N.S.S. COLLEGE OF ENGINEERING

PALAKKAD, KERALA – 678008

UNIVERSITY OF CALICUT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAIN PROJECT REPORT
PHASE - II
2014-18

MULTILABEL NODE CLASSIFICATION ON BIOLOGICAL NETWORKS USING DEEP LEARNING

SUBMITTED BY

ATHIRA C	**NSAOECS016**
CHITHRA J	**NSAOECS019**
JAISON KURIAKOSE	**NSAOECS031**
KANNAN K	**NSAOECS033**
NADHIYA P	**NSAOECS038**

GUIDED BY

Mr. Anuraj Mohan
Assistant Professor
Dept. of Computer Science & Engineering
NSSCE, Palakkad.

N.S.S. COLLEGE OF ENGINEERING

PALAKKAD, KERALA – 678008

UNIVERSITY OF CALICUT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this is the bonafide report of the Main Project Phase - II entitled "**MULTILABEL NODE CLASSIFICATION ON BIOLOGICAL NETWORKS USING DEEP LEARNING**" done by **ATHIRA C, CHITHRA J, JAISON KURIAKOSE, KANNAN K** and **NADHIYA P** in a partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science & Engineering under University of Calicut.

GUIDE & STAFF IN CHARGE

Mr. ANURAJ MOHAN

Assistant Professor

HEAD OF DEPARTMENT

Dr. VIJI RAJENDRAN V

Associate Professor

ACKNOWLEDGEMENT

Many noble hearts have contributed immense inspiration and technical assistance for the successful completion of this main-project work.

First of all, we thank the Almighty God, for granting us the strength, courage and knowledge to start up with this project.

We acknowledge our sincere thanks to the management of N.S.S College of Engineering for their help in the successful completion of this project. We would also like to express our gratitude to the principal **Dr. Sudha T** for her support.

We take this opportunity to express our profound gratitude to the Head of the Department, **Dr. Viji Rajendran V**, for all her support towards making this project a remarkable one.

We are extremely thankful for our project guide & staff-in-charge **Prof. Anuraj Mohan** for his valuable guidance, and his dedication towards the project which helped us a lot to gain confidence and knowledge.

We also extend our sincere gratitude towards the Project Evaluation Committee members, **Prof. S Sindhu** and **Prof. Maya Mohan** and to all other teaching staffs for their valuable guidance throughout our project.

We would also like to thank all the non-teaching staff of our department for their constant encouragement throughout this project. This helped us in proper installation and demonstration of our project. Last, but not the least, we take pleasant privilege in expressing a heartfelt thanks to all our friends who were of precious help throughout this project.

SYNOPSIS

Recently, a great effort has been taken place for analyzing and processing biological networks which have strong relevance in the real world scenario. Performing recent network analysis like node classification in order to assign labels for an unlabeled node in a biological network such as PPI network, disease-disease network have strong relevance. Understanding the functions, enzymatic modification took place after synthesis, the architecture of an unlabeled node based on an already labeled node in PPI network have very much importance in different applications such as new drug development, disease diagnosis, controlling biological process and so on. Apart from this, predicting the structure of a protein helps to identify its function. Likewise, other networks also have importance in the real world scenario.

However, analyzing these types of large networks require careful effort. Because, in order to perform multi-label node classification that is predicting more than one labels of an unlabeled node effectively, requires features of each node in the network. The traditional network embedding methods have several challenges when dealing with a large network having billions of nodes. Since most of the traditional method is based iterative or combinatorial approach, embedding process requires high computational complexity for large networks. In addition to that, traditional classification methods are based on i.i.d assumptions. But, when a network is concerned, it should preserve similarities and dependencies between nodes. These methods are not scalable when the network grows large. The traditional paradigms failed to solve these problems.

Thus, the proposed system aims to apply deep learning techniques to learn feature vectors of each node effectively and map them to low dimensional space by preserving the actual network structure. Then, it performs multi-label node classification on the different biological network for predicting the labels of the unlabeled node. The system mainly focuses on applying two categories of embedding techniques such as random walk based (DeepWalk & node2vec) and non-random walk based (SDNE & DNGR) methods. Then, it aims to evaluate and compare the performance of these methods using different evaluation metrics.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER Page No.

CHAPTER 1

INTRODUCTION

Any complex biological systems can be represented in the form of network. Nodes and edges are the basic components of any network. In a biological network, the nodes represent biological units such as DNA, RNA, proteins, metabolites etc. and their interaction can be taken as edges. An understanding of these networks plays an important role in a variety of disciplines. Within the field of biology and medicine, this kinds of the network can be used for drug target identification, predicting functions of proteins or genes, providing early diagnosis of diseases and designing effective strategies for various diseases etc.

The highly important biological networks include protein-protein interaction (PPI) network, Genetic interaction network, Gene/transcriptional regulatory network and metabolic network. Protein-protein interaction network represents the physical relationship between proteins which is essential to every process taken place in a cell. Proteins are represented by nodes that are linked by directed edges. The genetic interaction is the functional relationship between different genes. In this network, genes are considered as nodes and their relationship as edges. The Gene/transcriptional regulatory network describes how gene expression is controlled. Genes and transcriptional factors represent nodes and their relationship is depicted by directed edges. Metabolic network shows how metabolites are transformed in order to produce energy and to synthesize specific substances.

1.1 MOTIVATION

These biological networks have very much significance in the real world scenario. The analysis, modeling, visualization, and study of these type of biological network play an important role in various task within in the field of biology and medicine. Protein-Protein

interaction network, Disease-Disease interaction network, and RNA-RNA interaction network are these kinds of the biological network which can be used in various disciplines. Proteins are responsible for every process taken place in the cell such as it can act as a catalyst in the biochemical reaction, maintenance of cellular environment, transporters for other substances and so on. A single protein itself has no significance. So that, the interaction between proteins can be made useful in different applications.

Understanding the functions of proteins and predicting the function of annotated proteins is useful for the development of drugs, disease diagnosis and also for the development of synthetic biochemical such as biofuels and so on. In addition to that, each protein has some other features such as their structure, sequences, post-translational modification and so on. When a protein-protein interaction network is concerned, predicting these features corresponds to each protein plays very much importance in the field of medicine. In this, post-translational modification is the enzymatic modification taken place after the synthesis of protein. This modification can be done in various ways. For example, covalent addition of functional groups, degradation of entire proteins, post-translational cleavage etc. are some the modification taken place. So that, identifying and predicting these PTM of a protein is crucial for research in cell biology, controlling biological process, disease treatment and prevention etc. Similarly, if it is possible to predict the structure of a protein based on the interaction of proteins in the PPI network, then its function can easily find out. Apart from this, one can easily predict the drug or molecule that bind to it.

Likewise, other biological networks have also a major role in a variety of disciplines. For example, predicting the organ which is going to be affected by a disease and finding out the number of disorder class from a disease-disease interaction network can be done using the various machine learning algorithm. These help for early disease diagnosis and for designing effective strategies for disease treatment. Also, the protein that relates to each RNA can be able to determine from the RNA-RNA interaction network and RNA-protein network.

Node classification or predicting these type of features of an unlabeled node in the biological network by considering already labeled node is very difficult and tedious task. Because, as far as a large network like the biological network is concerned, the network analysis and processing is a challenging task. In order to work on this network, we need to extract the features of each node in the network or learn the network representation effectively.

But, learning this representation from a large network requires careful effort. The traditional network representation learning techniques face several challenges when dealing with a large network having billions of nodes. Most of the traditional methods use iterative or combinatorial approach for learning this representation. But, it is not applicable for a large network, since it results in high computational complexity. When a network is concerned, there exists some similarity and dependencies between nodes. That is, in a network, some of the nodes are having homophily (belongs to same community) and structural equivalence (same structural role) similarities. But, the traditional machine learning methods assume that node is independent and identically distributed (i.i.d assumptions). So that, the traditional methods cannot preserve these similarities and network structure when the nodes are mapped into a vector space. Since the data volume of all the networks grows exponentially and data is linked in real-world cases, the traditional methods are not scalable for learning representation from a large network. In this scenario, deep learning gains more importance. As the deep learning method for node embedding such as DeepWalk, node2vec make use of random walk procedure, it is able to preserve similarity and dependencies between nodes. That is, the proposed system aims to apply deep learning techniques to learn network representation in such a way that actual network structure is preserved.

1.2 PROBLEM DEFINITION

The proposed system aims to find an efficient method for multi-label node classification in a biological network using deep learning techniques. Node classification or graph labeling is the process of assigning labels to the unlabeled nodes by considering the labels of its neighbors. As far as a biological network is concerned it is likely to have more than one label for each node. Hence, it becomes a multi-label classification problem in which there will be more than one features needs to be predicted for a single node. So, in the context of a biological network, the nodes represent biological units such as protein, RNA, DNA etc. and the interaction between them is considered as edges. The proposed system aims to analyze three different kinds of biological network such as protein-protein interaction network, disease-disease network, and RNA-RNA interaction network. When we consider a protein-protein interaction network, proteins are taken as nodes and the features such post-translational

modification (PTM) and the protein structure is considered as labels of these nodes. In this, some of the nodes may be unlabeled. Because of the complex structure of this biological network, it is very difficult to predict the labels of unlabeled nodes using traditional classification methods. The system tries to predict labels such as PTM and the structure of a protein by considering its neighbors which are already labeled. Thus, it becomes a multi-label node classification problem.

Similarly, in the disease-disease interaction network, the nodes are diseases and two diseases are connected by an edge if mutation of the same gene leads to that diseases. The organ which is going to be affected by the disease (disorder class) and a number of each disorder class is taken as the labels of the nodes. As in the PPI network, the system tries to predict the labels of an unlabeled node by considering the dependency with already labeled nodes. Similarly, the proposed system also analyzes another biological network called RNA-RNA interaction network, in which the protein that relates to each RNA is going to predicted.

In order to incorporate these type of network analysis like node classification, the system should be able to learn the features of nodes in the network effectively. So that, before doing multi-label node classification, the proposed system aims to perform node embedding or learn continuous feature representation of each node in the network using deep learning techniques. During embedding of each node, the nodes are mapped into a low dimensional vector space such that should preserve similarities and dependencies between nodes. For that, the proposed system applies different node embedding algorithm for learning a good representation of each node in the network.

1.3 OBJECTIVE

The overall objective of the proposed system is to analyze different networks having the biological significance in such a way that various machine learning approaches are applied to every network and the results are evaluated. There are three different kinds of networks having biological significance and they are given below.

1. Protein-Protein Interaction (PPI) network
2. Disease-Disease Interaction network
3. RNA-RNA Interaction network.

These networks are given as input to various node embedding algorithms to generate their feature vectors and to map it into a low dimensional space. There are various approaches used for generating the embedding for a node. Here, this system uses four different embedding techniques

1. DeepWalk
2. node2vec
3. SDNE (Structural Deep Network Embedding)
4. DNGR (Deep Neural Graph Representation)

Using these node embedding algorithms we obtain the vectors for each node in the network. These representations give the information regarding the location of the nodes in the vector space and dependency with other nodes. The two approaches are in the domain of deep learning. The DeepWalk and node2vec are based on the random walks but the SDNE and DNGR are not based on random walks method. From there the next step is the multi-label classification in which it makes use of four different classifiers namely,

1. Random Forest
2. K- Nearest Neighbors
3. Support Vector Machine
4. Naïve Bayes classifier

Every network learned by these classifiers is having a different model score as well as they are having different prediction accuracy. So that, the system uses various evaluation metrics in order to evaluate the performance of each classifier in a different network. The important evaluation measures that the system makes use of are

1. F1-measure (macro, micro, weighted)
2. Hamming Loss
3. Mean Squared Logarithmic Error

The proposed system tries to make use of all these machine learning approaches for performing node classification in the context of a biological network and results are evaluated.

CHAPTER 2

LITERATURE SURVEY

The survey mainly focuses on studying different kinds of network representation learning methods and how node classification is done in different biological network effectively.

