificity - 1$$

with $sensitivity = true\ positives/(true\ positives + false\ negatives)$, and $specificity = true\ negatives/(true\ negatives + false\ positives)$.

Of significant interest is the evaluation of the AUC computed on the ROC curve. It is a performance measurement used to evaluate classification models with different settings. Therefore, we used it to evaluate the model in terms of accuracy. Formerly, it is defined as follows:

$$AUC = \sum TPR\Delta FPR; \text{ where}$$

$$TPR = \frac{|True\ Positives|}{|True\ Positives \cup False\ Negatives|}, \text{ and}$$

$$FPR = \frac{|False\ Positives|}{|False\ Positives \cup True\ Negatives|}$$

3.6. Implementation and Deployment

We implemented the proposed solution in Python (version 3), by using the Graph Auto-Encoder (GAE), belonging to the PyTorch framework [45]. GAE is an end-to-end trainable neural network model for unsupervised learning on graph-structured data.

Table 1 reports a list of the main PyTorch's modules used in the implementation of the proposed solution.

Table 1. Main PyTorch's modules used in the implementation of the proposed solution.

Name	Description	Documentation
PyTorch Geometric (PyG)	It consists of several methods for geometric deep learning.	https://pytorch-geometric.readthedocs.io (accessed on 19 May 2023)
Sparse	It allows optimized sparse matrix operations on tensor.	https://pytorch.org/docs/stable/sparse.html (accessed on 19 May 2023)
NN	It provides several NN layers (e.g., convolutional layers).	https://pytorch.org/docs/stable/nn.html (accessed on 19 May 2023)
Tensor	It allows handling tensors.	https://pytorch.org/docs/stable/tensors.html (accessed on 19 May 2023)

Note that our implementation is optimized to be executed on a graphics processing unit (GPU).

We deployed our solution on Google Colaboratory (Colab; https://colab.research.google.com, accessed on 19 May 2023). To date, Colab (free plan) allows connecting to a session with the following specifications:

- CPU: Intel Xeon CPU @2.20 GHz;
- Memory: 12 GB;
- GPU: Tesla K80.

Note that Colab (free plan) could limit the session according to both the time of use and the workload of the server hosting it.

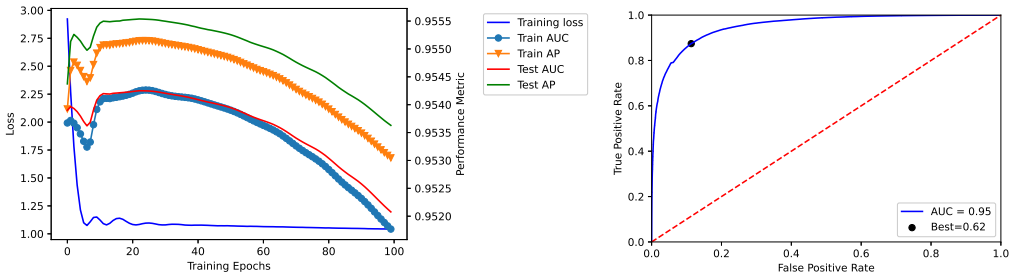
4. Results

We tested our solution by using the DisGeNET database as source, using both the *Curated*- and *All*-versions. The former was used to train and to evaluate the model in terms of AUC and AP, as well as to identify a set of candidate GDAs in a real use case (i.e., link prediction). The latter was used to evaluate our predictions by using no-human resources. We removed from the *All*-version the GDAs included in the *Curated*-version, in order to eliminate redundancy. Furthermore, we investigated scientific literature to evaluate a probable truthfulness (or hypothetical match) from recent studies, for the candidate GDAs not available in the *All*-version of DisGeNET.

Finally, we applied the proposed solution on a secondary dataset, to evaluate the model performance on a different data source as well as to corroborate the results obtained by processing the DisGeNET datasets.

