# Node Influence Prediction in Complex Networks

*Towards Network Embedding based Features*

by
Rommy Gobardhan

TU Delft

# Node Influence Prediction in Complex Networks

## Towards network embedding based features

by

Rommy Virindra Gobardhan

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday November 22, 2021 at 11:00 AM.

Student number: 4124227

| Thesis committee: | | |
|---|---|---|
| | Dr. ir. Huijuan Wang | TU Delft, Chair |
| | Prof. Dr. Ir. Rob Kooij | TU Delft |
| | PhD Shilun Zhang | TU Delft, Supervisor |




TU Delft

# Abstract

The study of epidemic spreading processes on contact based complex networks has gained a lot of traction in recent years. These processes can entail a variety of problems such as disease spreading, opinion spreading in social networks or even airport congestion in airline networks. One of the key tasks in this area of research and also of this work is the prediction of the final epidemic size of an outbreak in a network, given that a contagion process has been initiated by a seed node. More specifically, the objective is to predict to which extent a seed node is able to activate the rest of the nodes in a network using a supervised learning model. In this work, this task is termed: “The node influence prediction problem”. Being able to predict the epidemic footprint of a node allows the design of robust networks and the application of efficient intervention strategies.

Recently, a limited number of studies have proposed methods on how to utilize classical network topology based features to predict the nodal influence. However, two main challenges still persist: (1) individual topology based features do not fully capture the information of a node and (2) it is tedious to obtain these features for nodes in large scale networks. As an alternative solution, this work aims to utilize network embedding based features instead, where feature vectors of the nodes are *learned* from the network topology. In this research we assume that the network topology and the nodal influence of a small subset of the nodes are known. We then proceed to show how to build and optimize a machine learning framework where only 10% of the nodes are used as training data and which could even be applicable on large scale networks. Additionally, we also demonstrate why network embedding based features are applicable in the node influence prediction task.

The findings show that node pairs which are closer in proximity in the network, are also embedded closer in the embedding space (exhibiting a higher similarity). The performance evaluation of the predictive models illustrate that network embedding based features can compete with classical topological metrics, despite the disadvantage of their higher dimensionality. This is achieved by combining the embedding features with individual low cost topology features such as the degree.

# Preface

After having graduated once in the field of Chemical Engineering, I still remember of having a desire to learn more about programming. At that time, I only had a limited experience with MATLAB and I thought: What if I could use the tools from Computer Science as a Chemical Engineer as well? I never thought that after four years I would be at the point to graduate once again as a Computer Scientist by investigating how to predict influential nodes in complex networks! This thesis project marks the end of that journey which had its ups and downs. In these tough times (from which the world is still recovering from), this thesis project was conducted mainly with the support and guidance from several people, to whom I would like to express my gratitude. To my daily supervisors *Dr. Huijuan Wang* and *PhD Shilun Zhang*, I am thankful for your guidance, patience and scientific knowledge during the many meetings that we had. I have learned that performing experimental work is not a straight point from start to finish. Instead, it takes trial and error to define a good piece of work. To Prof. Rob Kooij I would like to express my gratitude for being part of my thesis committee. I am also grateful for my family and friends who have supported me in all the aspects outside of the university throughout these years.

*Rommy Virindra Gobardhan*
*Delft, November 2021*

# Contents

# 1 Introduction

This chapter provides a brief description of the topic to the reader. Section 1.1 explains the concept of complex networks and its utility in both research and society in the context of spreading processes. After listing some of the advancements, a short summary is presented with the current gaps in research. The main focus of this work and its associated research questions are addressed in Section 1.2. Lastly, this chapter is concluded with a list of expected contributions (Section 1.3).

## 1.1. Motivation

The world is becoming increasingly dependent on complex systems, where each may consist of a set of interacting components. For example, a set of airports communicating with each other in order to optimally schedule its flights, a set of genes in a cell collectively regulating its biological processes or even the electrical power grid where failures are mitigated by efficient power rerouting, are among others some of the most widely studied systems [8, 61, 4]. The common factor among these complex systems is the inherent network structure. Here, the system level components are identified as nodes and the interactions between these components as links. As a result, a dataset can be created with the ”wiring diagram” representing the system architecture. Real world systems with such a characteristic are called *complex networks* and have been studied from a network science perspective in several disciplines for good reason: investigating the underlying network topology allows us to better understand, design and control those systems [1, 14, 60].

An active area of research is the study of epidemic spreading processes on contact based networks. Dynamic processes can entail a variety of problems such as disease spreading, opinion spreading in social networks or even airport congestion in airline networks [2, 64, 39, 8, 44]. In order to research these phenomena, the Susceptible-Infected-Recovered (SIR) epidemic model is commonly used, where nodes can occur in any of those three states at any given time. The key idea is that a susceptible node can become infected if it came into contact with another infected node. After being infected for a while, the node can recover and therefore become immune to the disease. When an epidemic process initiated by a seed node, unfolds on a network, there are two possible outcomes: (1) the infection does not cause an outbreak or (2) an outbreak occurs, in which case a significant fraction of the nodes will be affected by the disease. In both cases, the infection will terminate when all infected nodes have been recovered. An important quantity of interest in epidemiology and also in this work is the expected value of the fraction of nodes recovered in the final stable state. This quantity (also termed the final epidemic size) reflects the *influence of the seed node* and it varies per seed node. That is, each node that initiated an infection (or any other diffusion process) in the network has a different impact on the network [24]. The influence of a node has several utilities depending on the type of network in question. Some are listed as follows [33]:

- In physical contact and infrastructure networks, it can be used to control the outbreak of a disease by devising intervention strategies.
- In social networks it can be used to prevent the spreading of misinformation (such as fake news).

- In the computer router and electrical networks it can be used to prevent failures by isolating the influential nodes.

The majority of the studies conducted so far have focused on ranking the nodes in the network by their nodal topological properties. More specifically, the goal has been to identify the top $k$ fraction of the highest influential nodes. On the other hand, little attention has been paid to quantify the magnitude of the influence of a seed node by utilizing the network topology and the influence of a small set of the nodes (*i.e.* the node influence prediction task). Work conducted in [7, 6, 65, 46, 36] and [57] have made attempts to (1) identify which topological features in a network are indicative of the nodal influence and (2) build predictive models. However, several shortcomings exist. First, multiple studies have proposed prediction frameworks based on nodal topological features with the following drawback: the influence of a large fraction of the nodes in the network should be known beforehand. While this provides insight for feature engineering, the framework itself may not be applicable in practice where data is limited. Second, the influence of a node as defined by the SIR epidemic model is dependent on the infection rate $\beta$ and recovery rate $\mu$. Across several studies conducted so far, the dataset is generated with different values for these parameters. As a result, positioning the results of each work poses yet another challenge. Lastly, most of these studies have used classical network topology based features to predict the nodal influence, which might not fully capture the information of a node. In addition, when dealing with large scale networks with millions of nodes, some of these features can be computationally expensive to be obtained. This poses yet another major challenge.

To allow graph analysis on large scale networks, network embedding techniques have been developed and proven to be effective solutions [19, 42, 63]. In network embedding, each node in the network is mapped to a low-dimensional vector space where the distance between two nodes represent some proximity measure in the original network topology [12]. As a result, each node is represented by a $d$ dimensional *learned* feature vector. We see the potential of using the embedding vectors of nodes to perform the task of the node influence prediction: In the first step, the optimal embedding of a network is generated such that it best preserves its topology structure. This is achieved by optimizing the network embedding parameters with respect to the link prediction task [19]. In the second step, the learned features are used as input in a prediction framework to estimate the influence of the nodes in the network. This has the advantage of not needing to manually handcraft features as is the case with the traditional network topology based features.

In short, the task of the node influence prediction with limited data has not received enough attention yet, especially where embedding based features are utilized. Furthermore, the main focus of the limited number of studies in this area has been the identification and utility of well performing classical topology based features. As networks grow to millions of nodes, there is a need to incorporate network embedding based features into the prediction framework. While network embedding based features seem promising for the given supervised learning task, a prior investigation is needed in order to understand whether the pairwise nodal similarity in the network embedding reflects some network topology based distance measure. An embedding which preserves the distance of node pairs in the network topology can be especially effective in the node influence prediction task, as nodes which are closer in the topology could have a similar influence.