2.1 CLASSIFICATION IN BIOLOGICAL NETWORKS

Proteins are the essential component of every biological process taken place in our body whose function prediction is used in new drug development, synthetic biochemical design etc. The proteins interaction can be modeled as a protein-protein interaction network (PPI Graph) where protein represents nodes in the network, interactions correspond to edges and protein's function represents labels associated with each node. Predicting the function of an annotated protein based on functions of other proteins has strong relevance in the real world scenario [1]. Gene Ontology is a recently proposed functional naming system in order to name function of a protein. GO classifies proteins into three major functional ontologies such as molecular function (MF), biological process (BP), cellular function (CF). There are lots of recent techniques to perform network embedding such as DeepWalk [6], LINE [9], GrarRep, node2Vec [7]. The node2vec algorithm can be used for producing continuous vector representation of each node of PPI graph. Node2vec is a deep model which uses biased random walk procedure and it is able to capture nonlinear network structure. Thus, it also preserves nodes neighborhoods. After producing the vector representation of each node using node2vec, learning algorithms such as SVM or KNN can be used for function prediction.

Prediction of new members of a partially known protein complex/pathway of an organism using random walk has very much importance. The goal is to find a list of candidate

protein. Rank the proteins in the network according to the probability of membership in the partially known complex. Complex or pathway problem is similar to Network reliability problem, which is to find close proximity proteins. The exact solution to reliability problem is NP-hard. It is possible to approximate the reliability between two nodes using Monte Carlo simulation. This technique is not scalable for large protein-protein interaction network random walk technique achieves similar or better accuracy with more than 1,000 times speed-up compared to the best competing technique. The random walk technique a suitable solution for complex or pathway problem [2].

Deep multilayer neural network framework [3] can be used to learn complex features automatically from unlabeled data by using unsupervised learning method. There are two steps in this approach. Firstly, training algorithm of auto-encoders to initialize the parameters of a deep multilayer neural network. Then train the deep multilayer neural network model by applying the gradient descent method using backpropagation. This approach uses greedy layer-wise unsupervised learning to initialize the parameters which not only makes learning deep multilayer neural network possible but also learns much information from large unlabeled data automatically. Four processing step in this method is (1) Named Entity Recognition (NER) to identify the protein names in the biomedical text (2) feature extraction generates raw features using Enju parser and Principal Component Analysis (PCA) (3) auto-encoder learns complex and abstract features by unsupervised greedy layer-wise training on the raw features (4) neural network trains a supervised neural network by the auto-encoder.

The relationship between human genetic disorders and corresponding disease genes can be used for constructing a bipartite graph consisting of two set of nodes based on this. The first set of nodes indicate all the genetic disorders and the second set of nodes correspond to all the disease genes. An edge of disorder and disease gene is formed if the mutation in the gene leads to that disease. From this bipartite graph, two another network can be constructed. These are human disease network (HDN) [4] and disease gene network (GDN). In disease-disease interaction network, disease represents nodes in the network. An edge of diseases is formed if they are associated with a gene in common. Similarly, in the disease gene interaction network, disease genes correspond to nodes and they are connected by an edge if their mutation causes to same genetic disorder.

2.2 NETWORK REPRESENTATION LEARNING METHODS

Node embedding or Network representation learning [5] is the process of mapping nodes in a network to low dimensional vector space in order to extract the features of nodes. The traditional network representation methods face several challenges when dealing with large networks. Most of the traditional methods use iterative or combinatorial approach for learning this representation. But, it is not applicable for a large network, since it results in high computational complexity. Also, these methods map the nodes into vector space based on the assumptions that nodes are independent and identically distributed. So that, the traditional methods cannot preserve the actual network structure. This leads to the development of new network embedding algorithm. There are different recent node embedding methods that preserve network neighborhood, network structure, and community structure has been developed. Some of the methods discussed are node2vec, DeepWalk, LINE, SDNE [8] and so on.

Representation learning [14] deals with how to represent or encode network such that it can be easily exploited by a machine learning algorithm. The traditional methods for feature learning in a graph such as a matrix factorization method, random walk based methods, and graph convolutional networks. It is possible to embed each node as well as the entire network.

DeepWalk[6] is introduced as a tool to analyze graphs to build long-lasting representations which is suitable for statistical modeling. It is a novel approach for learning a latent representation of vertices in a network, DeepWalk learns the social representation of a graph vertices by modeling a stream of deep random walks. The algorithm of DeepWalk mainly consists of random walk generator in which a walk samples uniformly from the neighbors of the last vertex visited until the maximum length (t) has reached and updating procedure which consists of SkipGram and Hierarchical softmax.

DeepWalk can only be applied to a network with an unweighted edge, for embedding both unweighted and weighted edges a method called Large Scale Information Network Embedding (LINE) [9] is proposed. Its objective functions preserve both the first-order and second-order proximities, which are complementary to each other. This method tackles the limitation of stochastic gradient descent and improves the effectiveness and efficiency of the interference. The major difference between DeepWalk and LINE is that DeepWalk makes use

of random walk to expand neighbors which are similar to depth-first search whereas LINE makes use of breadth-first search for the second order proximity.

Deep learning, unsupervised feature learning technique which has proven successful in natural is deployed in network analysis. Deep learning learns social representation with the following characteristics: adaptability, community aware, low dimensional and continuous [10]. The probabilistic model formalism gives two possible paradigms, directed graphical models such as sparse coding and undirected kind models such as Boltzmann machines. This model tends to have many problems when the aim is to extract stable deterministic numerical feature values which efficiently carried out by alternative non-probabilistic feature learning paradigm Auto-Encoders. One of the main criticism addressed to artificial neural networks and deep learning algorithms is that they have many hyperparameters and variants.

Dense matrix multiplications are not involved in the Skip-gram model and hence training is extremely efficient. But the word representations will lack in representing the idiomatic phrases which are not composed of individual words. The whole phrases represented as the vectors make Skip-Gram model expensive and hence other techniques like recursive Autoencoder which aim to represent the meaning of sentences by composing the word vectors. Hierarchical softmax uses a binary tree representation of the output layer with the W words as leaves and for each node represents the relative probabilities of its child nodes. These define a random walk which assigns probabilities of words. A very interesting result of this work which is also called word2vec [13] is that word vectors can be meaningfully combined.

node2vec [7] is an efficient scalable algorithm framework for learning continuous feature representation for nodes in the network that efficiently optimizes a novel network-aware and neighborhood preserving objective using stochastic gradient descent. node2vec employs a flexible random walk procedure that allows exploring the neighborhood in a breadth-first and as well as depth-first fashion. The random walks are based on two parameters, one is the return parameter which controls the likelihood of immediately revisiting a node and the in-out parameters which differentiate between the inward and outward nodes. The random walk strategy is more beneficial than pure breadth-first and depth-first search. The algorithm makes use of the word2vec strategy to convert the list of nodes generated by random walks. Which learns the feature representation of the input graph which can be used for applications like link prediction, multi-label node classification.

Network embedding [5] is an important method to learn low-dimensional representations of vertexes in the network and aiming to capture and preserve the network structure. If network structure is complex, shallow models cannot capture the highly non-linear network structure, resulting in sub-optimal network representations.SDNE [8] method can solve this problem. It is a deep model with a semi-supervised architecture, which simultaneously optimizes the first-order and second-order proximity. The semi-supervised deep model has multiple layers of non-linear functions, thereby being able to capture the highly non-linear network structure. The first-order proximity is used to preserve the local network structure and second-order proximity is used to capture the global network structure. By jointly optimizing them in the proposed semi-supervised deep model, SDNE can preserve the highly-nonlinear local-global network structure. Then evaluate the generated network representations in a variety of network datasets and applications. This method outperforms the state-of-art method in terms of quality and speed.

DNGR (Deep Neural Graph Representation) [16] is a node embedding strategy which can learn graph structural information without using any sampling method like a random walk. It makes use of a random surfing model that is inspired by PageRank model for ranking pages for generating a probabilistic co-occurrence matrix from a given weighted input graph. From this, PCO matrix it calculates PPMI matrix. In order to learn complex, dense structural information from the graph a Stacked Denoising Autoencoder is used. That generates low dimensional vector representation from PPMI matrix.

Edge classification [15] is the challenging area in network analysis which is used to determine labels of unlabeled edges in a content-based network. As nodes are having labels or features, each edge in the network is associated with some text content. In the context of the online social network, it can be content of the communication. Since most of the methods are developed for node classification rather than edge classification, this paper describes a scalable matrix factorization method for edge classification which is called *ECONOMICS.*

CHAPTER 3

SYSTEM ANALYSIS

3.1 NEED OF PROPOSED METHOD

The biological interaction network like PPI network, disease-disease network and RNA-RNA interaction network have strong relevance in various applications Most of the biological activities is taken place by the interaction between biological units such as protein, DNA, etc. Due to the numerous biochemical interaction between these units and signaling interaction between cellular components, the human biological system becomes more complex to analyze, visualize and for modeling. The traditional paradigm for node classification in biological network mainly involves hand-engineering, iterative graph-based approach. But, the complexity and noise in the biological interaction network make it unanalyzable by traditional classification methods. Since, the data volume of all the networks grows exponentially and data is linked in real-world cases, the traditional methods are not scalable for learning representation from a large network. In addition to that, traditional classification methods are based on i.i.d assumptions. That is, these methods map the nodes into vector space without considering the dependencies and similarities between nodes. But, when a network is concerned the similarity and the network structure should be preserved. Also, experimental approaches gives only a low throughput. Thus, the accuracy of the classification become very less.

This lead to the development of computational techniques which can give a high throughput result by analyzing the biological interaction network effectively. So that, the proposed system performs multi-label node classification in a biological network using deep learning techniques. The system uses approaches based on both random walks and without a random walk for node embedding. It analyses three different interaction network using various approaches and the result are evaluated.

Since all these biological networks have strong relevance in the real world scenario. For example, predicting the protein structure helps to identify the function of the protein which in turn help to the development of new drugs. In addition to that, identifying the organ which is going to be affected by a disease from the disease-disease interaction network is also useful for early disease diagnosis. Due to this relevance of this kind of biological network gives importance for developing an effective computational technique for analyzing the network. In this scenario, the proposed system gains more importance.

3.2 FEASIBILITY ANALYSIS

3.2.1 Operational Feasibility

The proposed system is operationally feasible. It is efficient in solving the problem identified during the requirement analysis of the project. The input is processed and the generated output meets the requirement of the defined problem. Thus, the system is operationally feasible. The system is able to work independently, so it can be used by anyone and can be easily understood by everyone.

3.2.2 Technical Feasibility

The proposed system makes use of the hardware and software which are efficient to satisfy the requirements. All the technologies needed for implementing the system are known to work. Make use of the python 2.7.13 version and the libraries needed. The classifier needed for the training purpose is easily understandable. So anyone can easily handle. The system can be made to work in any of the environment which satisfies the above specification.

3.2.3 Economic Feasibility

The system is economically feasible. Free software is used to develop the system and also the libraries which are available freely are only used by the system. The dataset used for training the classifier is available to the public and hence it is downloaded for free.

CHAPTER 4

SYSTEM DESIGN

The proposed system aims the analysis of three different networks having the biological significance. These networks are given as input to various machine learning algorithms and their performance is measured. Fig 4.1 shows the complete process of the proposed system.