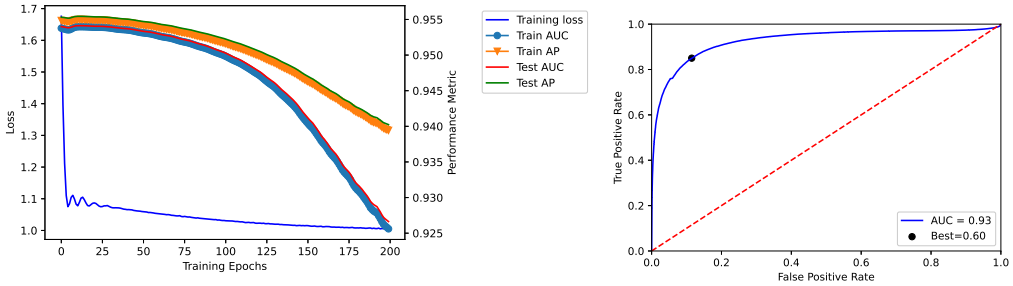
4.1. Model Performance

We trained the proposed model throughout 100, 200, and 300 epochs, by showing the results in Figures 2–4, respectively. In addition, we evaluated empirically the optimal number of features by performing the training, validation and testing throughout 100 epochs (see Table 2); as shown, it obtained the best result in terms of AUC.



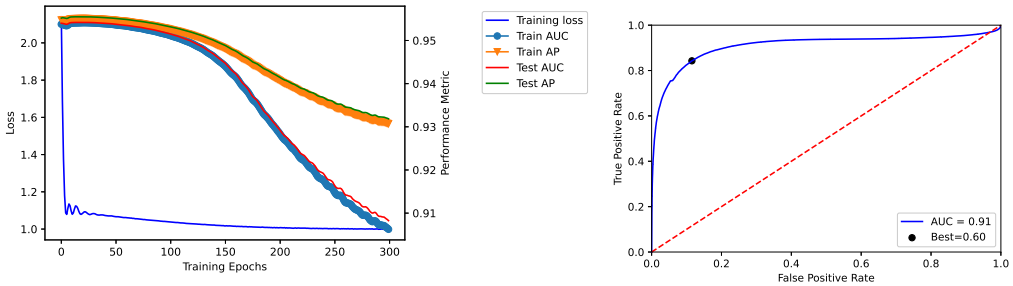
(**a**) (**b**)

Figure 2. Dataset: DisGeNET. Training, validation and testing throughout 100 epochs; dataset: *Curated*-version. (**a**) Loss curve across the training process over the epochs. In addition, TPR (*Sensitivity*) and FPR ($1 - Specificity$) were related through the ROC curve in (**b**) by also reporting the related AUC and best threshold (i.e., "Best").



(**a**) (**b**)

Figure 3. Dataset: DisGeNET. Training, validation and testing throughout 200 epochs; dataset: *Curated*-version. (**a**) Loss curve across the training process over the epochs. In addition, TPR and FPR were related through the ROC curve in (**b**), by also reporting the related AUC and best threshold (i.e., "Best").



(**a**) (**b**)

Figure 4. Dataset: DisGeNET. Training, validation and testing throughout 300 epochs; dataset: *Curated*-version. (**a**) shows the loss curve across the training process over the epochs. In addition, TPR and FPR were related through the ROC curve in (**b**), by also reporting the related AUC and best threshold (i.e., "Best").

Based on the same specification and configuration, we also tested our solution on DG-AssocMiner to evaluate the performance in a different dataset and to corroborate our results. The resulting AUC and best threshold is shown in Table 3.

Table 2. Dataset: DisGeNET. Training, validation and testing throughout 100 epochs, by incremental number of features. Bold font indicates the best result.

No. of Features	AUC	Best Threshold
25	0.95	0.62
50	**0.95**	**0.63**
75	0.95	0.62
100	0.95	0.61

Table 3. Dataset: DG-AssocMiner. Training, validation and testing throughout 100 epochs, by incremental number of features. Bold font indicates the best result.