## 1.2. Project Scope

The aim of this thesis project is to address the previously mentioned needs. Therefore, a predictive model is designed especially for the setting where limited data is available. In this case, it is assumed that the network topology and the influence of a subset of its nodes are known. The predictive model utilizes properties of nodes in the given subset and its influences, in order to predict the influence of the remaining nodes in the network. In order to achieve this feat, two types of features are extracted from the network topology and compared: **network centrality metrics** and **network embedding based features**. A second objective is to perform an exploratory study on the optimal network embedding to determine how the distance in the embedding space is related to the proximity in the network topology. Based on the objectives, the following research questions can be identified:

1. **How can network embedding based features be utilized in order to predict the information diffusion capability (influence) of a node in a network?**
   To answer this research question, it is divided into the following sub questions:

(a) **How is the proximity between the nodes in the network topology captured within in the network embedding, which optimally preserves the network structure?**

(b) **In the presence of limited training data, how effective are network embedding based features in contrast to the classical network topology based features in the node influence prediction task?**

(c) **Does the incorporation of the network topology based features into the network embedding based prediction models improve the prediction of the nodal information diffusion capability?**

As previously mentioned, network embedding methods represent each node in the network by a learned embedding (feature) vector, whose utility has been proven to be useful in various supervised learning tasks [12]. Research question **A** addresses the motivation why these nodal embedding features may also be applicable for the main objective of this work: predicting the nodal influence using its embedding features. This utility is also based on the premise that nodes which are closer in proximity in the network topology possibly have a similar influence [19]. Therefore, it is hypothesized that a network embedding in which the pairwise nodal similarity is correlated with their corresponding proximity in the network topology, could be effective when it is used in the node influence prediction task. Once it has been investigated how the network embedding captures the nodal proximity in the network topology, research question **B** will address how well the embedding can be utilized to predict the nodal influence. In order to achieve this, its prediction performance is also compared to baseline models where classical network topology based features are utilized. Finally, research question **C** will help answer whether further optimization of the network embedding features with the classical network topology based features is useful to the objective of this work.

## 1.3. Contributions

The main contributions of this thesis project can be classified into two categories:

1. As stated before, little attention has been paid into unraveling the relation between the network embedding and the topological properties of a graph $G$. In several applications, the link prediction task is used to produce an optimal embedding which in turn is applied to downstream application tasks. With a systematic analysis on the correlations between the cosine similarity (between any two nodes in the embedding space) and the shortest path distance (between any two nodes in the network topology), it is shown that the network embedding does preserve the shortest path distance to some extent.

2. After the exploratory analysis on better understanding the nature of the embedding vector of a node, a prediction framework has been constructed which utilizes the network embedding based features in order to predict the nodal influence in a setting where limited training data is available. Furthermore, it is investigated when the utility of the network embedding based features is beneficial in comparison with the baseline: the utility of classical network topology based features.

## 1.4. Outline

This report is structured as follows. Chapter 2 introduces the reader to the theory behind complex networks, the network embedding algorithm and the task description of the proposed method for node influence prediction. In addition, several research gaps are identified based on a literature review. To answer the research questions of this thesis project, a set of experiments have been defined. The details on how these can be reproduced is found in Chapter 3. Chapter 4 presents: (1) the investigation of how the distance metrics in the embedding- and topology space are related and (2) performance evaluation of the proposed node influence prediction framework. Insights gained from these experiments are then discussed, with concluding remarks in Chapter 5. Finally, a set of follow up experiments are presented for future work in Chapter 6

# 2

# Background

This chapter aims to introduce the reader to the several topics discussed in this work. As a start, the formal definition of graphs are discussed with a focus on their use, advantages and disadvantages (**Section 2.1**). Afterwards, the theory on the background regarding the two main objectives of this work is elaborated on:

1. **Concept and properties of the network embedding (Section 2.2):** With an emphasis on distance measures between node pairs in the embedding- as well as the topology space. **Node2Vec**, one of the most commonly used embedding method is also explained in detail.

2. **Node Influence Prediction using Network Embedding techniques (Section 2.3):** Emphasizing the specific task description and the state of the art on the existing prediction methods.

## 2.1. Networks

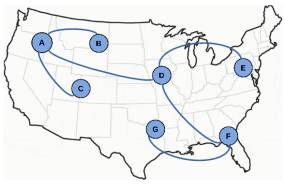
As described in Chapter 1, examples of complex networks (CN) can be found across various disciplines, each being a system with its own set of interacting components. For analysis, these systems are mathematically defined as **Graphs** by identifying its components as **Nodes** (or **Vertices**) and interactions between these components as **Links** (or **Edges**). Figure 2.1 presents a simplified example of such a CN (US airline network [8]). Airports on the geographical map have been identified as nodes and connecting flights between any two airports as links (Figure 2.1a). It should be noted that CN in the real word are rather large (thousands if not millions of nodes) and evolve over time. In this specific example the number of nodes is kept small for clarity and the CN is assumed to be static. Abstracting away the domain properties of the CN and by only considering the existing nodes and links, the topology representation is obtained as shown in Figure 2.1b. Formally, it can be defined as follows [58, 1]:

**Definition 1.** *Let **G(V, E)** denote a graph with vertex set **V** and undirected edge set **E**:*

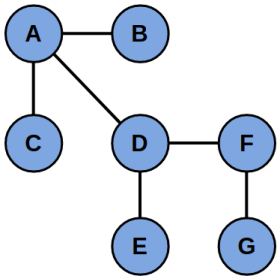
- $V = \{v_1, v_2, \ldots, v_N\}$*, where* $N = |V|$ *and* $v_i$ *denotes vertice* $i$*.*
- $E = \{(v_i, v_j) \mid i, j = 1, 2 \ldots, N\}$*, where* $(v_i, v_j)$ *is an edge in **G** between vertices* $v_i$ *and* $v_j$*.*

### 2.1.1. Adjacency Matrix

A mathematical representation of a graph topology is found by encoding the set of edges **E** as an $N \times N$ square adjacency matrix **A**, where $N$ denotes the number of nodes (Figure 2.1c). The elements $a_{ij} = 1$ if the edge $(n_i, n_j)$ exists between the nodes $n_i$ and $n_j$, otherwise they are set to $0$ [27]. Assuming that, (i) each $a_{ij}$ is either $0$ or $1$ (binary), (ii) each $a_{ij} = a_{ji}$ in matrix **A**, and (iii) self loops do not exist, the graph is said to be undirected and unweighted. All real world CN investigated in this work are of this graph type.




(a) Complex Network.




(b) Graph Topology representation.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| A | **0** | 1 | 1 | 1 | 0 | 0 | 0 |
| B | 1 | **0** | 0 | 0 | 0 | 0 | 0 |
| C | 1 | 0 | **0** | 0 | 0 | 0 | 0 |
| D | 1 | 0 | 0 | **0** | 1 | 1 | 0 |
| E | 0 | 0 | 0 | 1 | **0** | 0 | 0 |
| F | 0 | 0 | 0 | 1 | 0 | **0** | 1 |
| G | 0 | 0 | 0 | 0 | 0 | 1 | **0** |

(c) Adjacency matrix representation.

Figure 2.1: Example of a complex network (US airline network adapted from [8]) with a possible topology- and adjacency matrix representation.

### 2.1.2. Network Analysis

Traditionally, the adjacency matrix $A$ of a graph has been mainly utilized to calculate various topological metrics used in graph analysis tasks. A taxonomy and description of the most widely used metrics is shown in [21] and [31]. Next, a brief description is given for the topology metrics used in this thesis project. The motivation behind the choice of these features are given in Section 2.3 instead.

#### Degree Centrality

The degree centrality ($d_u$) of a node $u$ is the number of directly connected neighbours it has, normalized by the number of nodes in the graph. This quantity can be computed using the adjacency matrix $A$ as follows:

$$d_u = \frac{1}{N-1} \sum_{k=1}^{N} a_{uk} \tag{2.1}$$

where $a_{ij} \in A$.

#### Closeness Centrality

The closeness centrality ($c_u$) of a node $u$ is a measure of the average distance between $u$ and all the other nodes in the graph. It denotes the capability of $u$ to exchange information between itself and the rest of the nodes and is computed as follows:

$$c_u = \frac{N-1}{\sum_{u \neq k} H_{uk}} \tag{2.2}$$

where $H_{uk}$ denotes the hopcount of the shortest path between the nodes $u$ and $k$.