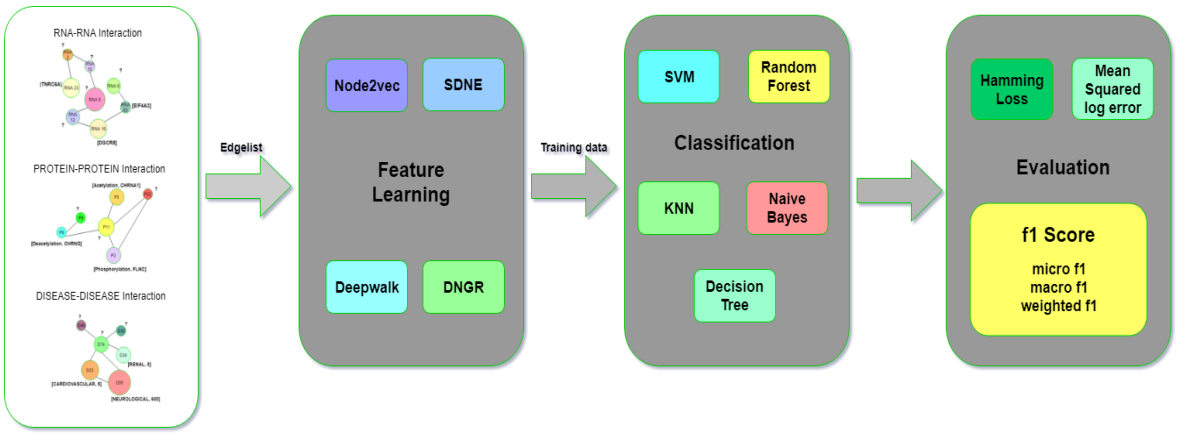


Fig 4.1: System Design

4.1 BIOLOGICAL NETWORKS

The proposed system analyzes three different biological networks which are having strong relevance in the real world scenario and performs node classification. The networks used are listed below.

1. Protein-Protein Interaction (PPI) network
2. Disease- Disease Interaction network
3. RNA-RNA Interaction network

4.1.1 Protein-Protein Interaction (PPI) Network

The first network chosen is the protein-protein interaction network in which the nodes represent the proteins and the edges represent the interaction between them. Human protein network is chosen for this. Fig 4.2 shows the PPI network. The protein is the component that is inside the cell of every living organism which is needed in formulating the behavior and coordinating the body functions of an organism. Every protein in the human cell have some features such as its functions, post-translational modification (PTM), structure, sequences and so on which can be taken as the node labels in the PPI network. Since we perform multi-labels classification, we use PTM and protein structure as the labels and trying to predict labels of unlabeled nodes by considering the dependency or interaction with other nodes. Predicting these features some relevance in the field of medicine and biology like it the development of new drugs, functionality identification by knowing the structure and so on.

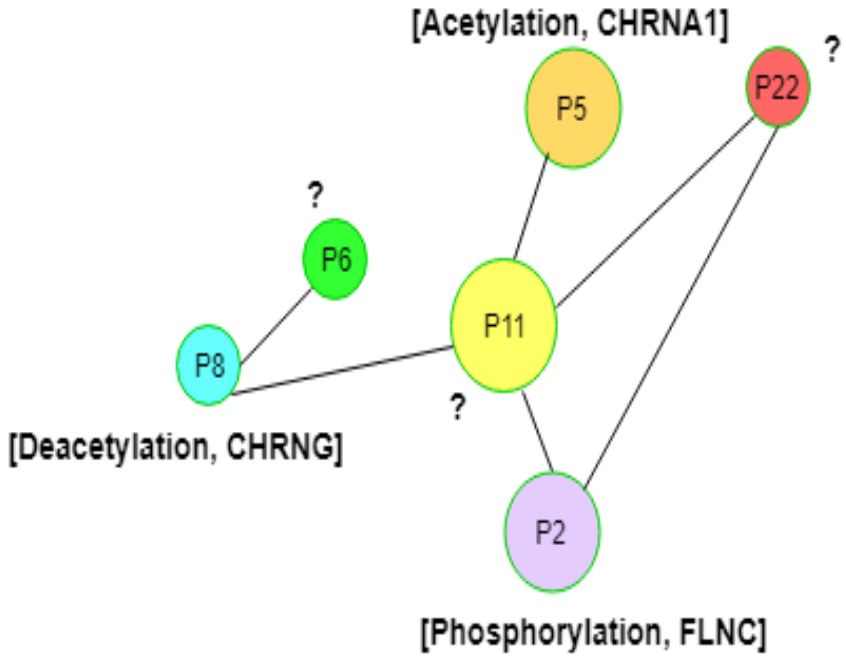


Fig 4.2: Protein-Protein Interaction (PPI) Network

Here, the diagram is plotted by taking the nodes as the names of proteins and the post-translational modification and the protein structure as the label. In this above network, the Acetylation and CHRNA 1 are the labels of node P5 which indicates the post-translational

modification and the protein structure of that protein respectively. Similarly, node p8 have also two labels Deacetylation and CHRNG. Since, node P11, P6, P2, P22 is not assigned with any label, our task is to find out the labels of these unlabeled nodes based on the labels of neighbor nodes which is already labeled.

4.1.2 Disease-Disease Interaction Network

In the disease-disease network, the nodes are the diseases and the edges are their interactions. The two diseases are connected to each other when they have associated with a common gene. This implies the probability of occurrence of one disease when the other is present. The labels are the disorder classes and the types of disorder classes. The disorder class depicts to which organ the disease is going to affect. From the gene-disease database the relationship between the diseases are found out and hence the edge list is obtained. The string labels are twenty-two in numbers and for processing, they are given numbers. They have a larger significance in the real world. According to a particular disease, the drug can be prescribed so the disease interactions are necessary for this. In addition to that, identifying an organ affected by a disease helps for early disease diagnosis. So this is very relevant in a real-world scenario.

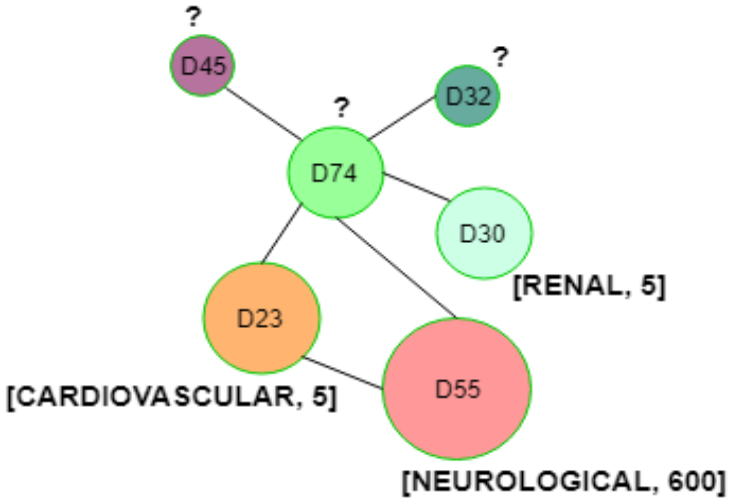


Fig 4.3: Human Disease Network

The Fig 4.3 above shows the disease network. In this, nodes D32, D23, D30, etc. indicate diseases and the edge is formed if a mutation in a common gene leads to that diseases.

In the given network, the node D23 is having two labels CARDIOVASCULAR and 5 that represents the disorder class and the number of type of that disorder class. The proposed system needs to predict the labels of unlabeled nodes like D74, D32, D45, etc.

4.1.3 RNA-RNA Interaction Network

The Human RNA - RNA interaction is taken as another network. Ribonucleic acid (RNA) is a polymeric molecule essential in various biological roles in coding, decoding, regulation, and expression of genes. RNA and DNA are nucleic acids and along with lipids, proteins and carbohydrates, constitute the four major macromolecules essential for all known forms of life. Like DNA, RNA is assembled as a chain of nucleotides, but unlike DNA it is more often found in nature as a single-strand folded onto itself, rather than a paired double-strand. Cellular organisms use messenger RNA (mRNA) to convey genetic information (using the nitrogenous bases guanine, uracil, adenine, and cytosine, denoted by the letters G, U, A, and C) that directs the synthesis of specific proteins. Many viruses encode their genetic information using an RNA genome.

The data is provided as an interaction between miRNA which is microRNA. A microRNA (abbreviated miRNA) is a small non-coding RNA molecule (containing about 22 nucleotides) found in plants, animals and some viruses that functions in RNA silencing and post-transcriptional regulation of gene expression. While the majority of miRNAs are located within the cell, some miRNAs, commonly known as circulating miRNAs or extracellular miRNAs, have also been found in the extracellular environment, including various biological fluids and cell culture media.

In this RNA-RNA interaction network, RNA is considered as nodes and their interaction as edges. From the RNA-protein network, we are able to find protein relates to each RNA. Thus, that protein can be assigned a label of each node in RNS-RNA interaction network. Identification of such proteins corresponds to each node has also some significance. Because the proteins and the corresponding RNA are responsible for most of the functions taken place in a cell. The below Fig 4.4 shows RNA-RNA interaction network. The nodes RNA 23, RNA 16 and so on indicate the interacting RNAs. The TNRC6A, DGCR8 denotes the proteins that relate to the RNA 23, RNA 16 respectively which is taken as the labels of that nodes.

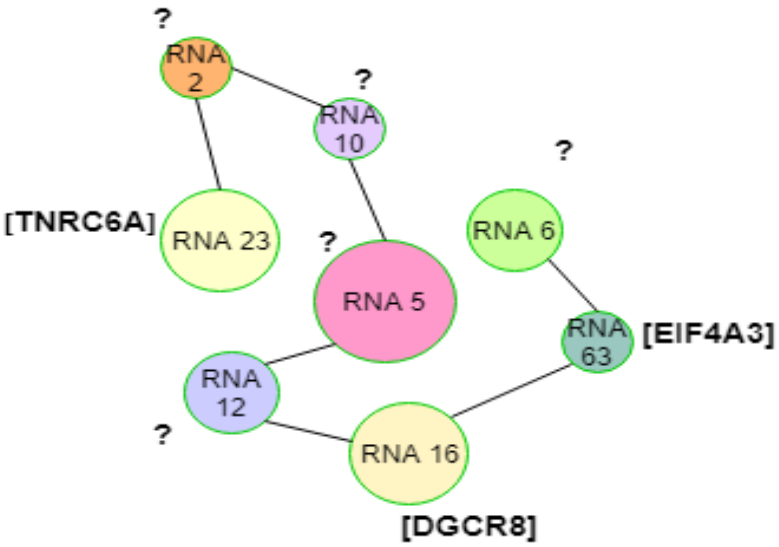


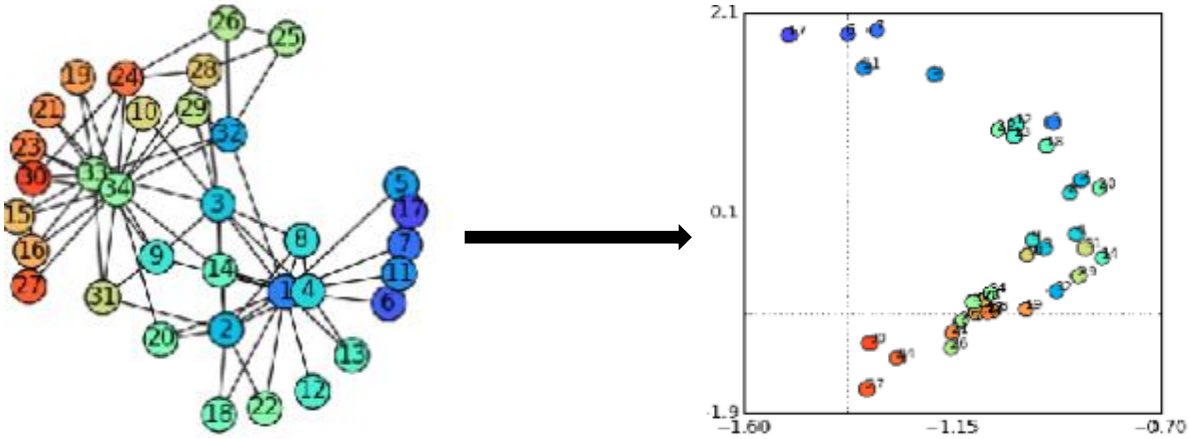
Fig 4.4: RNA-RNA Interaction Network

4.2 NODE EMBEDDING

The three networks are chosen accordingly and they are preprocessed for obtaining a graph. Later, these graphs have to be embedded into a vector space using the node embedding algorithms. Node embedding or Graph representation learning means that extracting features of nodes in a network. Nodes are embedded into low dimensional space with dimension d in such a way that similarity and dependency between nodes should be preserved.