No. of Features	AUC	Best Threshold
25	0.95	0.60
50	**0.95**	**0.62**
75	0.95	0.60
100	0.95	0.61

We performed an independent samples *t*-test, to detect mean differences in "Best Threshold" between the two presented datasets. The two-tailed *p*-value equals 0.0106; by conventional criteria (i.e., *p*-value < 0.05) it can be considered statistically significant. The AUC between the two datasets is the same for all experiments; therefore, we cannot analyze perfect data with a *t*-test, and the statistical test on this parameter was omitted.

4.2. Identification of Candidate GDAs

Firstly, we studied the *Curated*-version of DisGeNET by performing a node-level descriptive statistics of the resulting network; Table 4 shows the 10 genes and 10 diseases with the highest degree. These should be among the most recurring in the next test.

Table 4. Node-level descriptive statistics for the network built from DisGeNET (*Curated*-version). We reported the 10 genes and 10 diseases with the highest degree.

Gene Name	Node Degree	Disease UMLS	Disease Description	Node's Degree
TNF	340	C0006142	Malignant neoplasm of breast	1074
SOD2	285	C0036341	Schizophrenia	1031
IL6	270	C0023893	Liver Cirrhosis, Experimental	774
POMC	241	C0009402	Colorectal Carcinoma	702
PTGS2	239	C0376358	Malignant neoplasm of prostate	616
TP53	232	C0033578	Prostatic Neoplasms	616
IL1B	231	C0678222	Breast Carcinoma	538
MTHFR	192	C0005586	Bipolar Disorder	536
NOS2	184	C1458155	Mammary Neoplasms	527
PTEN	182	C4704874	Mammary Carcinoma, Human	525

Subsequently, we applied our solution to the described network, to identify a set of candidate GDAs. The top-15 ones (highest dot product) are reported in Table 5. In addition, we correlated these with the GDAs inferred by DisGeNET in its own *All*-version, by integrating no-human datasets. We removed from the *All*-version the GDAs included in the *Curated*-version in order to eliminate redundancy.

Where a match with DisGeNET *All*-version was not found, we reported the PubMed IDentifier (PMID) of the articles, which may corroborate the candidate GDA of interest. In other words, for each predicted GDA not present in DisGeNET, we manually investigated the literature to find a scientific article that reports biological evidence of that GDA.

Table 5. Top-15 candidate GDAs (highest dot product) predicted by our solution, of which 47% (7/15) matched DisGeNET (*All*-version). Where a match with DisGeNET *All*-version was not found, the PMID of an article highlighting a possible match was found; we marked a GDA as *untraceable* when no match was found.

Gene	Disease		Reference
	UML	Description	
TNF	C0040517	Gilles de la Tourette syndrome	PMID: 25256363
TNF	C0006413	Burkitt Lymphoma	**DisGeNET (All)**
TNF	C0079588	Ichthyosis, X-Linked	**DisGeNET (All)**
SOD2	C0040517	Gilles de la Tourette syndrome	PMID: 31468582 (SOD, enzyme)
SOD2	C0006413	Burkitt Lymphoma	PMID: 28483518
SOD2	C0079588	Ichthyosis, X-Linked	PMID: 28540003 (*SOD1)
TNF	C0009404	Colorectal Neoplasms	**DisGeNET (All)**
POMC	C0040517	Gilles de la Tourette syndrome	*untraceable*
POMC	C0006413	Burkitt Lymphoma	PMID: 29296973
SOD2	C0009404	Colorectal Neoplasms	**DisGeNET (All)**
IL6	C0040517	Gilles de la Tourette syndrome	PMID: 35087475
IL6	C0006413	Burkitt Lymphoma	**DisGeNET (All)**
TP53	C0040517	Gilles de la Tourette syndrome	**DisGeNET (All)**
POMC	C0079588	Ichthyosis, X-Linked	PMID: 22289416
TP53	C0006413	Burkitt Lymphoma	**DisGeNET (All)**

5. Discussion

The model related to the proposed solution reported very satisfactory performance. According to the loss curves (training loss, train AUC, train AP, test AUC, and test AP) as reported in Figures 2a, 3a, and 4a, the proposed model was able to identify positive links with high sensibility, without interference from the negative links and by reporting a low number of false positives. The related values of AUC are reported in Figures 2b, 3b, and 4b.