#### Eigenvector Centrality

The eigenvector centrality ($x_u$) of node $u$ is a measure which takes into account the importance of its neighbours, in addition to its degree. Thus, it is interpreted as a weighted degree [58]. This metric can be computed using:

$$x_u = \frac{1}{\lambda} \sum_{k=1}^{N} a_{uk} x_k \tag{2.3}$$

where $\vec{x}$ denotes an eigenvector associated with the eigenvalue $\lambda$. In order to ensure that all the components in $\vec{x}$ are positive, the eigenvector associated with the largest eigenvalue is chosen.

#### Resistance Distance

The shortest path distance ($H_{uk}$) is a distance metric that only considers a single path between the nodes $u$ and $k$. In contrast, the resistance distance ($\Omega_{uk}$) also takes into account multiple paths. It is based on the premise that the ”effective distance” decreases, if there are more paths between nodes $u$ and $k$, not necessarily limited to shortest paths [25]. This is relevant in graphs in which flow is transported between two nodes. If there are more paths to be utilized, the smaller the resistance. The

resistance matrix $\Omega$ with all pairwise $\Omega_{uk}$ can be computed as follows:

$$\Omega = \vec{z} \cdot \vec{u}^T + \vec{u} \cdot \vec{z}^T - 2\hat{Q} \tag{2.4}$$

where $\hat{Q}$ is the pseudo inverse of the Laplacian matrix $Q$ of graph $G$, $\vec{z}$ is the vector containing the diagonal elements of $\hat{Q}$ and $\vec{u}$ the all one vector [58].

### 2.1.3. Limitations

While the traditional approach of graph characterization based on topology metrics has proven its utility across several inference tasks, it is limited to small scale graphs [3, 35, 13]. The analysis of large scale graphs ($N > 10^4$) using the adjacency matrix $A$ poses the following challenges (some may also apply to small scale graphs):

- **Manually handcrafting features is tedious:** As described in Section 2.1.2, using the graph topology one can compute various topology metrics. Determining the most effective metric(s) for a given application task is challenging as one would have to consider every single metric. Furthermore, it has been shown in the work by Li ***et al.***that these topology metrics are correlated and dependent. The dependency structure between the metrics may also change with varying graph topology [31, 29].
- **Computing features are intractable for large scale graphs:** A subset of graph metrics, in particular from the distance class are computationally expensive. For example, the closeness centrality metric requires the computation of the shortest path length between all possible node pairs. This is intractable for large scale graphs. Another example is the principal eigenvector component which is computationally expensive since it is computed iteratively for convergence [12].
- **The adjacency matrix $A$ cannot be used directly as input for classification and prediction problems:** Machine learning algorithms require each observation in the dataset to be independent. This is not the case for nodes in a graph. Each node is related to a subset of other nodes as specified by the edge set $E$. CN are in general sparse with a small average degree [2]. As a result, the majority of the elements in $A$ are zeros denoting that the corresponding row vector of a node is not suitable as a feature vector for machine learning applications.

## 2.2. Network Embedding

An alternative option to allow network inference on large scale graphs is to *learn* a low-dimensional vector representation for each node in the graph. Figure 2.2 illustrates this concept, which is also referred to as ***Network Embedding*** in the network science community. The objective of this approach is to represent each vertice in an alternative latent vector space, where the distance between vertices encode some task specific property (this depends on the embedding algorithm used). In the last decade, several embedding algorithms have been developed which can be classified in three main categories [12, 11, 18, 62, 10]:

1. **Factorization based methods:** In matrix factorization based methods, the aim is to learn a low rank approximation of an input matrix representing a graph. This input matrix can be the adjacency- or the laplacian matrix. Techniques such as Singular Valued Decomposition (SVD) and Principal Component Analysis (PCA) are often used to produce embeddings. The key characteristic of this class of embedding methods is that it is considered as a dimensionality reduction technique [9, 11].
2. **Random walk based methods:** In random walk based methods, the aim is to approximate node similarity by utilizing random walks. Given a starting node $u$, its neighbourhood $N_u$ can be defined as a sequence of nodes which are sampled using some sampling strategy **S**. For any node $v \in N_u$, the embedding vectors $\vec{e_u}$ and $\vec{e_v}$ of the nodes $u$ and $v$ respectively, are then optimized such that the dot product of $\vec{e_u}$ and $\vec{e_v}$ approximates the probability that $v$ occurs in $N_u$. Frameworks that are based on this method are **Deepwalk** and **Node2Vec** [42, 19](see Section 2.2.3).

3. **Deep learning based:** Deep learning based methods are yet another class of algorithms especially aimed to capture highly nonlinear structures in the network [12, 59]. Furthermore, deep learning based frameworks can be constructed that provide end to end solutions to the application task. In this case, instead of learning embedding vectors and applying off the shelf ML methods, the network is directly used as input for the task at hand [30].

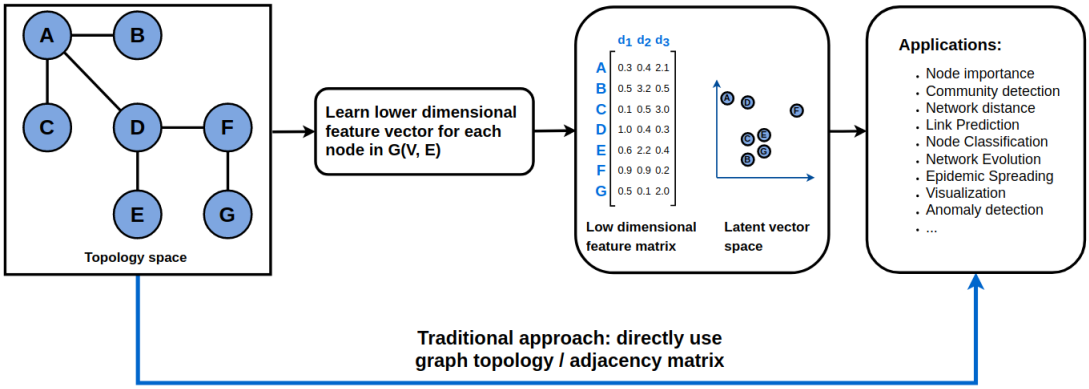



Figure 2.2: Pictorial overview of the concept of network embedding. The traditional approach of using the graph topology is compared with the more recent approach of using a network embedding method to map the nodes in the graph onto a lower dimensional vector space (adapted from [12]).

### 2.2.1. Properties of the Embedding Space

An embedding of a graph is defined as follows [43]:

**Definition 2.** *Given a graph G(V, E) with vertex set V and edge set E, its corresponding embedding $G_E$ is a mapping function $f : V \rightarrow \mathbb{R}^d$, where each vertex $v_i \in V$ is mapped to a $d$-dimensional, dense and continuous vector with the following properties:*

- $d \ll |V|$.
- *$f$ preserves some distance measure between node pairs of graph G in the embedding space, i.e., similar nodes in the topology space should be embedded closer in the latent vector space.*

A popular metric used to compare vertices in the embedding space is the **Cosine Similarity**, which is essentially the cosine of the angle of the corresponding embedding vectors of the vertices. Let $\vec{a}$ and $\vec{b}$ denote the embedding vectors for two vertices, then the cosine similarity is defined by Equation 2.5. In this work, this metric is utilized for two objectives, (i) assessing the link prediction performance when using an embedding, and (ii) for studying its correlation with the topology metrics mentioned in Section 2.1.2.

$$Cosine(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \tag{2.5}$$

### 2.2.2. Research Gap

The invention of the random walk based embedding methods and the more recent advances in deep learning based methods have proven their utility across a wide range of applications [12, 19]. However, a theoretical understanding on how the embedding vector space is related to the graph topology has yet to be uncovered. One particular aspect open to research is the correlation between the distance based metrics in the embedding- and the graph topology space. Given a network, either its topology ($G_T$) or embedding ($G_E$) can be used in network inference tasks. For the network embedding to be effective, it should at least be able to preserve the network topology [12]. That is, node pairs which are immediate neighbours or positioned closer in the topology, should be embedded in closer proximity. Figure 2.3 illustrates this concept with an example schematic. Let $S_T(X, Y)$ and $S_E(\vec{e_X}, \vec{e_Y})$ denote a distance (or

similarity) measure between the nodes $X$ and $Y$ in the topology- and embedding space, respectively. Then, by investigating the correlation between $S_T(X,Y)$ and $S_E(\vec{e_X},\vec{e_Y})$ for every node pair $X$ and $Y$, insight can be gained on how the topology of the network is captured in the corresponding embedding.