It can be formally defined as Let $G = (V, E, X, Y)$ be the network where V represents a set of nodes, E is the set of edges, X and Y denotes the node attributes and node labels respectively. Then, for each node u $\in V$, we define a mapping function $f: V \longrightarrow R^d$. This indicates the mapping of each node in the network into a low dimensional vector space having dimension d. Here, R^d is the learned feature vector representation of each node u $\in V$ and f is the mapping function and also represents the matrix of size $|V| \times$ d. The transformation f should preserve the actual network information. This means that two nodes which are similar in the network should also be represented similarly in the vector space also. The learned feature representation R^d should also satisfy some conditions such as 1) it should be low dimensional, d << |V | 2) informative, i.e., preserve actual network structure 3) continuous, learned representation must be continuous.

The Fig 4.5 (a) and Fig 4.5 (b) shows input graph and the corresponding vector representation of each node respectively.



(a) Input Graph $G = (V, E, X, Y)$ (b) Vector representations

Fig 4.5: Node Embedding

When we map a node into a low dimensional vector space, it should preserve some network proximities such as first-order proximity, second order proximity (higher proximity), intra-community proximity (homophily) and structural equivalence. The first order proximity can be defined as, if $(u_i, v_j) \in E$, the first order proximity between u_i and v_j is w_{ij} otherwise 0. That is, if there is an edge between two nodes, then they should be closer to each other in the vector space also. When two nodes u_i, v_j have same neighborhood nodes, then they are mapped closely together in vector space, it is called second order proximity. Intra-community proximity is defined as If two nodes u_i and v_j belongs to the same community of nodes, then when we map these nodes into vector space it should be close together in the vector space. In addition to that, if two nodes in different communities possess same structural role then after node embedding, both nodes should have similar embedding. This similarity is referred as structural equivalence. During node embedding, it should be able to learn feature representation of any node by preserving these proximities. In fact, the main objective any Feature learning algorithm is to learn features of nodes in the network in such a way that local neighborhood of node is preserved. The three important different network representation learning algorithms are given below.

1. DeepWalk
2. node2vec
3. SDNE (Structural Deep Network Embedding)
4. DNGR (Deep Neural Graph Representation)

In this, DeepWalk and node2vec are based on random walk method and these two methods focus on preserving network neighborhood. But, SDNE and DNGR is not based on a random walk and try to preserve network structure.

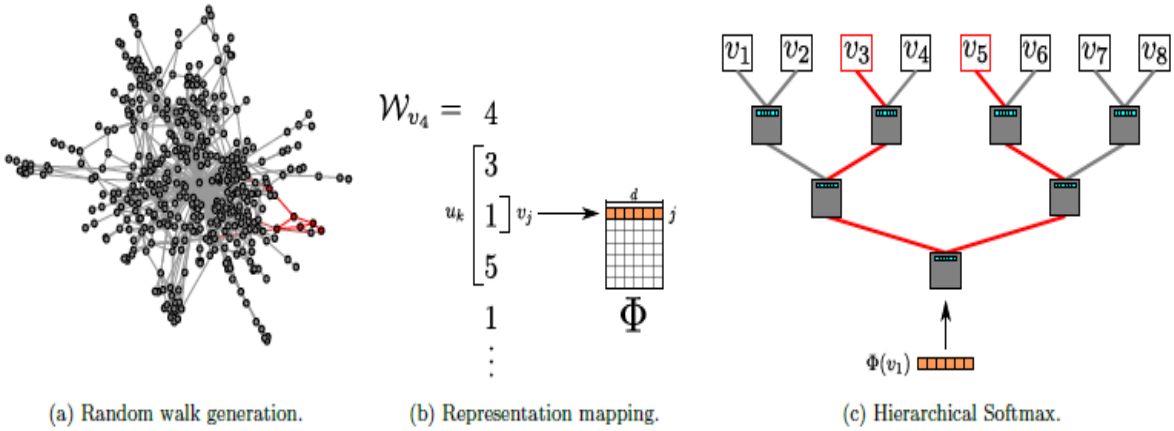
4.2.1 DeepWalk

The DeepWalk is a method for learning a latent representation of each node in the network using a truncated random walk so that it can preserve the node inference such as the neighborhood of nodes, community structure and so on. This approach generalizes the recent techniques like language modeling and deep learning techniques by treating words to graphs. This makes use of the truncated random walks to learn representation in such a way by treating the words as sentences. The algorithm comprises two important components

1. Random walk generator
2. Update procedure

The random walk generator is used to generate a truncated random walk for each vertex thereby it can preserve local neighborhood of each node. This random walk generator takes a graph *G* as input and also define the term embedding size, walk length, walk per node etc. Then, it chose a vertex randomly and performs a random walk of length *t*. The length of the walk is not fixed. Also, it is able to perform a random walk on the same vertex multiple times. After getting random walk corresponds to each vertex, then use it to update the representations. DeepWalk uses SkipGram model and hierarchical softmax to update the representation such that nodes neighborhood property can be preserved. For each vertex representation, the SkipGram model maximizes the probability of occurrence of neighbors in the random walk using conditional probability. The hierarchical softmax gives an improvement over the SkipGram model in terms of training time by assigning the nodes as leaves of a binary tree and

trying to maximize the probability of a specific path. The following Fig 4.6 shows an overview of DeepWalk.



(a) Random walk generation. (b) Representation mapping. (c) Hierarchical Softmax.

Fig 4.6: Overview of DeepWalk

4.2.2 node2vec

The node2vec algorithm is the most recent approach for node embedding which automates this process by casting the feature extraction process as a representation learning problem. The node2vec framework is made inspired by the Skip-gram model, and consider the network as a document. In a typical node classification problem, we are interested in predicting the most probable labels for the nodes. The node2vec algorithm has the compositionality feature to extend the learned labels for the nodes to the pair of nodes by composing the learned features of the individual nodes by using simple binary operators. The node2vec algorithm is able to explore the neighborhood of nodes by using a biased random walk method. That is, it can preserve network neighborhood when a node is mapped to low dimensional vector space.

The node2vec algorithm also preserves two kinds of network similarity during embedding homophily and structural equivalence. If two nodes u and v belongs to the same community of nodes, then when we map these nodes into vector space it should be close together in the vector space. In addition to that, if two nodes in different communities possess same structural role then after node embedding, both nodes should have similar embedding.

This similarity is referred as structural equivalence. For preserving this similarity and network neighborhood of nodes, it uses two sampling strategies.

1. Breadth-first sampling (structural equivalence)
2. Depth-first sampling (homophily)

Apart from DeepWalk, it performs a biased random walk procedure in such a way that neighbors can be explored in BFS as well as in DFS fashion. The random walks are taken care by the two parameters the Return parameter (p) and the In-out parameter (q). The return parameter takes care of immediately revisiting a particular node. And q take care of the search either in BFS and DFS fashion. Thus a random walk from an unlabeled node to a labeled node is carried out and then vectorize the sequence of nodes which map the nodes to a low-dimensional vector space. The random walks are more efficient since they are efficient in both reduced time and space complexity.

4.2.3 SDNE (Structural Deep Network Embedding)

It is a semi-supervised deep model for mapping vertices of a network into low dimensional vector space and to get the feature vectors of each node. The traditional methods for network embedding face some problem like learning highly complex non-linear network structure is difficult. Also, most of the networks are sparse in nature and preserving both local and global structure simultaneously is difficult for the complex network. In order to address all these problems, a new approach called SDNE is developed. It uses multiple layers of non-linear functions to capture highly complex non-linear network structure. Similarly, for preserving local and global network structure, it defines first order and second order proximities.

First order proximity means if two vertices are linked by an edge in a network, then there is a positive first order proximity between that edges. That is, the two vertices are similar or mapped closely together in vector space if they are connected by an edge. Based on that, it is possible to define second order proximity. It indicates that if two edges have same neighborhood nodes, then they are mapped closely together in vector space. In fact, the first order proximity preserves the local structure and second order proximity is used to capture network global structure. During network embedding, both proximities should be preserved.

The following Fig 4.7 illustrates a semi-supervised deep model of SDNE which tries to preserve first order and second order proximity using a deep autoencoder.

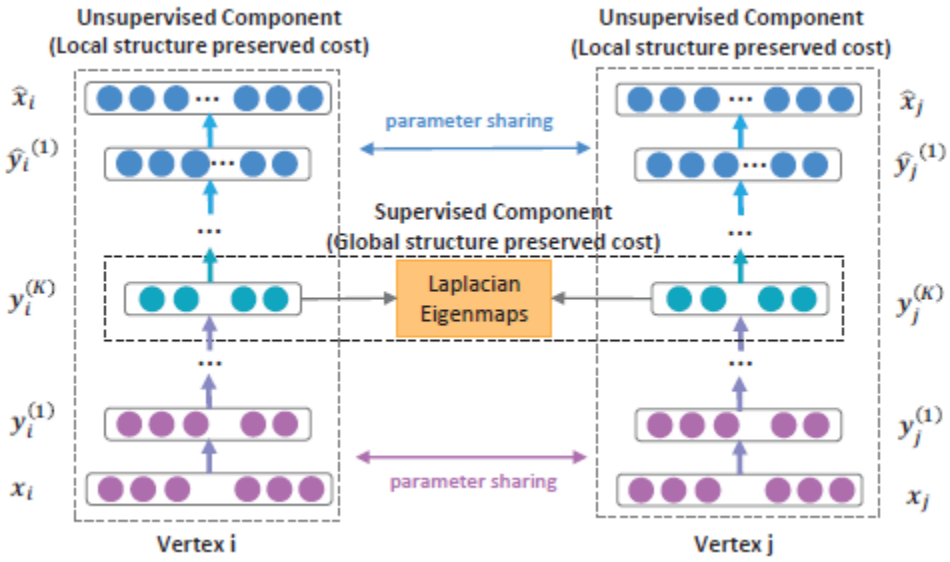


Fig 4.7: Framework of SDNE

4.2.4 DNGR (Deep Neural Graph Representation)

Apart from other node embedding method, DNGR uses a different strategy for learning vector representation from a graph. Other NRL methods like DeepWalk performs sampling strategies called a truncated random walk in order to transform an unweighted graph to a linear structure. While DNGR captures graph structural information directly from a weighted graph effectively without using any sampling strategies. The main components in DNGR are

1. Random Surfing (Input – graph, Output – PCO matrix)
2. Calculation of PPMI matrix (Input – PCO matrix, Output – PPMI matrix)
3. Stacked Denoising Autoencoder

Random surfing model is inspired from a PageRank model used for ranking pages. At first, DNGR performs random surfing which takes a graph as input and produces probabilistic co-occurrence matrix without using any sampling method like a random walk. Then, it calculates PPMI (pointwise mutual information) matrix. Finally, it makes use of SVD (singular value decomposition) procedure in order to learn low dimensional vector representation from

the PPMI matrix. But, the main objective of DNGR is to learn high quality, dense low dimensional vector representation from the PPMI matrix. For that, it uses stacked denoising autoencoder to learn low dimensional vector representation such that most of the graph structure information can be captured. The following Fig 4.8 shows main components of DNGR.

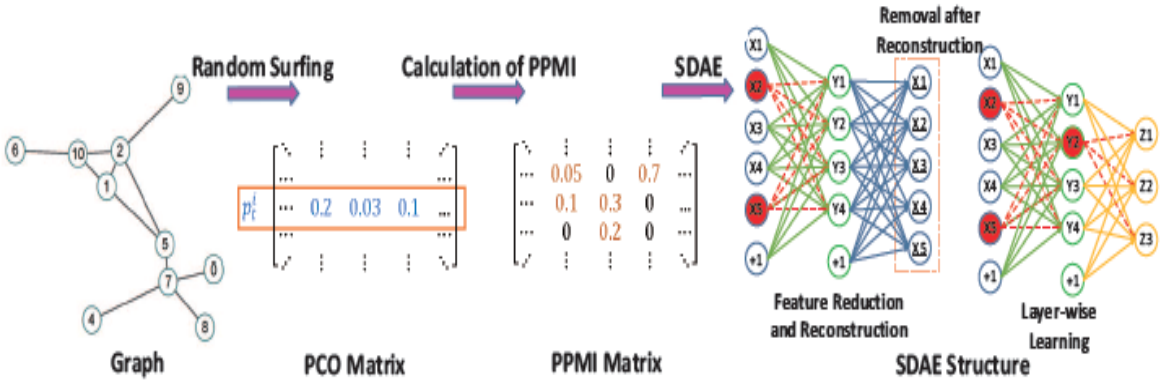


Fig 4.8: DNGR Components

4.3 MULTILABEL CLASSIFICATION

The feature vectors learned using node embedding algorithm are then used for the classification task. Here, the proposed system performs multi-label node classification. The process of node classification is used to assign the labels for the unlabeled nodes based on known labels of neighbor nodes. Since the system aims to do multi-label classification, the classifier needs to predict two or more labels or features of each node.