In our tests, AUC values demonstrate what was observed. The average value of AUC is 0.93, and it refers to a number of correct predictions equal to 93% of the total ones. In addition, the results report an AUC of 0.95 for the number of epochs (100) our model was ultimately trained on; the same experiments replicated on the DS-AssocMiner dataset produced a performance evaluation of 0.95 in terms of AUC. Therefore, 95% of predictions can be considered correct for the final model, being that the ones obtained for 200 and 300 epochs were discarded.

Summarizing, the proposed solution reported an excellent value of AUC in all tests. According to Nahm et al. [46], the interpretation of AUC may be addressed as follows:

- $AUC \geq 0.9$: Excellent;
- $0.8 \leq AUC < 0.9$: Good;
- $0.7 \leq AUC < 0.8$: Fair;
- $0.6 \leq AUC < 0.7$: Poor;
- $0.5 \leq AUC < 0.6$: Fail;
- $AUC < 0.5$: Incorrect.

It is possible to note that once the 100 epochs are exceeded, the model begins to show overfitting. Furthermore, the empirical evaluation of the optimal number of features reports 50 as the optimal value to obtain the best results in terms of AUC and best threshold. As discussed, in the final model, we defined 100 and 50 as values for the number of epochs and features, respectively.

In addition, we tested a real use case, proceeding to identify the Top-15 candidate GDAs (highest dot product) inferred by the proposed solution (Table 5). We evaluated an existing or probable relationship between these and what was inferred by the DisGeNET (*All*-version purged of redundancies with the *Curated*- one). We also investigated the scientific literature by using the Pubmed search engine (https://pubmed.ncbi.nlm.nih.gov/, accessed on 19 May 2023), provided by the National Center for Biotechnology Information (NCBI), by reporting in Table 5 the Pubmed ID (PMID) of the articles that corroborate the possible correlations. Note that the PMID is reported only where a match with DisGeNET *All*-version was not found since the latter is already validated.

According to Table 5, we found a 47% (7/15) match between the GDAs already validated by the DisGeNET *All*-version, and the candidate ones predicted by our solution. Of the remaining 8 candidate GDAs, an existing or probable relationship was found for 7/8 of these by studying the scientific literature. Definitely, 14/15 candidate GDAs, 93.3% (14/15), had a positive match.

Deepening the study of our Top-15 candidate GDAs, we can see that the associations are referable to the following genes and diseases:

- Genes: TNF, SOD2, POMC, IL6, and TP53.
- Diseases UMLS (and Description): COO40517 (Gilles de la Tourette syndrome), C0006413 (Burkitt Lymphoma), C0079588 (Ichthyosis, X-Linked), and C0009404 (Colorectal Neoplasms).

For instance, *TNF* is a protein coding gene, which encodes a multifunctional proinflammatory cytokine, belonging to the tumor necrosis factor (i.e., *TNF*) super-family. The DisGeNET *All*-version reports 2/3 associations inferred by our solution; in detail, it reports an explicit association with *C0006413* and *C0009404*. Furthermore, the association with *C0009404* could be inferred from *PMID:25256363* [47], where Keszler et al. investigated two TNF promoter polymorphisms on the genetic susceptibility to Tourette syndrome (TS); the authors concluded by reporting: "Based on these findings, the *TNF* -308 G-allele can be associated with Tourette syndrome". Similarly, the mentioned genes have been studied in the literature to evaluate the candidate associations inferred by our solution, but have not yet been included in DisGeNET.

Furthermore, we showed a greater recurrence of the nodes having a high-level degree (Table 4) among our candidate GDAs. This result was expected, in that the GCNs, as well as the GNNs making use of convolutional layers, express the accuracy by also taking into account the node's degree. Indeed, from empirical observations, a GNN embeds the features with more accuracy during the training and testing, for nodes with the highest degree, even if these are under-represented [48].

Summarizing, the results showed an AUC of 95% for training, validation and testing throughout 100 epochs, which, in the real case, translated into a positive response for 93% of the candidate GDAs predicted by our solution.