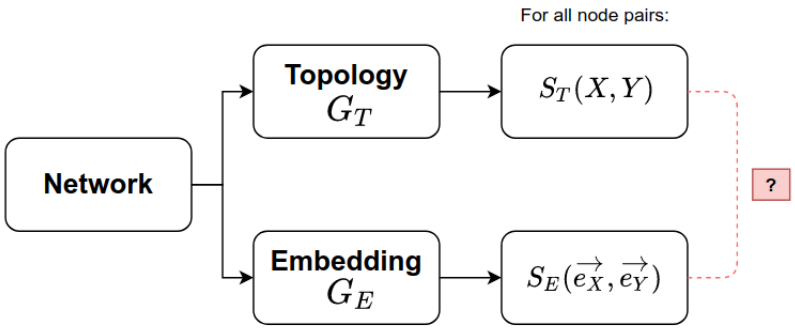



Figure 2.3: Schematic diagram of a network, whose topology- and embedding related properties are reflected in the pairwise distance (similarity) measures $S_T(X,Y)$ and $S_E(\vec{e_X},\vec{e_Y})$ for the topology- and embedding space, respectively.

### 2.2.3. Node2Vec

**Node2Vec** is a random walk based framework, developed by Grover *et al.* in an attempt to express the node similarity in the embedding space by incorporating not only local- but also higher order neighbourhood information [19]. Its key improvement over previous well performing frameworks such as **LINE** [53] and **Deepwalk** [42] is that it generates node sequences using *biased* random walks. The Node2Vec embedding algorithm can be divided into three steps, which are discussed next [28].

**Step 1: Sample fixed length short random walks**

Let $W_u$ denote a random walk that starts in node $u$ with a length of $k$. Then, nodes are added to $W_u$ by sampling one of the neighbours of the last visited node, until $\|W_u\| = k$. There are two strategies to sample a node: (i) using Breadth First Search (BFS), and (ii) using Depth First Search (DFS). These strategies are visualized in Figure 2.4a. In BFS, nodes are sampled from the neighbours of $u$, aiming to preserve a local microscopic view of node $u$. In contrast, DFS samples nodes at increasing distance from $u$. Thus, a macroscopic global view is obtained of node $u$. In order to combine both these search strategies, the authors designed a biased $2^{nd}$ order random walk method where two parameters can control to which extent nodes are sampled in a BFS or a DFS manner. In the $2^{nd}$ order random walk, the next node transition takes into account not only the last visited node $v$, but also the node visited before $v$, called $t$. Figure 2.4b presents this principle where the random walk started in some node $u$, visited node $t$ and currently is at node $v$. Let $\alpha_{tx}$ denote the unnormalized transition probability, then:

$$\alpha_{tx} = \begin{cases} \frac{1}{p}, & \text{if } d_{tx} = 0 \\ 1, & \text{if } d_{tx} = 1 \\ \frac{1}{q}, & \text{if } d_{tx} = 2 \end{cases}$$

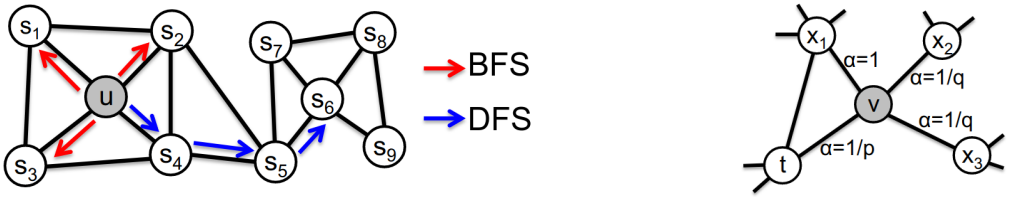
where $d_{tx}$ is the shortest path distance between nodes $t$ and $x$, $p$ the return parameter and $q$ the in-out parameter. Depending on the choice of $p$ and $q$, the next node is sampled from $\{t, x_1, x_2, x_3\}$ with a different (biased) transition probability. By tuning $p$ and $q$ it is thus possible to capture both the local- and higher order neighbourhood information in each random walk $W_u$.

**Step 2: Construct the neighbourhood set**

In the next step, the previous random walk generation procedure is used to concretely define the dataset of the neighbourhood of each node $u \in V$. To reduce any implicit bias induced by the starting node $u$, multiple walks are generated for $u$. Thus, the dataset for each node $u$ consists of $r$ biased random walks where each walk has a length of $k$. In short, $N_u = \{W_u^1, W_u^2, ..., W_u^r\}$.

**Step 3: Optimize the embedding such that it encodes the statistics of the random walks**

In the last step, the previously generated dataset for each node is used in the (extended) Skip-gram




(a) Concept of flexible random walks using Breadth First Search (BFS) and Depth First Search (DFS).

(b) Biased random walk procedure to approximate the BFS and DFS search strategy.

Figure 2.4: Search strategy used in Node2Vec to sample nodal neighbourhood [19].

model to generate the embedding vector for each node [37]. The trick is to formulate the feature representation learning task as an optimization problem where techniques such as Stochastic gradient Descent (SGD) can be used fro optimization. More precisely, the goal is to find an embedding vector $\vec{e_u}$ of a node $u$ such that it can predict the neighbourhood $N_u$ by utilizing the dataset containing multiple instances (examples) of neighbourhoods. Therefore, the following objective function is maximized:

$$\max_{\vec{e_u}} \sum_{u \in V} \log P(N_u | \vec{e_u}) \tag{2.6}$$

where $P(N_u|\vec{e_u})$ denotes the log probability of observing neighbourhood $N_u$ given the embedding vector $\vec{e_u}$. After incorporating the assumption of conditional independence and using a softmax parametrization step, the fully derived loss function is given by:

$$\max_{\vec{e_u}} \sum_{u \in V} \sum_{x \in N_u} \log \left( \frac{\exp(\vec{e_u}^T \vec{e_x})}{\sum_{n \in V} \exp(\vec{e_u}^T \vec{e_n})} \right) \tag{2.7}$$

One drawback of this derivation is that the normalization term in the denominator iterates once again over all the nodes in the network. As a result, the run time complexity becomes $O(|V|^2)$. To address this issue, negative sampling is used to estimate the normalization term. A detailed derivation of this loss function can be found in [19] and [28]. Apart from the fact that a richer notion of neighbourhood can be expressed in the embedding vectors, Node2Vec has the advantage of parallel execution.

### 2.2.4. Topology preservation and link prediction

In the link prediction task, the goal is to determine the likelihood of the existence of a link between any two nodes in the network [32]. Let $E$ denote the set links of an observed graph $G$. Clearly, $E \subseteq L$, where $L$ denotes the set of all possible links that can exist in $G$. For the observed graph, $L - E$ denotes the set of links that can be either missing or will appear in the future (in case the network is changing over time). The link prediction task aims to identify these links. This concept is illustrated in Figure 2.5. Several methods exist to perform link prediction such as (i) similarity based algorithms, (ii) maximum likelihood based, and (iii) probabilistic models. Lu *et al.* summarizes the progress of each of these methods in [34]. Another utility of the link prediction task nowadays is to *assess the quality of an embedding framework* that aims to preserve the network structure. This is precisely how the link prediction task is used in this thesis project: to quantify to which extent an embedding produced by Node2Vec, preserved the inherent network structure of a graph. The intuition is that an optimal learned embedding should be able to reconstruct the network it was trained on [59]. The exact procedure of this application is described in Section 3.2.

One method to perform the link prediction using network embedding is to assign a similarity score $S_{xy}$ to each unobserved link $l_{x,y} \in \{L - E\}$ between the nodes $x$ and $y$. In this project, $S_{xy}$ is set to the Cosine Similarity of the corresponding embedding vectors $\vec{e_x}$ and $\vec{e_y}$ of the nodes $x$ and $y$, respectively. As a result, a higher similarity in the embedding space between the vectors $\vec{e_x}$ and $\vec{e_y}$ denotes a larger likelihood of a link to exist between the nodes $x$ and $y$ in the original network. Generally, the method to assess the embedding reconstruction performance for a given network $G(V, E)$, proceeds as follows:

1. Remove a small fraction of the links in $G$, while ensuring that the resulting network ($G_{train}$) remains connected. The removed links are denoted as *positive* samples.