Multi-label classification can be formally defined as Let X be the input space which contains input samples $x \in A_1 \times A_2 \times ... \times A_f$ such that A_1, A_2,…., A_f denotes the features of the node. Let L be the set of all labels and $P(L)$ indicates the power set of labels. *Let* Y be the output space, where $Y \in P(L)$. Let F represents a multi-label classifier, defined as $F: X \longrightarrow Y$ such that input to X will be x and output will be $y \in Y$.

The classifier follows through two processes the testing and the training. The training process is used to train our classifier to predict a label for the new data entering without a label and the testing process is used to test the classifier for its accuracy. The classifiers used are given below

1. Random Forest
2. K- Nearest Neighbors
3. Naïve Bayes
4. Support Vector Machine
5. Decision Tree

4.3.1 Random Forest

Random forests are ensemble learning method for classification, which is operated by constructing a multitude of decision trees at training time. Then the output will be the class that is the mode of the class. It uses a bagging algorithm to create a random sample. From the given dataset it creates a new dataset. Then the model is trained on the new dataset. Unlike a tree, there is no pruning in a random forest. We obtain several trees and the final prediction is obtained. The advantages of random forest are that they can be used for the unsupervised learning task, can be used as a feature selection tool.

4.3.2 K- Nearest Neighbors

It is a non-parametric method used for classification. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification. When the KNN is used for classification then the class membership will be the output. It is a lazy learner.

4.3.3 Naive Bayes

This is a probabilistic classifier, which takes the independent assumptions between the features. Naive Bayes classifier uses the concept of Bayes theorem. Let y be a class variable and x1, x2…..xn be the independent features. Then, using the Bayes theorem, the probability that the features belong to class y is given below in the Eq. 4.1.

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)} \quad (4.1)$$

For all i, it can be simplified using Eq. 4.2.

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)} \quad (4.2)$$

Since, P(x1, x2…..xn) is a constant, it can be eliminated.ie, for a class y, we are trying to maximize

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y), \quad (4.3)$$

ie, the features x1,x2…..xn ∈ y iff P(y|x1,x2….xn) is maximum.

4.3.4 Support Vector Machine

Support Vector Machines are the discriminative classifier which is used to classify the given data points. It is the supervised approach in which given training set with the labels and it will predict the labels for the incoming new data. By using the kernel trick the SVM can perform the non-linear classification efficiently, which is mapping the inputs implicitly to a higher dimensional space. For the data which are not labeled, an unsupervised learning approach called as the support vector clustering is used. Thus this classifier can perform the task of classification over a large dataset with very high accuracy.

CHAPTER 5

IMPLEMENTATION

5.1 DATASETS

The three networks chosen which are having the biological significance are obtained from different databases. The first network is the PPI network is obtained from the HPRD database and the dataset have the nodes and their various features. The Post Transcriptional Modification and protein structure are chosen as the labels of each node and using node classification task the classifier assigns labels for unlabeled nodes by considering the labels of its neighborhood.

The second network is the Disease-Disease interaction network. This dataset is obtained from the Github-Gephi repository. Here the dataset needs to be preprocessed. The Disease-Gene interactions are processed to make the Disease-Disease interaction. That is, this network has a larger significance in the real world scenario because the Disease-Disease interaction is formed when two disease has the same genetic disorder. The organ or disorder which is going to be affected by the disease and the number or type of that disorder is taken as the label of each node.

The third network is the RNA-RNA interaction network and is obtained from the RAID database. From the RNA-protein interaction which is also obtained from the RAID database used for getting the labels. i.e., the protein that relates to that RNA can be identified by the RNA-protein interaction which in turn helps for assigning the labels of nodes.

5.1.1 Protein-Protein Interaction (PPI) Network

Protein-Protein interaction network is available on the Human Protein Reference Database (HPRD). Human Protein Reference Database (HPRD) is a rich resource of

experimentally proven features of human proteins. Protein information in HPRD includes protein-protein interactions, post-translational modifications, Protein structure, enzyme/substrate relationships, disease associations, tissue expression.

The main files used for our experiment is the human protein-protein interaction and file containing entries of Post Translational Modification (PTM) and protein architecture of all proteins. It is taken as the labels of each node in the PPI network. The network contains 7970 nodes and 39420 interactions between proteins. PTM label of each node indicates the enzymatic modification of each protein taken place after the protein synthesis. This modification can be done in various ways. For example, covalent addition of functional groups, degradation of entire proteins, post-translational cleavage etc. are some the modification taken place. So that, identifying and predicting these PTM of a protein is crucial for research in cell biology, controlling biological process, disease treatment and prevention etc. There are 14352 PTM labels. In this, some of the PTM labels are Acetylation, Phosphorylation, Glycosylation, Sumoylation and so on.

Since it is a multi-label classification, we use protein structure as another label of the nodes. A total of 209 protein structure is used for this purpose. The protein structure includes FAU, CHRNA1, ASCL1 and so on. Predicting the structure of the protein has strong relevance in the real world scenario. That is, if it is possible to predict the structure of a protein based on the interaction of proteins in the PPI network, then its function can easily find out. Apart from this, one can easily predict the drug or molecule that bind to it. The main objective is to assign these two labels for unlabeled nodes based on labels of neighbor nodes.

5.1.2 Disease -Disease Interaction Network

A network of disorders linked by known disorder-gene associations of humans. Two diseases have a connection in common if they share a gene between them. The dataset is available on Human Disease Network, Goh K-I, Cusick ME, Valle D, Childs B, Vidal M, Barabási A-L (2007), Proc Natl Acad Sci USA 104:8685-8690. The disease-disease interaction network is constructed from a disease- gene bipartite graph. The network has 1308 known diseases connected together which accounts for 2376 interactions. Each disease is associated with a particular part of the human body, which together called as a disorder class. There are

22 disorder classes. The disorder class and the type of disorder class is chosen as the two labels of each node in the disease-disease interaction network. Some of the disorder class labels are neurological, renal, psychiatric etc. There are more than 600 neurological disorders. So that, the type of each disorder class is taken as another label of each node. For This input, dataset consists of 1308 nodes and 2376 edges or interactions. And they are given to the representation learning algorithm. The vectors representation for each of the nodes is obtained. Then these nodes need to be labeled.

5.1.3 RNA-RNA Interaction Network

The RAID v 2.0 database is used for obtaining the RNA- RNA interaction information. The RAID is an RNA- associated interaction database which contains both RNA-RNA interaction and RNA-Protein interaction of more than 60 species. From this, only human RNA-associated information is taken for our work. The network consists of 1384 nodes and 871569 interaction between RNA. Since each RNA is associated with a protein so that it is taken as the label of each node.

5.2 TECHNOLOGY/TOOLS

- Python 2.7 and 3.6

  Python is a high-level object-oriented programming language.

- NetworkX Package

  NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

- Sklearn

  Sklearn is a free software machine learning library used for Python programming language. It is designed to interoperate with the Python numerical and scientific libraries Numpy and SciPy

- Gensim

  Gensim is a robust open-source vector space modeling and topic modeling toolkit implemented in Python.

● NumPy Library

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

● Matplotlib Library

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

5.3 SYSTEM PROCESS

The table 5.1 below shows overall process carried out

Sl. No	Network Selected	Node Embedding Algorithm	Classifier	Evaluation Metrics
1	Protein-Protein Interaction(PPI) Network	1) DeepWalk 2) Node2vec 3) SDNE 4) DNGR	1) Random forest 2) K-Nearest Neighbors 3) Support Vector Machine 4) Naive Bayes	1) F1-measure (weighted, macro, micro) 2) Hamming Loss 3) Mean Squared log Error
2	Disease-Disease Interaction Network			
3	RNA-RNA Interaction Network			

Table 5.1: Overall System Process

The node classification in biological networks is carried out to get the labels for each of the unlabeled nodes. The three networks chosen are all initially preprocessed to get the edge

list in such a way that they can be given as input to the node embedding algorithms. The edge list constitutes the interaction between the two nodes. In protein-protein interaction networks, the interacting nodes are the proteins. In case of RNA-RNA interaction network the two interacting RNA nodes will constitute an edge and in Disease-Disease interaction networks, the edge list is formed based on the common gene which is the cause of the disease.

The input edge list gave the embedding algorithms. The algorithms used are the node2vec, DeepWalk, SDNE. This node embedding is done in order to get the feature representation of each node in the network. In case of the DeepWalk algorithm, it finds out the embedding of the nodes by carrying out a random walk from a node chosen. In node2vec the random walk is a biased random walk and the corresponding representation is found out. A different approach for node embedding is done by SDNE, in which auto encoders are used and representation is found out. This method is not following any walk based approach to learn the representations.

After the representations are obtained the next step is to give this as input to the classifiers for the node classification. The basic classifiers used for this purpose are Support Vector Machines, Random Forest, K-Nearest Neighbors, and Naïve Bayes. The classifiers will classify the nodes accordingly and they predict the labels for the unlabeled nodes. The dataset is split into two for the testing and the training purpose and the interest is on multi-label classification. The multi-label represents that a single node has more than one label. Thus the classifier has to predict those labels accurately. As the nodes in the networks are multi-labeled then the evaluation measures used for them are a little different. After this classification process, these are subjected to the evaluation measures for evaluating the performance of each classifier.

5.4 EVALUATION MEASURES

The evaluation measures used for evaluating the performance of classifiers are given below

1. F1- measure(macro, micro, weighted)
2. Hamming Loss
3. Mean Squared Logarithmic Error

5.4.1 F1- measure

The F1-score can be defined as a measure of test's accuracy. It is calculated in Eq. 5.1 by taking the harmonic mean of two basic evaluation measures-precision and recall. Generally,

$$F_\beta = (1+\beta^2)\cdot\frac{\text{precision}\cdot\text{recall}}{(\beta^2\cdot\text{precision})+\text{recall}} \tag{5.1}$$

Where β = 0.5,1 or 2.

When β=1,

$$\mathrm{F}_1 = \frac{2\cdot\mathrm{PREC}\cdot\mathrm{REC}}{\mathrm{PREC}+\mathrm{REC}} \tag{5.2}$$

Precision (PREC) is the number of correct positive predictions divided by a number of all positive predictions returned by the classifier and Recall (REC) is the number of correct positive predictions divided by a total number of positives. It can be calculated as follows using Eq. 5.3.

$$Recall = \frac{TP}{TP+FN} \qquad PREC = \frac{TP}{TP+FP} \tag{5.3}$$

Macro-f1 and micro-f1 are used for the multi-label classification task.Macro-f1 evaluates the classifier's label set prediction performance which takes an average of f1-measure of different labels.For example, for a label denoted by A, we first calculate the FP(A), TP(A) and FN (A).This indicates a number of false positives, true positives, false negatives for all the instances that are predicted as A. Based on this, Macro-f1 can be defined using the following Eq. 5.4.

$$Macro-F1 = \frac{\sum_{A\in C} F1(A)}{|\,C\,|} \tag{5.4}$$

Where C is the over label set and F1(A) is the f1-measure of label A. If the Macro-f1 measure is larger, then better the performance.