6. Conclusions

GNN allows using a NN for graph data modeling, with relevant advantages in the context of node classification, link prediction and graph classification. In this paper, we presented a solution for GDAs prediction based on GNN. Our model was based on graph convolutions and node embeddings. The latter were computed for the input network built on a set of GDAs, in order to map each node into a vector of real numbers in a multidimensional space. Our tests were conducted by using the DisGeNET *Curated*-version for model training, while we used the *All*-version in conjunction with the literature to evaluate the candidate GDAs identified in a real use case. Results show an excellent AUC

(95% throughout 100 epochs) for training, validation and testing throughout 100 epochs, which in the real case translated into a positive response for 93% of the Top-15 (highest dot product) candidate GDAs identified by our solution. Briefly, our results may be summarized as follows: we found a 47% match (7/15) between the GDAs already reported in the literature by the DisGeNET *All*-version, and the candidate ones identified by our solution; of the remaining 8, an existing or probable relationship was found for 7/8 of these by studying the scientific literature.

Future directions related to GDA are mainly focused on inferring novel associations, as well as how to handle (e.g., store and visualize) the relevant data both in terms of accessibility and scalability. The study of GDAs has an important impact on investigating novel associations between genes (or variants) and diseases. It provides a relevant effort for improving the knowledge for disease etiology [49], being an area of active research, for which in-depth studies are essential, particularly focused on the validation of the predictions made by bioinformatics methodologies, e.g., based on machine learning and deep learning.

Author Contributions: Data curation, P.C.; Formal analysis, P.C.; Investigation, P.C. and M.C.; Writing—Original Draft Preparation, P.C.; Writing—Review and Editing, P.C. and M.C.; Methodology, P.C.; Supervision, M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Gene–Disease Associations used for training and testing our model during the current study are available on DisGeGNET database (https://www.disgenet.org, accessed on 19 May 2023). Results are available on our own GitHub Repository (https://github.com/pietrocinaglia/gnn_gda, accessed on 19 May 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Anderson, A.N.; Roncaroli, F.; Hodges, A.; Deprez, M.; Turkheimer, F.E. Chromosomal profiles of gene expression in Huntington's disease. *Brain* **2008**, *131*, 381–388. [CrossRef]
2. Chen, X.; Wang, L.; Qu, J.; Guan, N.N.; Li, J.Q. Predicting miRNA-disease association based on inductive matrix completion. *Bioinformatics* **2018**, *34*, 4256–4265. [CrossRef] [PubMed]
3. Chen, X.; Yin, J.; Qu, J.; Huang, L. MDHGI: Matrix Decomposition and Heterogeneous Graph Inference for miRNA-disease association prediction. *PLoS Comput. Biol.* **2018**, *14*, e1006418. [CrossRef]
4. Huang, L.; Zhang, L.; Chen, X. Updated review of advances in microRNAs and complex diseases: Taxonomy, trends and challenges of computational models. *Brief Bioinform.* **2022**, *23*, bbac397. [CrossRef] [PubMed]
5. Chen, X.; Xie, D.; Zhao, Q.; You, Z.H. MicroRNAs and complex diseases: From experimental results to computational models. *Brief Bioinform.* **2019**, *20*, 515–539. [CrossRef] [PubMed]
6. Cinaglia, P.; Guzzi, P.H.; Veltri, P. INTEGRO: An algorithm for data-integration and disease-gene association. In Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Madrid, Spain, 3–6 December 2018; pp. 2076–2081. [CrossRef]
7. Piñero, J.; Saüch, J.; Sanz, F.; Furlong, L.I. The DisGeNET cytoscape app: Exploring and visualizing disease genomics data. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 2960–2967. [CrossRef]
8. Pinero, J.; Bravo, A.; Queralt-Rosinach, N.; Gutierrez-Sacristan, A.; Deu-Pons, J.; Centeno, E.; Garcia-Garcia, J.; Sanz, F.; Furlong, L.I. DisGeNET: A comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Res.* **2017**, *45*, D833–D839. [CrossRef]
9. Pinero, J.; Queralt-Rosinach, N.; Bravo, A.; Deu-Pons, J.; Bauer-Mehren, A.; Baron, M.; Sanz, F.; Furlong, L.I. DisGeNET: A discovery platform for the dynamical exploration of human diseases and their genes. *Database* **2015**, *2015*, bav028. [CrossRef]
10. Thomas, P.; Rocktaschel, T.; Hakenberg, J.; Lichtblau, Y.; Leser, U. SETH detects and normalizes genetic variants in text. *Bioinformatics* **2016**, *32*, 2883–2885. [CrossRef]
11. Smigielski, E.M.; Sirotkin, K.; Ward, M.; Sherry, S.T. dbSNP: A database of single nucleotide polymorphisms. *Nucleic Acids Res.* **2000**, *28*, 352–355. [CrossRef]
12. Consortium, T.U. UniProt: The Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.* **2022**, *51*, D523–D531. [CrossRef] [PubMed]
13. Leskovec, J.; Krevl, A. *SNAP Datasets: Stanford Large Network Dataset Collection*; Stanford University: Stanford, CA, USA, 2014.