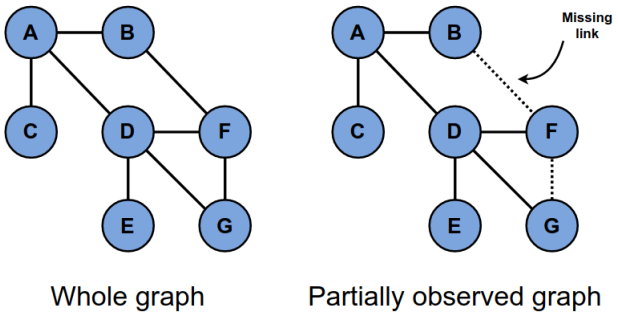



Figure 2.5: Pictorial representation of a typical problem occurring in practice. On the right, the observed graph is missing two links. The complete or evolved graph is shown on the left. Figure adapted from [34].

2. Sample a small fraction of links from the set of nonexistent links $\{L - E\}$ and denote this as *negative* samples.
3. Use $G_{train}$ to produce an embedding $E_{G_{train}}$ using some learning framework.
4. Evaluate the link prediction capability of $E_{G_{train}}$ using the *positive-* and *negative* links.

It is expected that an optimal embedding gives a higher similarity score to a positive link, rather than a negative link. This is due to the fact that the positive link already existed in the original network and that the optimal embedding was able to preserve the structural information of the network. The link prediction performance is then quantified by performing $n$ comparisons of positive- and negative links. The probability of a positive link being assigned a higher similarity score than a negative link, is a metric indicative of the overall link prediction performance of $E_{G_{train}}$. This metric is known as the Area Under the Curve (AUC) score [34]:

$$AUC = \frac{n' + 0.5n''}{n} \tag{2.8}$$

where $n$ denotes the number of comparisons of the similarity score between the positive- and negative links. $n'$ denotes the number of times when $S_{positive} > S_{negative}$ and $n''$ denotes the number of times when $S_{positive} = S_{negative}$.

## 2.3. Predicting Node Influence

### 2.3.1. Influential nodes

#### Definition of node influence

The influence of a node is commonly defined in the context of epidemic spreading processes unfolding on a network, as it generalizes to a variety of problems in different domains. Some examples are the spreading of diseases, failure cascade in electrical networks and opinion spreading in social networks *etc.*. Therefore, the influence of a node is best described using a dynamic process. Suppose a contagion process starts in some node $u$ in a network $G$. When this process terminates, $G$ will have either a fraction or all of its nodes being affected by the epidemic outbreak. This quantity, also called the prevalence or the final epidemic outbreak size, denotes the influence of node $u$ [33, 39, 17, 41].

#### Advantages

Most studies on the node influence have been conducted on the *identification task* rather than its *quantification*. Surveys such as [33], [41] and [20] list several methods and advantages of identifying the top influential nodes, but lack information on the merits of knowing their exact influence. Recent work in [7] sheds some light on this aspect. As mentioned by Bucur *et al.*, being able to predict the effect of a seed node, given a dynamic process, one can select influential nodes according to some threshold instead of simply selecting the top fraction. Another advantage of having a predictive model for the node influence is that it can deepen our understanding on how nodal features facilitate a contagion process, which is often unknown. Lastly, on large scale networks a predictive model is computationally more efficient in contrast to the stochastic simulation of the epidemic process for each node (even assuming that the exact contagion process and its parameters are known).

### State of the art methods

Predicting the influence of a node using its network structure is more challenging compared to the ranking problem [7], due to the following reasons:

- The influence of a node does not only depend on the network topological features, but also on the effective transmission rate ($\tau = \frac{\beta}{\mu}$) of the contagion process.
- In the node influence prediction, the machine learning (ML) step is yet another variable to be optimized. Here, not only the feature engineering step, but also the availability of training data can negatively impact the generalization performance of the predictive models.

At the time of writing, only a limited number of articles are available that concretely addresses this task. To justify the next step in this research direction and thus the work conducted in this project, a brief description is given for each of those articles with their findings. To make it more convenient to compare each individual work, we will use the following terminology: Let $\vec{X}_i$ denote the feature vector and $y_i$ the predicted node influence, then $f(\vec{X}_i)$ denotes the proposed predictive model (see Figure 2.6).

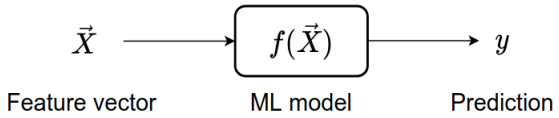



Figure 2.6: Generalized notation of a prediction model.

In 2019, Bucur *et al.* proposed a machine learning framework to predict the expected epidemic size $y_i$ using a set of classical centrality features [7]. Here, the influence of each node was determined using an exact Susceptible-Infected-Recovered (SIR) model with $\frac{1}{16} \leq \beta \leq 16$ (see Table 2.1 for details). The dataset consisted of all non-isomorphic, connected, simple undirected graphs with a size of $6 \leq N \leq 10$ nodes. Thus, instead of predicting the node influence in the context of a single topology, the aim was to build a model which generalizes over all possible graph topologies. The findings in this work conclude that not all centrality metrics are required to achieve a good prediction performance. Instead, combinations of metrics such as the degree with a spectral based centrality metric are sufficient as features (one local- and one global metric). In addition, the overall prediction performance slightly decreased with increasing $\beta$.

The previous analysis was extended to a wide variety of real world complex networks in [6] by the same author. The aim was to generalize the previous observations to multiple networks and to investigate *why* some combinations of topological metrics (see Table 2.1) were good predictors for the top influential nodes. It should be noted that the ML framework predicted whether a node was a top influential spreader or not, rather than prediction its influence. Based on a SIR contagion process where $\tau = \tau_c$ it could be observed that (i) the predictive power of individual metrics were inconsistent across multiple CNs, (ii) one local- and one global metric was needed to accurately identify the top influential nodes, and (iii) using all classical centrality metrics as features, near perfect prediction performance could be achieved.

While the previous two studies focused on the effect of centrality metrics on the prediction performance, others have proposed novel prediction frameworks. Zhao *et al.* have designed a prediction method where training and testing has been conducted on different networks. More precisely, let $G_1$ and $G_2$ denote two different networks, then the model $f(\vec{X})$ has been trained using all the nodal features from $G_1$. After training, $f$ was evaluated by predicting the influence of the nodes in $G_2$. A similar approach has been used in the work of Sebastian M. *et al.* and has been termed: *transfer learning* in [36]. While these approaches seem novel, some key observations are worth mentioning (see Table 2.2):

- Every work uses a SIR model with a different transmission rate $\tau$. While some authors properly justify the choice of this parameter, others do not.
- The size of the training data set is inconsistent across multiple studies and is commonly larger than half of the network. In some cases, almost the whole network is used during the training phase.

Table 2.1: Overview of the most relevant work conducted on the node influence prediction. For clarity, the metrics used in the predictive models are abbreviated as follows: Degree Centrality (**DC**), Eigenvector Centrality (**EC**), PageRank (**PR**), Closeness Centrality (**CC**), Betweenness Centrality (**BC**), Clustering Coefficient Centrality (**CCC**). $\vec{x}_M$ denotes the embedding vectors obtained from method $M$. Additionally, machine learning models are abbreviated as: Random Forest Regression (**RFR**), Support Vector Regression (**SVR**), Support Vector Machine (**SVM**), $k$ Nearest Neighbours ($k$**-NN**), Logistic Regression (**LR**), Multi-Layer Perceptron (**MLP**), Graph Convolutional Networks (**GCN**), Graph Attention Networks (**GAN**). **XGBoost** denotes an extended version of a gradient boosted decision tree.

| **Author(s)** | **Article** | **Year** | **ML Framework Details** | | |
|---|---|---|---|---|---|
| | | | **Features** $\vec{X}$ | **Label** $\vec{y}$ | **Model** $f(\vec{X})$ |
| Bucur D. et al. | [7] | 2019 | DC, EC, PR, CC, BC, KC, k-Core | Exact SIR | RFR, SVR |
| Bucur D. | [6] | 2020 | DC, EC, PR, CC, k-Core, Neighbourhood, 2 Hop Neighbourhood | Stochastic SIR | SVR |
| Zhao G. et al. | [65] | 2020 | DC, EC, PR, CC, BC, Load Centrality, k-Shell, k-Core, CCC | Stochastic SIR | Naive Bayes, Decision Trees, RFR, SVM, k-NN, LR, MLP |
| Rodrigues F.A. et al. | [46] | 2019 | DC, EC, PR, CC, BC, CCC, k-Core | Stochastic SIR | RFR, Neural networks |
| Sebastian M. et al. | [36] | 2021 | DC, EC, PR, Average Out-degree, number of 2nd neighbours, $\vec{x}_{Node2Vec}$, $\vec{x}_{SnoRe}$ | Compartmental SIR | GCN, GAN, XGBoost |
| Torricelli M. et al. | [57] | 2020 | $\vec{x}_{Node2Vec}$ | Compartmental SI | Linear Regression |

- In some frameworks, the predictive model is evaluated on networks different than the one used during training.