Similarly, Micro-f1 can also be calculated. This also evaluates the classifier's label set prediction performance, only the difference is that it gives equal weight to each instance. The Eq. 5.5 given below is used to calculate Micro-f1 score.

$$Pr = \frac{\sum_{A \in C} TP(A)}{\sum_{A \in C} (TP(A)+FP(A))}, R = \frac{\sum_{A \in C} TP(A)}{\sum_{A \in C} (TP(A)+FN(A))}$$
$$Micro - F1 = \frac{2*Pr*R}{Pr+R} \tag{5.5}$$

Where *Pr* and *R* denote the precision and recall respectively. Similar to Macro-f1, if larger the value, better will be the performance.

5.4.2 Hamming Loss

Hamming loss is one of the evaluation measures which is useful in the multi-label classification task. It measures the fraction of wrong labels to a total number of labels during classification. Hamming loss for two set of samples is calculated based on the following Eq. 5.6.

$$L_{Hamming}(y, \hat{y}) = \frac{1}{n_{\text{labels}}} \sum_{j=0}^{n_{\text{labels}}-1} 1(\hat{y}_j \neq y_j) \tag{5.6}$$

Where y_j is an actual label and $\hat{y}_j$ is the predicted label. And n_{labels} denotes a total number of labels.

5.4.3 Mean Squared Logarithmic Error

Let consider $\hat{y}_i$ as the predicted value of i-th input sample and y_i represents truth value of i-th sample. It is calculated using the following Eq. 5.7.

$$\text{MSLE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (\log_e(1+y_i) - \log_e(1+\hat{y}_i))^2. \tag{5.7}$$

CHAPTER 6

EXPERIMENTAL RESULTS

6.1 EFFECT OF EMBEDDING

The multi-label node classification is performed in three biological networks are yielded with the following results. Each of the three networks is embedded using four different node embedding methods and their performance is evaluated using different evaluation measures. The below Fig 6.1 shows the performance of each node embedding algorithms in the disease-disease interaction network.

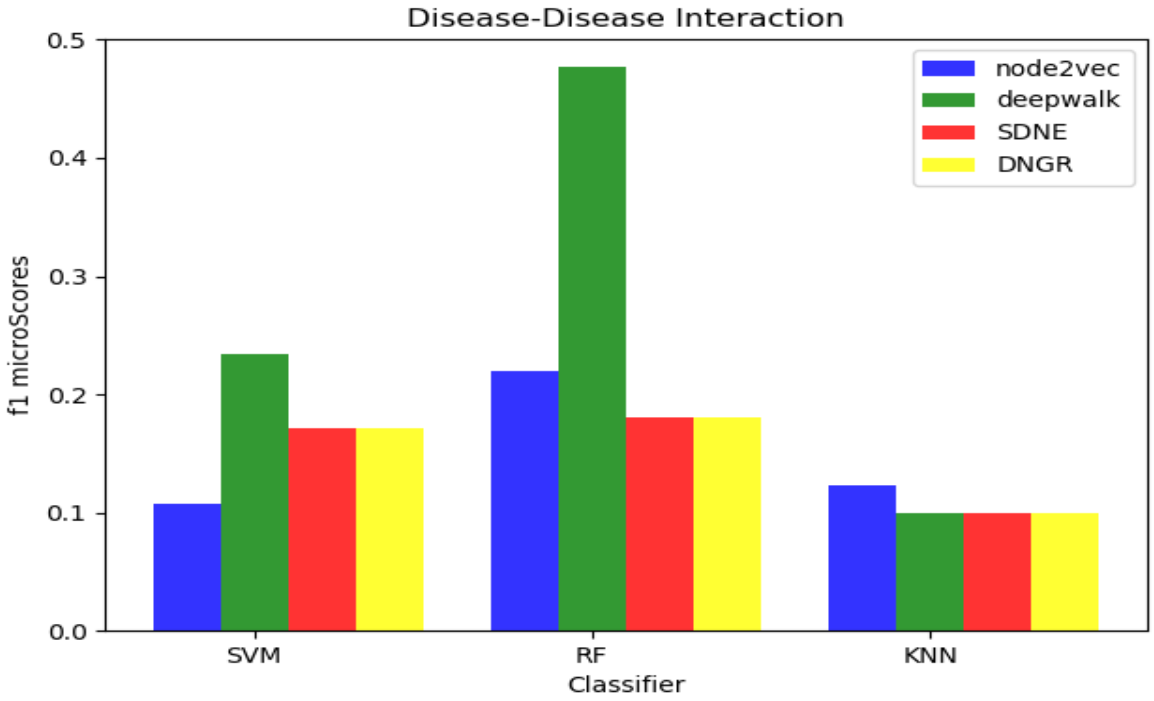


Fig 6.1: Disease-Disease Interaction -Embedding vs. f1-micro score

The above graph shows the performance of each of the node embedding methods. The f1-micro score is chosen as the measure of evaluation and the bar graph is obtained. The

f1-micro score for the four different classifiers for a fixed dimension is shown in the diagram. The performance of DeepWalk is more compared with the others and when the embedding without a random walk is considered then both SDNE and DNGR has almost the same performance in the Disease-Disease network. This is because DNGR and SDNE both of the embedding methods generate vector representation without considering the random walk strategy. They both make use of the auto-encoders for the purpose of dimensionality reduction. And this is the reason why they show the same performance.

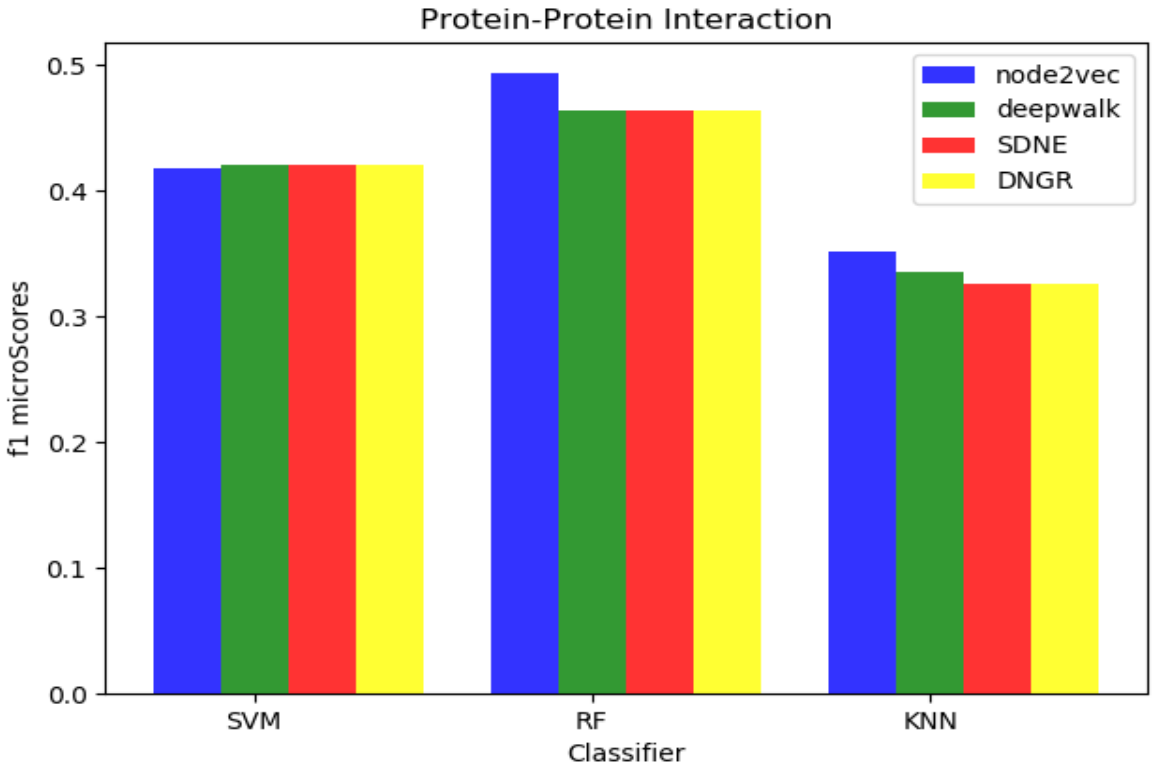


Fig 6.2: PPI network - Embedding vs. f1-micro score

The protein-protein interaction network has the performance as shown in the above Fig 6.2. The number of nodes in this network is more and also the interacting edges are also more. So as the evaluation proceeds it is found that as the model changes the embedding methods shows a variable performance. Each embedding is having a different score for the different models in the protein-protein interaction network. As analyzed then the two models the Random Forest and the K-nearest neighbors have a greater score in the node2vec embedding. All these models are non-parametric in nature and their principle of working is

different. The maximum marginal classifier (SVM) works by finding out the maximum margin to classify the data points. The node2vec method outperforms the other embedding methods. The DNGR and SDNE have almost the same performance.

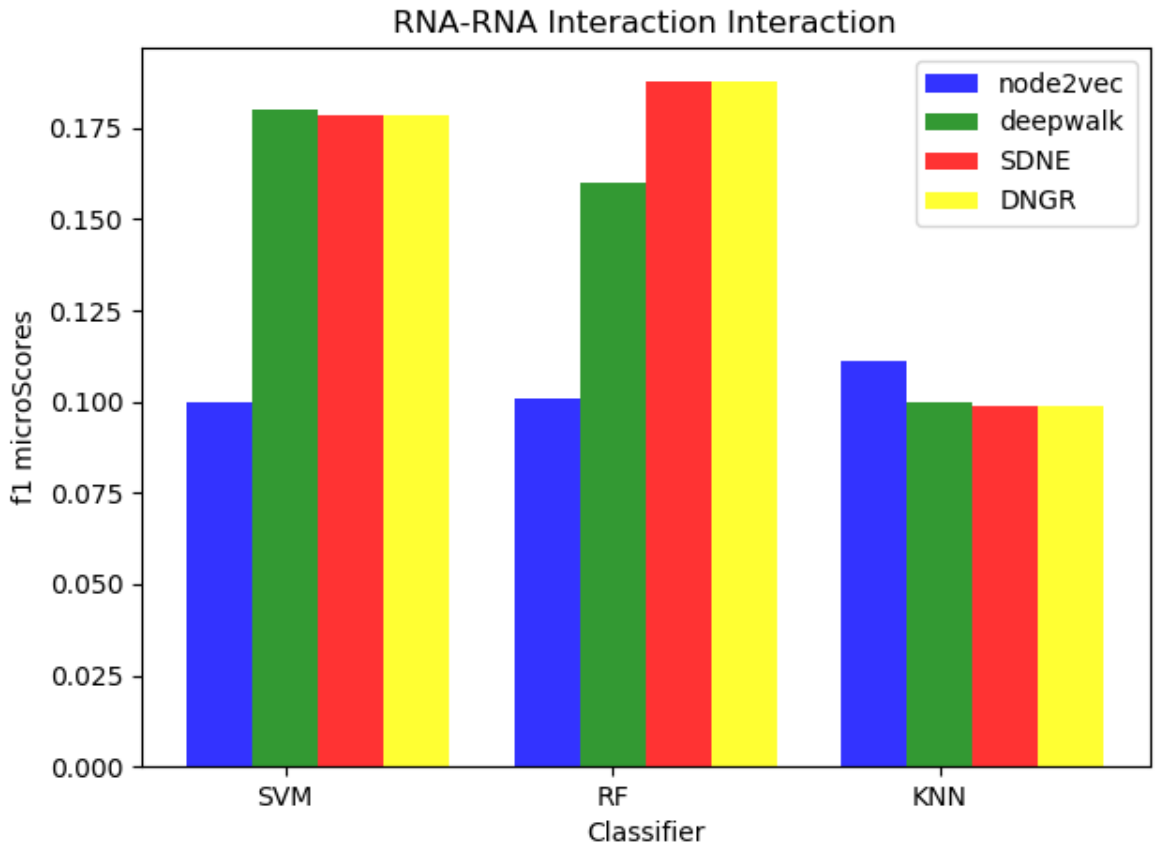


Fig 6.3: RNA-RNA Interaction – Embedding vs. f1-micro score

In the RNA-RNA interaction network, the evaluation is as follows in Fig 6.3. The embedding methods without random walks are outperforming the others for the Random Forest model.. And they are almost at the same score for all the models. And as the random walk based embedding is considered then the model SVM gives a larger score for deepwalk whereas the KNN yields good score for node2vec. Since the node2vec method uses a biased random walk for sampling the nodes it has a greater score. In the case of random forest and SVM, it is different. Random Forest work by constructing the multitude decision trees and the class will be the mode of the classes. That is why there occurs a change in the score for different models.