14. Cinaglia, P.; Cannataro, M. Network alignment and motif discovery in dynamic networks. *Netw. Model. Anal. Health Inform. Bioinform.* **2022**, *11*, 38. [CrossRef]
15. Chow, K.; Sarkar, A.; Elhesha, R.; Cinaglia, P.; Ay, A.; Kahveci, T. ANCA: Alignment-based Network Construction Algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *18*, 512–524. [CrossRef] [PubMed]
16. Cinaglia, P.; Cannataro, M. Forecasting COVID-19 epidemic trends by combining a neural network with Rt estimation. *Entropy* **2022**, *24*, 929. [CrossRef] [PubMed]
17. Elhesha, R.; Sarkar, A.; Cinaglia, P.; Boucher, C.; Kahveci, T. Co-evolving Patterns in Temporal Networks of Varying Evolution. In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Bcb '19, New York, NY, USA, 7–10 September 2019; pp. 494–503. [CrossRef]
18. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural. Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef]
19. Harada, S.; Akita, H.; Tsubaki, M.; Baba, Y.; Takigawa, I.; Yamanishi, Y.; Kashima, H. Dual graph convolutional neural network for predicting chemical networks. *BMC Bioinform.* **2020**, *21*, 94. [CrossRef]
20. Yang, H.; Zhuang, Z.; Pan, W. A graph convolutional neural network for gene expression data analysis with multiple gene networks. *Stat. Med.* **2021**, *40*, 5547–5564. [CrossRef]
21. Zhang, Z.; Chen, L.; Zhong, F.; Wang, D.; Jiang, J.; Zhang, S.; Jiang, H.; Zheng, M.; Li, X. Graph neural network approaches for drug-target interactions. *Curr. Opin. Struct. Biol.* **2022**, *73*, 102327. [CrossRef]
22. Zeng, Y.; Wei, Z.; Pan, Z.; Lu, Y.; Yang, Y. A robust and scalable graph neural network for accurate single-cell classification. *Brief Bioinform.* **2022**, *23*, bbab570. [CrossRef]
23. Kim, Y.; Jeong, Y.; Kim, J.; Lee, E.K.; Kim, W.J.; Choi, I.S. MolNet: A Chemically Intuitive Graph Neural Network for Prediction of Molecular Properties. *Chem. Asian J.* **2022**, *17*, e202200269. [CrossRef]
24. Wan, X.; Wu, X.; Wang, D.; Tan, X.; Liu, X.; Fu, Z.; Jiang, H.; Zheng, M.; Li, X. An inductive graph neural network model for compound-protein interaction prediction based on a homogeneous graph. *Brief Bioinform.* **2022**, *23*, 1–13. [CrossRef] [PubMed]
25. Li, Y.; Qian, B.; Zhang, X.; Liu, H. Graph Neural Network-Based Diagnosis Prediction. *Big Data* **2020**, *8*, 379–390. [CrossRef] [PubMed]
26. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
27. Jia, S.; Jiang, S.; Zhang, S.; Xu, M.; Jia, X. Graph-in-Graph Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Neural. Netw. Learn. Syst.* **2022**, 1–15. [CrossRef]
28. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural. Netw. Learn. Syst.* **2022**, *33*, 6999–7019. [CrossRef]
29. Li, X.S.; Liu, X.; Lu, L.; Hua, X.S.; Chi, Y.; Xia, K. Multiphysical graph neural network (MP-GNN) for COVID-19 drug design. *Brief Bioinform.* **2022**, *23*, bbac231. [CrossRef]
30. Yu, L.; Cheng, M.; Qiu, W.; Xiao, X.; Lin, W. idse-HE: Hybrid embedding graph neural network for drug side effects prediction. *J. Biomed. Inform.* **2022**, *131*, 104098. [CrossRef]
31. Zhang, G.; Li, M.; Deng, H.; Xu, X.; Liu, X.; Zhang, W. SGNNMD: Signed graph neural network for predicting deregulation types of miRNA-disease associations. *Brief Bioinform.* **2022**, *23*, bbab464. [CrossRef]
32. Gao, J.; Zhang, X.; Tian, L.; Liu, Y.; Wang, J.; Li, Z.; Hu, X. MTGNN: Multi-Task Graph Neural Network based few-shot learning for disease similarity measurement. *Methods* **2022**, *198*, 88–95. [CrossRef]
33. Kang, C.; Zhang, H.; Liu, Z.; Huang, S.; Yin, Y. LR-GNN: A graph neural network based on link representation for predicting molecular associations. *Brief Bioinform.* **2022**, *23*, bbab513. [CrossRef]
34. Zhang, Z.; Leng, J.; Ma, L.; Miao, Y.; Li, C.; Guo, M. Architectural Implications of Graph Neural Networks. *IEEE Comput. Archit. Lett.* **2020**, *19*, 59–62. [CrossRef]
35. Bianchi, F.M.; Grattarola, D.; Livi, L.; Alippi, C. Graph Neural Networks With Convolutional ARMA Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3496–3507. [CrossRef] [PubMed]
36. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980 .
37. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
38. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
39. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
40. Cinaglia, P.; Cannataro, M. Alignment of Dynamic Networks based on Temporal Embeddings. In Proceedings of the 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Las Vegas, NV, USA, 6–8 December 2022; pp. 2511–2518. [CrossRef]
41. LeDell, E.; Petersen, M.; van der Laan, M. Computationally efficient confidence intervals for cross-validated area under the ROC curve estimates. *Electron. J. Stat.* **2015**, *9*, 1583–1607. [CrossRef]
42. Satopaa, V.; Albrecht, J.; Irwin, D.; Raghavan, B. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, USA, 20–24 June 2011; pp. 166–171. [CrossRef]