Table 2.2: Overview of the parameters used in the proposed ML frameworks.

| **Author(s)** | **Article** | **Year** | **Size Training Data** | $\tau$ |
|---|---|---|---|---|
| | | | **[% of full dataset]** | **[-]** |
| Bucur D. et al. | [7] | 2019 | 5 - 75* | 1/16 - 16 |
| Bucur D. | [6] | 2020 | 50 | $\tau_c$ |
| Zhao G. et al. | [65] | 2020 | 70 | 0.01 - 0.2 |
| Rodrigues F.A. et al. | [46] | 2019 | 50 | 0.3 |
| Sebastian M. et al. | [36] | 2021 | 100** | 0.1 |
| Torricelli M. et al. | [57] | 2020 | 90 | 1*** |

As the dataset consisted of all non-isomoprhic networks, the size of the training data was varied*.
Training and testing conducted on completely different networks**.
SI contagion model used where only $\beta$ is used to control the epidemic spreading***.

## 2.3.2. Research Gap

The discussion in the last section only partly addresses some of the issues related to the node influence prediction problem. In a more broader view, some of the main areas open to research are discussed next.

### From feature engineering to feature learning

Determining how classical centrality metrics affect epidemic spreading phenomena has been the main topic of interest. However, in a practical setting with large scale networks, computing the global based centrality metrics becomes intractable. A good candidate metric is the closeness centrality which enhances the prediction power while being expensive to be computed (with a running time complexity of **O(**$|V|^2$**)**). Therefore, even if certain classical metrics have been proven to be appropriate to the given prediction problem, the following question remains: *Can those metrics be computed efficiently?*

An alternative to this issue is to learn latent features instead and use those as input in the several ML algorithms. As described in Section 2.2, network embedding allows each node $v$ in a network to

be encoded as a latent feature vector $\vec{e_v}$. The advantages are twofold (i) embedding vectors can be readily used as input in ML algorithms. Earlier work in [36] already adopted this idea in which embedding vectors were utilized as nodal features, and (ii) the dimensionality of the embedding vectors can be tuned for more flexibility. Work conducted in [57] show that increasing the embedding dimension resulted into slightly better predictions when using a linear regression model.

**Predictive models practical to a real world scenario**
In a real world scenario where a contagion process unfolds on a network, limited data is available. Additionally, the full contact structure of the network is often incomplete. Therefore, a predictive model which utilizes minimal training data is more desirable. Most studies so far have proposed prediction frameworks learned on at least 50 % of the network structure. While useful for statistical inference, these frameworks have limited use in a practical setting, especially in the context of large scale networks. Thus, there is a need to research on how limited data can be utilized more efficiently during the ML training phase.

**Parameter tuning in embedding- and prediction models**
The node influence prediction task can broadly be divided into three main steps. In the first step, features are manually handcrafted or learned using some embedding method. In the second step, a contagion process is utilized to generate the final epidemic size for each node. Lastly, a machine learning framework is constructed for prediction. Each of those steps contain its own set of parameters that affect the final performance. Currently, there is a need to concretely assess how the embedding- and ML parameters affect the node influence prediction.

### 2.3.3. Task Description

The node influence prediction framework used in this thesis project is defined as follows and is visualized in Figure 2.7:

**Definition 3.** *Given a graph G(V, E), let S denote the set of nodes where for every $s \in S$, the expected prevalence $E_s[R]$ is known. Then, the aim is to predict $E_v[R]$ for every $v \in \{E - S\}$ using a model $f$ trained only on the data in $S$. In doing so, the following constraints are imposed:*

1. *$|S| = a|V|$, where $a \approx 0.1$.*
2. *When generating $E_i[R]$ for every node $i$ using an SIR model, the transmission rate $\tau$ is chosen such that $E_{max}[R] \approx 0.1$. The choice of $\tau$ is unique for every network $G$.*

The aim is to train a predictive model $f(\vec{X})$ using only the node influence and features of the nodes in the training data set $S$. Afterwards, $f(\vec{X})$ is used to predict the node influence of the unknown nodes in $E - S$. The novelty in this task definition is the additional two constraints imposed on the learning framework. First, the training dataset is kept deliberately small in order to better represent a real world scenario. Second, $E_i[R]$ is maintained at a maximum prevalence level of $\approx 0.1$. Instead of using the epidemic threshold $\tau_c$ to guide the choice of $\tau$, the prevalence level is fixed instead. To generate $E[R]$ for every node, a stochastic SIR model is used which is discussed in the next section. While the latter two constraints make the task description more rigid, at the same time network embedding based features generated using Node2Vec are utilized in the prediction models.

### 2.3.4. SIR Epidemic Model

The Susceptible-Infected-Recovered (SIR) epidemic model is a common choice to investigate the dynamics of many real world diffusion processes and is best explained in the context of disease spreading [39, 24]. In this model, each node in the network can occur in one of the three states at any given time step:

- **Susceptible state (S):** The initial state that most of the nodes start in. In this state, the node is vulnerable for an infection which can occur with some probability per unit time, $\beta$. This is also called the infection rate.

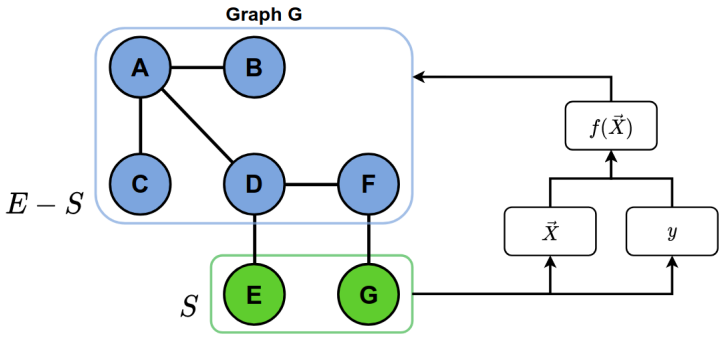



Figure 2.7: Schematic overview of the prediction model. Given a graph **G**, set **S** denotes the nodes whose influence is known and set **E-S** the set of nodes whose influence needs to be predicted.

- **Infected state (I):** Once a node has been infected with a disease, it resides in the infected state. In this state, two events may happen: it can infect other susceptible nodes with an infection rate $\beta$ or it can recover from the disease with a recovery rate $\mu$.
- **Recovered state (R):** A node can occur in this state only if it has been recovered from being infected. Once this state is reached, the node remains in this state (the equivalent of being immune to a disease after recovery).

An important quantity to characterize a network on its robustness towards an epidemic process is the epidemic threshold $\tau_c$. Let $\tau = \frac{\beta}{\mu}$, then an infected node can cause an epidemic outbreak of finite size if $\tau > \tau_c$. Otherwise, the diffusion process prematurely terminates without any severe consequences. However, in the stochastic SIR model even if $\tau > \tau_c$ it is very well possible that an outbreak does not occur [24]. This is demonstrated in Figure 2.8, where 1000 repeating simulations of an SIR process is conducted on a network with 849 nodes. While, $\tau > \tau_c$, a non zero fraction of simulations terminate without causing an outbreak. Assuming that the SIR process leads to an outbreak, it does not matter which seed node initiated it. However, it is the frequency of the occurrence of an outbreak being affected. As a result, the node influence is best quantified by the *expected* final epidemic size, which is simply the average of the final recovered fraction over many repeating simulations. In the remainder of this report, this quantity is denoted by $r_i$, where $r$ refers to the influence of node $i$, averaged over $1000$ iterations of the spreading process.