The below Table 6.1 shows the evaluation of different node embedding algorithms for a fixed dimension with respect to hamming loss.

Networks	Algorithm	Hamming Loss		
		SVM	KNN	Random forest
RNA-RNA Interaction	node2vec	0.99	0.90	0.95
	Deep walk	0.90	0.98	0.93
	SDNE	0.87	0.84	0.80
	DNGR	0.80	0.82	0.88
Disease -Disease interactions	node2vec	0.80	0.89	0.89
	Deep walk	0.76	0.71	0.733
	SDNE	0.80	0.81	0.83
	DNGR	0.72	0.76	0.72
Protein - Protein interactions	node2vec	0.52	0.50	0.53
	Deep walk	0.74	0.77	0.68
	SDNE	0.55	0.56	0.53
	DNGR	0.43	0.40	0.49

Table 6.1: Comparison of node embedding algorithm with respect to Hamming loss

The above table indicates the Hamming loss score for the four embedding methods node2vec, deep walk, SDNE, DNGR. For different models the scores are different. In the RNA-RNA interaction network, the node2vec method is having a more hamming loss than the others. For every classifier this is true. SDNE and DNGR are having almost the same

hamming loss since they work with more or less the same principle. As the disease-disease interaction network is analyzed the same is found. The node2vec embedding method has a greater hamming loss than deep walk. As non-walk based methods are considered then it is SDNE has more loss than DNGR. This result is evident from the above table. For the protein-protein interaction network, deep walk and SDNE have more loss than the other two. This result is for the dimension 64 and it is fixed for this evaluation. As the dimension changes the loss also changes. Our aim is to find out the minimum loss because the hamming loss finds outs the error between the actual and the predicted labels. Thus the evaluation proceeds like the above table.

6.2 EFFECT OF DIMENSIONALITY

The four node embedding methods are also evaluated by changing the dimension. It is illustrated below.

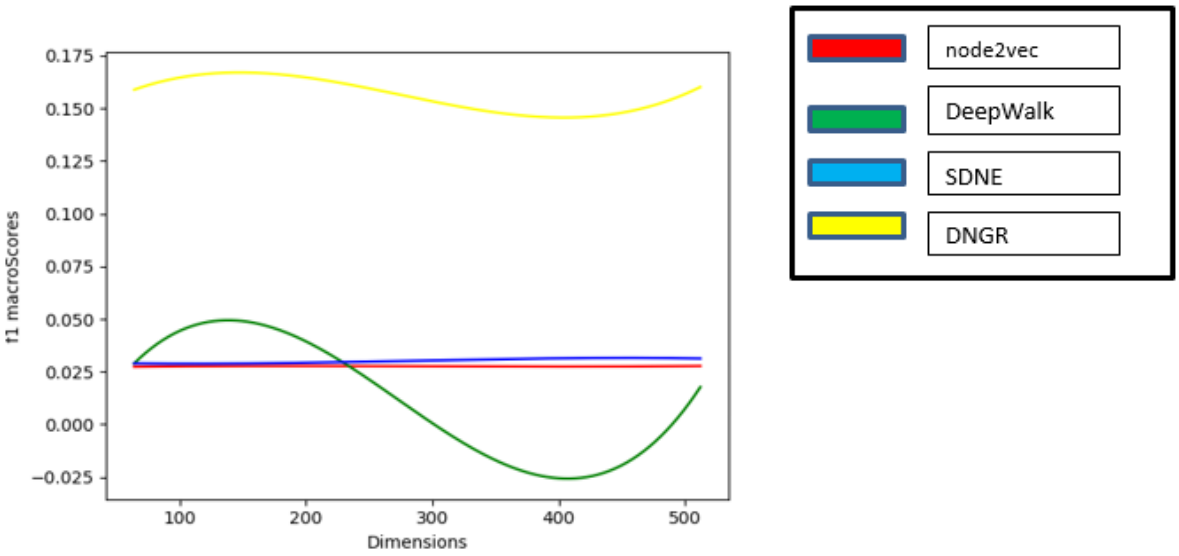


Fig 6.4: PPI Network- Dimensions vs. f1-score

The above shown Fig 6.4 shows the change in f1 score for the different embedding methods as the dimensions changes. As far as a network is considered the embedding of the nodes of the network has a significant importance in the dimensionality of the embedding. The graphs show that the f1 score changes rapidly as the dimensions changes. It is found that the node2vec method outperforms the other methods here. When considering the embedding

methods without random walks then SDNE is better than that of DNGR in case of this protein-protein interaction network.

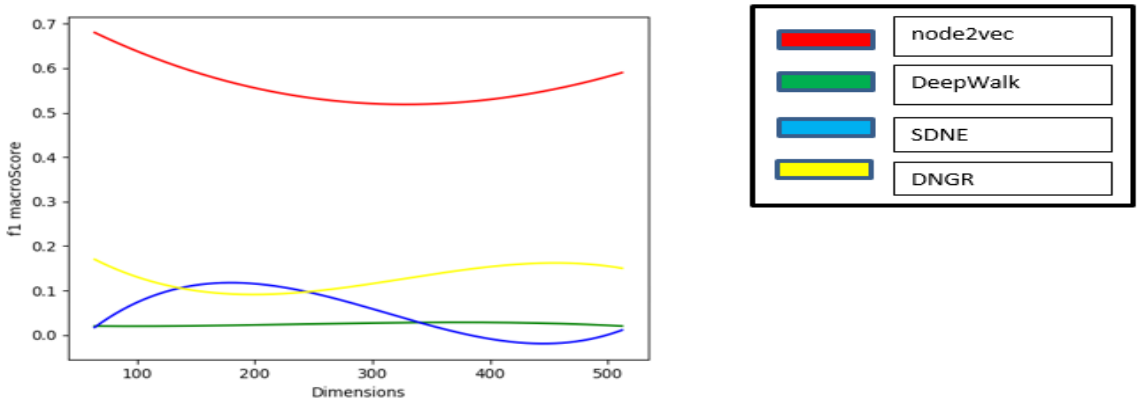


Fig 6.5: RNA-RNA Network- Dimensions vs. f1-score

The evaluation of dimension vs. f1 score in RNA-RNA interaction network is shown above Fig 6.5 and in this network, node2vec is outperforming the other methods. Due to the reason that it has the biased random walk for sampling the nodes in the network. At the lower dimensions, DNGR has more score than SDNE as dimension increases the score decreases. DeepWalk method is having a constant score in all the dimensions because of the random sampling strategy. DNGR is outperforming than SDNE because it optimizes the function for both the first and the second order proximity and then node embedding is done. Thus the node dependencies are all preserved.

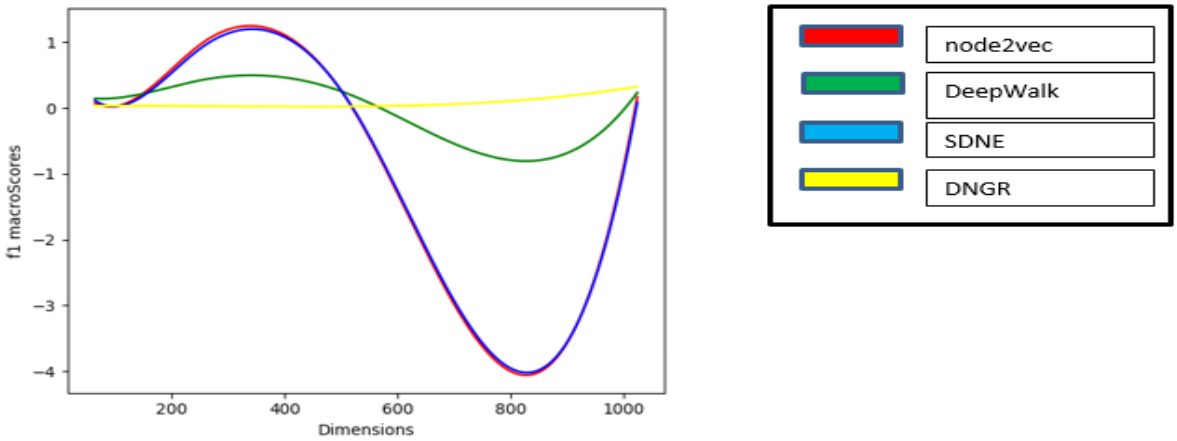


Fig 6.6: Disease-Disease Network –Dimensions vs. f1-score

The disease-disease network evaluation is shown above Fig 6.6 and from this, it is evident that the SDNE and the node2vec embedding is more efficient. Both these embedding methods are having a similar score in each of the dimensions. As the walk based methods are considered the score for the DeepWalk increases and then decreases suddenly as a dimension is increased, for the same dimension node2vec has a larger score than DeepWalk since it makes use of the biased random walk. The two proximities are preserved in node2vec. As the other two methods are considered SDNE performs well than DNGR. Both these methods use auto-encoders but a fair performance is achieved here by the SDNE method.

The below Table 6.2 indicates the evaluation of different node embedding method in disease-disease interaction network with respect to hamming loss and mean squared log error by changing the dimension.

Dimensions	**Models**	**Classifiers**					
		SVM		**Random Forest**		**KNN**	
		Hamming Loss	**Log error**	**Hamming Loss**	**Log error**	**Hamming Loss**	**Log error**
64	**Node2vec**	0.925	0.923	0.925	0.925	0.925	0.923
	DeepWalk	0.766	0.498	0.766	0.498	0.766	0.498
	SDNE	0.829	0.721	0.829	0.721	0.829	0.721
	DNGR	0.766	0.498	0.766	0.498	0.766	0.498
128	**Node2vec**	0.892	0.91	0.892	0.91	0.892	0.91
	DeepWalk	0.766	0.498	0.766	0.498	0.766	0.498
	SDNE	0.808	0.823	0.808	0.823	0.808	0.823
	DNGR	0.766	0.498	0.766	0.498	0.766	0.498
256	**Node2vec**	0.01	0.01	0.013	0.01	0.012	0.01
	DeepWalk	0.766	0.498	0.766	0.498	0.766	0.498
	SDNE	0.766	0.498	0.766	0.498	0.766	0.498
	DNGR	0.766	0.498	0.766	0.498	0.766	0.498

512	**Node2vec**	0.897	0.92	0.897	0.92	0.897	0.92
	DeepWalk	0.766	0.498	0.766	0.498	0.766	0.498
	SDNE	0.785	0.735	0.785	0.735	0.785	0.735
	DNGR	0.766	0.498	0.766	0.498	0.766	0.498
1024	**Node2vec**	0.831	0.841	0.831	0.841	0.831	0.841
	DeepWalk	0.766	0.498	0.766	0.498	0.766	0.498
	SDNE	0.780	0.66	0.780	0.66	0.780	0.66
	DNGR	0.766	0.498	0.766	0.498	0.766	0.498

Table 6.2: Comparison of node embedding methods for variable dimensions

In this above table, different node embedding algorithms are evaluated using two measures that is hamming loss and mean squared log error by changing the dimension of vector space. In different dimensions, the performance of these methods may slightly change. When random walk based methods such as node2vec and DeepWalk are concerned, node2vec outperforms DeepWalk in all dimensions because it uses the biased random walk. Also, SDNE is better than DNGR in non-random walk based method since it can learn more complex structure and preserve all the proximities.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The proposed system analyzed three different kinds of biological network such as PPI network, disease-disease network and RNA-RNA interaction network. Understanding the structure of proteins, finding the organ which is going to be affected by a disease and protein that relates to an RNA is useful for the development of drugs, disease diagnosis and so on. Thus, multi-label node classification or predicting the features of an unlabeled node in the biological network by considering already labeled node is very important in the present scenario. But, the traditional node embedding techniques faces several challenges when dealing with a large network having billions of nodes.