43. Peterson, L.E.; Coleman, M.A. Machine learning-based receiver operating characteristic (ROC) curves for crisp and fuzzy classification of DNA microarrays in cancer research. *Int. J. Approx. Reason* **2008**, *47*, 17–36. [CrossRef]
44. Youden, W.J. Index for rating diagnostic tests. *Cancer* **1950**, *3*, 32–35. [CrossRef]
45. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
46. Nahm, F.S. Receiver operating characteristic curve: Overview and practical use for clinicians. *Korean J. Anesthesiol.* **2022**, *75*, 25–36. [CrossRef]
47. Keszler, G.; Kruk, E.; Kenezloi, E.; Tarnok, Z.; Sasvari-Szekely, M.; Nemoda, Z. Association of the tumor necrosis factor -308 A/G promoter polymorphism with Tourette syndrome. *Int. J. Immunogenet.* **2014**, *41*, 493–498. [CrossRef] [PubMed]
48. Tang, X.; Yao, H.; Sun, Y.; Wang, Y.; Tang, J.; Aggarwal, C.; Mitra, P.; Wang, S. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM20, New York, NY, USA, 19–23 October 2020; pp. 1435–1444. [CrossRef]
49. Opap, K.; Mulder, N. Recent advances in predicting gene-disease associations. *F1000Research* **2017**, *6*, 578. [CrossRef] [PubMed]