### 2.3.5. Machine Learning algorithms

In this section, a brief description is given of the basic principle of three commonly used machine learning algorithms: Ridge-, Support Vector- and Random Forest Regression. The choice for these particular methods stems from the literature survey conducted in Section 2.3 and by adapting the methodology proposed by Bucur D. *et al.* in [7] and [6]

#### Ridge Regression

The simplest learning model is the linear regression model. Let $X_i^{(d)}$ denote a $d$ dimensional feature vector with the elements $\{x^{(1)}, x^{(2)}, ..., x^{(d)}\}$ and $y_i$ a real-valued quantity to be predicted for an observation $i$. Then, the linear model is defined as:

$$f(X_i) = \theta_0 + \sum_{j=1}^{d} X_i^{(j)} \theta_j \tag{2.9}$$

where $\theta$ represents the unknown parameters, found by minimizing the residual sum of squares over a dataset with $n$ observations:

$$RSS(\theta) = \sum_{i=1}^{n} \left(y_i - f(\vec{X}_i)\right)^2 \tag{2.10}$$

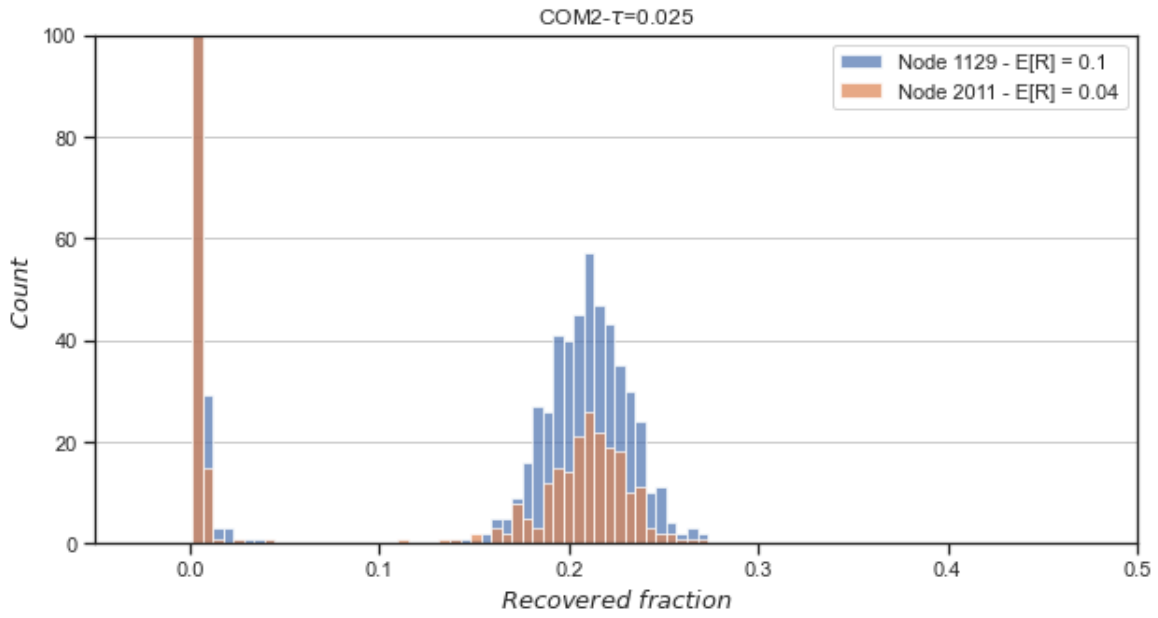



Figure 2.8: The distribution of the prevalence in a complex network with 849 nodes with a SIR epidemic model. The histogram is created on a basis of 1000 repeating simulations. The recovered fraction of nodes is measured after the infection terminates.

The main assumption of the linear model is that $y_i$ can be expressed by a linear dependency structure between its features in $X^{(d)}$. An issue with the standard definition of the linear model is overfitting. This occurs when $n \leq d$, resulting into a model with low bias and large variance. To improve the prediction performance, regularization can be applied. Here, a penalty is imposed in the loss function when $\theta$ becomes too large:

$$RSS_{penalized}(\theta) = \sum_{i=1}^{N} \left(y_i - f(\vec{X}_i)\right)^2 + \lambda \sum_{j=1}^{d+1} \theta_j^2 \tag{2.11}$$

where $\lambda$ is a model parameter controlling the regularization strength. The effect of the network structure on a dynamic process can be non-linear. While the linear regression model may seem simple, it can very well outperform non-linear regression models when the number of training samples is limited. Furthermore, the linear model can easily be extended to capture non-linear patterns by transforming the feature space using kernel methods [16].

### Support Vector Regression

Support Vector Regression (SVR) is a supervised learning method which performs well in a setting where less training data is available and the dimensionality of the feature vector is high. Let $n$ denote the number of training samples, then this regression method is still effective when $n \ll d$. In SVR, the goal is to find a function $f(X)$ such that for all training data $(x_i, y_i)$, $|y_i - f(x_i)|$ is at most $\epsilon$. While in the linear regression model, an unique and optimal $f(X)$ could be obtained (the least square solution), in this case any function that satisfies the constraint is a plausible. The function $f(X)$ can be found by minimizing the following loss function:

$$L = \frac{1}{2} \sum_{j=1}^{d+1} \theta_j^2 + C \frac{1}{N} \sum_{i=1}^{N} R_i^{\epsilon} \tag{2.12}$$

where $R_i^{\epsilon} = \max\{0, |y_i - f(x_i)| - \epsilon\}$, a quantity denoting deviations larger than $\epsilon$. $C$ is a trade off parameter between minimizing the magnitude of each coefficient in $\theta$ and error allowed on top of $\epsilon$ [49, 51]. An advantage of SVR is the application of non-linear kernels. Kernels can be used to map the original features to an alternative (higher dimensional) feature space allowing it to better capture non-linear patterns in the training data.

### Random Forest Regression

Random Forest based methods are build on the idea that a collection of learners will perform better compared to an individual one. In this case, each learner is a regression tree with low bias and high variance. In other words, each learner is able to capture complex structures in the data, while having poor generalization performance. It is this latter quality that is improved by aggregating the performance of a collection of trees (to reduce the variance). Briefly, the main steps in the RFR algorithm are described as follows [16]:

1. Given a training data set $X_n^{(d)}$ with size $n$ and dimensionality $d$, sample a subset and denote it as $X_{n'}^{(d')}$. Here, $n' < n$ and $d'$ denotes a subset of randomly picked features.
2. Build a regression tree $T(x)$ on the sampled dataset $X_{n'}^{(d')}$.
3. Repeat step 1 and step 2 $B$ times and collect all constructed trees in the set $\{T_b(x)\}_1^B$.
4. For prediction, aggregate the outcome of each individual $T_b(x)$ as follows:

$$y = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{2.13}$$

As described before, each individual tree will typically exhibit high variance as it it trained on a subset of the samples and features. When it is trained on a rather large subset, the size of the tree can grow large enough such that it overfits the training data set. This can be avoided by controlling the number of trees ($B$), tree depth ($D_T$), number of sampled data points ($n'$) and the number of sampled features ($d'$) [54].

# 3

# Methodology

This chapter covers the procedures necessary to reproduce the experiments performed in this work. It can be divided into three categories. First, all real world networks and its preprocessing steps are presented in **Section 3.1**, with an emphasis on characterization. Second, each step in the data analysis pipeline is presented in a separate section:

- The application of Node2Vec in order to produce the optimal network embedding is described in **Section 3.2**. Details on determining the link prediction performance is found in this section as well.
- **Section 3.3** contains the procedure and the network specific parameters used to simulate the SIR epidemic process.
- **Section 3.4** describes the framework used to learn the prediction models for the node influence. In this step, a minor study is conducted on the properties of the baseline- and network embedding based features and the challenges encountered to build the framework.

## 3.1. Network Data

The experiments proposed in this thesis project have been applied on a set of real world complex networks. As this project focuses on the properties of the embedding space and the creation of the node influence prediction framework, less attention is given to the domain related aspects of the networks. As a result, networks have been chosen from a variety of categories in order to investigate the generalization performance of the proposed method. A brief description for each network is as follows:

- **Protein protein interaction:** This is a network in which nodes represent proteins in a human cell. Proteins are macro molecules which participate or catalyze chemical reactions, often in conjunction with other proteins. The dataset consists of 2217 proteins whose interactions have been measured (see Figure 3.2a) [22, 15].
- **Facebook pages (food):** This is a social contact network in which the nodes represent facebook pages on food items. Whenever a user likes two different pages, this interaction is noted with a link between the corresponding nodes. It consists of a subset of all the interactions among 620 nodes. Figure 3.2b presents a visualization of the network. In contrast to the biological network, this network exhibits smaller well separated community structures [48].
- **Facebook messaging between users:** This network is comprised of university students messaging each other through Facebook. Each node represents an user and whenever two users communicate, a link is added between the nodes. This network consists of 1266 nodes (see Figure 3.2c) [38].
- **DNC email network:** This network represents a set of users which send emails to each other. Whenever such an event is observed a link is formed between the sender and the co-recipients of the email (see Figure 3.2d) [47].