As the analysis is done, it found that the deep learning method can do the classification in a very efficient way. Thus, the proposed system made use of four different node embedding algorithm with four classifiers and these are evaluated using different evaluation metrics. Both the random walk based and the non-random walk based methods are studied and found that node2vec and the SDNE yields a good result. Since, node2vec uses biased random walk and able to preserve all the proximities, so that it outperforms DeepWalk. In case of SDNE and DNGR, both are non-random walk based method. It is able to infer that SDNE is better than DNGR in non-random walk based method since it can learn more complex structure and preserve all the proximities.

The future work aims to extend the classification for the heterogeneous networks. In the real world scenario, most of the biological networks are all heterogeneous in nature. These embedding techniques used work only in the homogenous networks. So the heterogeneous networks should be transformed to homogenous and then we have to apply them. An improved method to transform the heterogeneous networks to homogenous has to find out and then have to analyze the same methods for different networks.

BIBLIOGRAPHY

1. Josifoski, Martin, and Kire Trivodaliev. "Instance Based Learning in Protein Interaction Networks." (2017): 200-205
2. Can, Tolga, Orhan Çamoğlu, and Ambuj K. Singh. "Analysis of protein-protein interaction networks using random walks." *Proceedings of the 5th international workshop on Bioinformatics*. ACM, 2005.
3. Zhao, Zhehuan, et al. "A protein-protein interaction extraction approach based on deep neural network." *International Journal of Data Mining and Bioinformatics* 15.2 (2016): 145-164.
4. Goh, Kwang-Il, et al. "The human disease network." *Proceedings of the National Academy of Sciences* 104.21 (2007): 8685-8690.
5. Cui, Peng, et al. "A Survey on Network Embedding." *arXiv preprint arXiv:1711.08752* (2017).
6. Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.
7. Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016.
8. Wang, Daixin, Peng Cui, and Wenwu Zhu. "Structural deep network embedding." *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016.
9. Tang, Jian, et al. "Line: Large-scale information network embedding." *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015.
10. Bengio, Yoshua, Aaron C. Courville, and Pascal Vincent. "Unsupervised feature learning and deep learning: A review and new perspectives." *CoRR, abs/1206.5538* 1 (2012): 2012.

11. Rossi, Ryan A., Rong Zhou, and Nesreen K. Ahmed. "Deep feature learning for graphs." *arXiv preprint arXiv:1704.08829*(2017).
12. Goyal, Palash, and Emilio Ferrara. "Graph embedding techniques, applications, and performance: A survey." *arXiv preprint arXiv:1705.02801* (2017).
13. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013
14. Hamilton, William L., Rex Ying, and Jure Leskovec. "Representation Learning on Graphs: Methods and Applications." *arXiv preprint arXiv:1709.05584* (2017).
15. Aggarwal, Charu C., et al. "On Edge Classification in Networks with Structure and Content." *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 2017.
16. Cao, Shaosheng, Wei Lu, and Qiongkai Xu. "Deep Neural Networks for Learning Graph Representations." *AAAI*. 2016.
17. Cao, Shaosheng, Wei Lu, and Qiongkai Xu. "Grarep: Learning graph representations with global structural information." *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015.
18. Chen, Haochen, et al. "HARP: Hierarchical Representation Learning for Networks." *arXiv preprint arXiv:1706.07845*(2017)
19. Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).
20. Cao, Shaosheng, Wei Lu, and Qiongkai Xu. "Grarep: Learning graph representations with global structural information." *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015.
21. Phyu, Thair Nu. "Survey of classification techniques in data mining." *Proceedings of the International MultiConference of Engineers and Computer Scientists*. Vol. 1. 2009.
22. Zhang, Daokun, et al. "Network Representation Learning: A Survey." *arXiv preprint arXiv:1801.05852* (2017).

APPENDIX

1. Commands for installing Python 3

sudo apt-get install python3

pip install python 3

Python is the programming language used and can be installed via the above specified Command.

2. Command for installing matplotlib

Sudo apt-get install matplotlib

pip install matplotlib

Matplotlib i s a plotting library for the Python programming language and its numerical mathematics extension NumPy.

3. Command for installing gensim

Sudo apt-get install gensim

pip install gensim

Gensim is a robust open-source vector space modeling and topic modeling toolkit implemented in Python.

4. Command for installing networkX

Sudo apt-get install networkX

pip install networkX

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

5. Command for installing numpy

Sudo apt-get install numPy

pip install numpy

Numpy i s a library for the Python programming language, adding support for large, Multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays

6. Screenshots

6.1 Edge list of Protein-Protein Interaction (PPI) Network

1	1	1
2	2761	7
3	16000	17
4	1513	115
5	3311	1281
6	3995	1281
7	4199	1281
8	4404	1281
9	5961	1281
10	6090	1281
11	6901	4060
12	6980	1281
13	9004	1281
14	9218	1281
15	1525	115
16	9273	1281
17	11829	1281
18	16176	1281
19	6901	5964
20	5859	1282
21	2557	1283
22	2558	1283
23	3143	1283

6.2 Labels of PPI Network

```
A2M 1
UBQ 2
NLS 3
CC 4
SP 5
TM 6
HLH 7
ACTIN 8
IGFLMN 9
CH 10
SECTRIN 11
EF 12
GS 13
S_T_kinase 14
AAA 15
BBOX 16
LZ 17
RING 18
BRCT 19
SH2 20
```

```
1	1
2	2
2	3
2	3
2	3
2	3
2	3
2	3
2	3
2	3
2	3
2	3
2	3
2	3
4	4
4	4
4	4
7	5
7	6
7	6
7	6
7	6
7	5
7	6
```

6.3 Learned Feature Vectors

1	-0.086	0.419827	0.922664	0.764443	-0.65767	0.362476	-0.29219	0.618755	-0.02738	0.170071
2	-0.22769	-0.17805	0.264121	-0.07888	0.101978	0.181259	-0.23951	0.456873	0.119967	0.157813
2	-0.22769	-0.17805	0.264121	-0.07888	0.101978	0.181259	-0.23951	0.456873	0.119967	0.157813
4	-0.18164	0.346195	1.231242	0.373941	-0.85892	0.58487	0.434807	0.452659	-0.20789	0.333557
4	-0.18164	0.346195	1.231242	0.373941	-0.85892	0.58487	0.434807	0.452659	-0.20789	0.333557
10	0.027979	-0.79513	0.473771	0.166741	-0.43287	1.256955	-0.37003	-0.13912	0.63968	0.463051
10	0.027979	-0.79513	0.473771	0.166741	-0.43287	1.256955	-0.37003	-0.13912	0.63968	0.463051
10	0.027979	-0.79513	0.473771	0.166741	-0.43287	1.256955	-0.37003	-0.13912	0.63968	0.463051
10	0.027979	-0.79513	0.473771	0.166741	-0.43287	1.256955	-0.37003	-0.13912	0.63968	0.463051
15	-0.4625	-0.10766	0.280586	0.69541	-0.46041	0.880252	0.45052	-0.5489	0.287463	0.064237
16	-0.07636	-0.12175	0.309307	0.033758	-0.6123	-0.0619	0.556225	0.189766	-0.30878	0.179834
17	-0.13669	-0.28289	0.596349	0.387626	-1.21755	-0.06207	0.478191	-0.3263	-0.54886	-0.15834
17	-0.13669	-0.28289	0.596349	0.387626	-1.21755	-0.06207	0.478191	-0.3263	-0.54886	-0.15834
17	-0.13669	-0.28289	0.596349	0.387626	-1.21755	-0.06207	0.478191	-0.3263	-0.54886	-0.15834
18	-0.40277	-0.51088	0.352201	0.059628	-0.50898	0.585114	-0.20873	0.53002	-0.57196	0.123663
18	-0.40277	-0.51088	0.352201	0.059628	-0.50898	0.585114	-0.20873	0.53002	-0.57196	0.123663
19	-0.39448	-0.02059	0.343517	0.256917	-0.85291	0.44284	0.300988	-0.29374	-0.25925	0.071113
19	-0.39448	-0.02059	0.343517	0.256917	-0.85291	0.44284	0.300988	-0.29374	-0.25925	0.071113
19	-0.39448	-0.02059	0.343517	0.256917	-0.85291	0.44284	0.300988	-0.29374	-0.25925	0.071113
19	-0.39448	-0.02059	0.343517	0.256917	-0.85291	0.44284	0.300988	-0.29374	-0.25925	0.071113
20	-0.34454	-0.50555	0.67173	0.257218	-0.81324	-0.05157	0.563127	0.394326	-0.21835	0.407179
20	-0.34454	-0.50555	0.67173	0.257218	-0.81324	-0.05157	0.563127	0.394326	-0.21835	0.407179

6.4 Input for Multi-label node classification

0.536056	-0.12069	-0.51818	0.959599	0.434003	0.835217	-0.15354	0.388485	0.172449	1	1
0.536056	-0.12069	-0.51818	0.959599	0.434003	0.835217	-0.15354	0.388485	0.172449	2	1
0.651197	-0.50676	-0.17639	0.417188	-0.26663	0.004029	0.58942	0.062896	-0.48056	1	2
0.651197	-0.50676	-0.17639	0.417188	-0.26663	0.004029	0.58942	0.062896	-0.48056	1	3
0.863093	-0.08074	-0.51033	1.184638	0.409729	1.056793	0.198892	-0.06733	-0.48136	1	4
0.863093	-0.08074	-0.51033	1.184638	0.409729	1.056793	0.198892	-0.06733	-0.48136	2	4
0.389426	0.511844	-0.05723	0.52923	0.181199	0.123188	0.148168	-0.36483	0.628183	2	5
0.389426	0.511844	-0.05723	0.52923	0.181199	0.123188	0.148168	-0.36483	0.628183	2	6
0.389426	0.511844	-0.05723	0.52923	0.181199	0.123188	0.148168	-0.36483	0.628183	3	5
0.389426	0.511844	-0.05723	0.52923	0.181199	0.123188	0.148168	-0.36483	0.628183	3	6
-0.08617	-0.71308	-0.34702	0.723383	0.27404	0.60067	0.511351	-0.2784	0.172347	1	8
0.292735	-0.44101	-0.23999	0.245467	-0.13033	0.520042	-0.37281	-0.38247	0.16033	2	8
0.205463	-0.53995	-0.10871	0.725395	0.298528	0.163399	0.178738	0.034353	0.192023	4	8
0.205463	-0.53995	-0.10871	0.725395	0.298528	0.163399	0.178738	0.034353	0.192023	1	8
0.205463	-0.53995	-0.10871	0.725395	0.298528	0.163399	0.178738	0.034353	0.192023	2	8
0.333588	0.499715	0.14594	0.302313	0.323531	0.713776	-0.32081	-0.24642	0.205055	2	9
0.333588	0.499715	0.14594	0.302313	0.323531	0.713776	-0.32081	-0.24642	0.205055	2	10
-0.136	-0.12744	0.1502	0.643548	0.07155	0.90708	-0.462	-0.44046	-0.25862	2	4
-0.136	-0.12744	0.1502	0.643548	0.07155	0.90708	-0.462	-0.44046	-0.25862	2	10
-0.136	-0.12744	0.1502	0.643548	0.07155	0.90708	-0.462	-0.44046	-0.25862	2	11
-0.136	-0.12744	0.1502	0.643548	0.07155	0.90708	-0.462	-0.44046	-0.25862	2	12
-0.29485	0.188478	0.187668	0.14525	0.430379	0.733904	-0.25836	-0.62416	-0.41501	2	10
-0.29485	0.188478	0.187668	0.14525	0.430379	0.733904	-0.25836	-0.62416	-0.41501	2	11

6.5 Output

```
64
rf

mean_squared_log_error
0.992864232331
hamming loss
0.505114693118

average model score
0.493955362678
average macrof1 score
0.0666725727515
average microf1 score
0.493955362678
average weightedf1 score
0.421240863602
KNN

mean_squared_log_error
1.0824740541
hamming loss
0.548977061376
```