Table 3.1: Overview of the static real world networks investigated in this thesis project. For each network, the number of nodes ($V$), the number of links ($E$), average hopcount ($E[H]$), link density ($\rho$) and average clustering coefficient ($ACC$) are shown.

| Network Type | Name | Notation | $\|N\|$ | $\|V\|$ | $E[H]$ | $E[D]$ | $\rho$ | $ACC$ |
|---|---|---|---|---|---|---|---|---|
| **Biological** | Protein protein interaction | BIO1 | 2217 | 6418 | 3.84 | 5.79 | 0.003 | 0.040 |
| **Social** | FB pages food | SOC1 | 620 | 2091 | 5.08 | 6.75 | 0.011 | 0.331 |
| **Social** | FB messages interaction | SOC2 | 1266 | 6451 | 3.31 | 10.19 | 0.008 | 0.068 |
| **Social** | Human contact | SOC3 | 410 | 2765 | 3.63 | 13.49 | 0.033 | 0.436 |
| **Communication** | DNC emails | COM1 | 849 | 10384 | 2.76 | 24.46 | 0.029 | 0.507 |
| **Communication** | Computer routers | COM2 | 2113 | 6632 | 4.61 | 6.28 | 0.003 | 0.246 |
| **Collaboration** | Citation (netscience) | CIT1 | 379 | 914 | 6.04 | 4.82 | 0.013 | 0.431 |

- **Computer routers:** This network contains 2113 nodes which represent a computer router. Whenever two routers communicate through packets, a link is formed in the network (see Figure 3.2e) [52].
- **Human contact network:** This temporal network contains 410 nodes, where each node represents a visitor during a science gallery event. Whenever two visitors came into close proximity, their interaction was recorded using a RFID badge in intervals of 20 seconds. In this work, the integrated static network is considered instead of the temporal network [23]. Figure 3.2f visualizes this network, where many community structures can be identified.
- **Author citation network (network science):** This network presents the co-authorship relationship between a subset of researchers in the field of network science. It consists of 379 nodes, where each node represents a researcher. Whenever two (or more) researchers collaborated on an article, a link has been identified between those researchers (see Figure 3.2g) [47].

Figure 3.1 presents the degree distribution for each of the previously listed networks. Based on the degree distribution the networks can be classified as follows:

- **Scale free networks:** In these networks, the degree distribution follows a power-law. On the log log scale, the dependency between the frequency of the nodes with its degree becomes approximately linear. In this case, only a handful of nodes have a very high degree, while the majority of the nodes have a low degree. These nodes, also termed hubs can play a vital role in the case of an epidemic spreading process. This phenomenon is a typical characteristic of sparse real world complex networks, which is why these networks have been chosen in this study. The networks that exhibit the scale free property are shown in Figures 3.1a-e: BIO1, SOC1, SOC2, COM1 and COM2.
- **Random networks:** In these networks, the degree of the nodes follow a poisson distribution. In this case, the majority of the nodes have a similar degree. In addition, very large hubs are not present in contrast with scale free networks. The networks that exhibit this property are shown in Figures 3.1f-g: SOC3 and CIT1.

Table 3.1 lists the additional properties of the previous networks such as the average hopcount ($E[H]$), average degree ($E[D]$), link density ($\rho$) and average clustering coefficient ($ACC$). The aim has been to select networks with varying properties in order to evaluate the robustness of the proposed prediction framework.

### 3.1.1. Preprocessing

Each complex network in Table 3.1 is shown after a pre-processing step. In this step, four aspects are evaluated:

- When the network dataset contains temporal information, the integrated static network is considered instead.
- When links are directed and (or) contains weights, these are converted to undirected- and unweighted links.

- Some complex networks may contain self loops. These are removed.
- Finally, the networks are evaluated on connectivity. In this case, the largest connected component is retained. This is a requirement for the network embedding step.

Note that the network properties are computed after pre-processing.

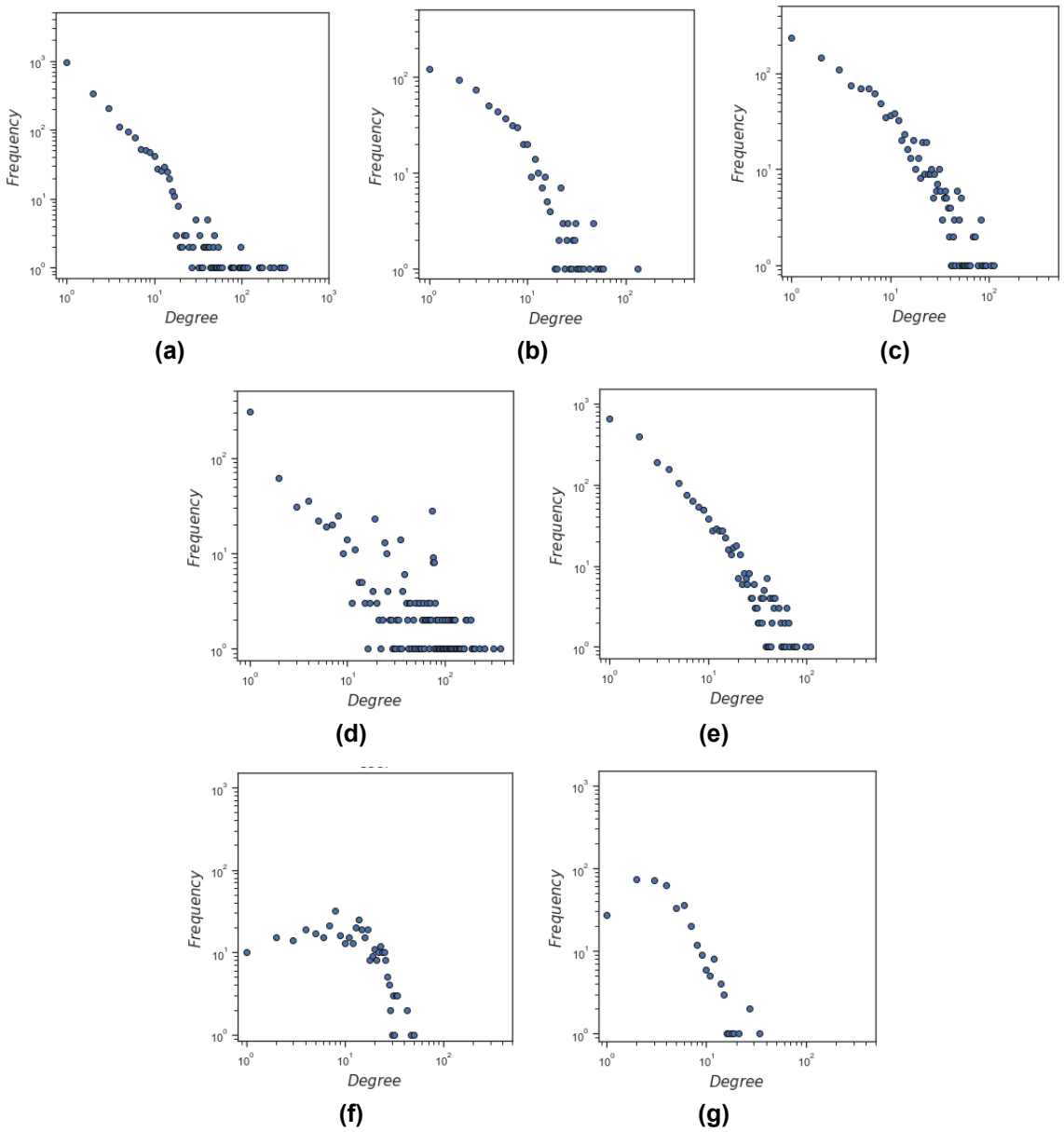



Figure 3.1: Degree distribution of the networks **(a)**: Protein Protein Interaction (BIO1), **(b)**: Facebook food pages (SOC1), **(c)**: Facebook messages (SOC2), **(d)**: DNC emails (COM1), **(e)**: Computer routers (COM2), **(f)**: Human contact (SOC3) and **(g)**: Citation network (CIT1).

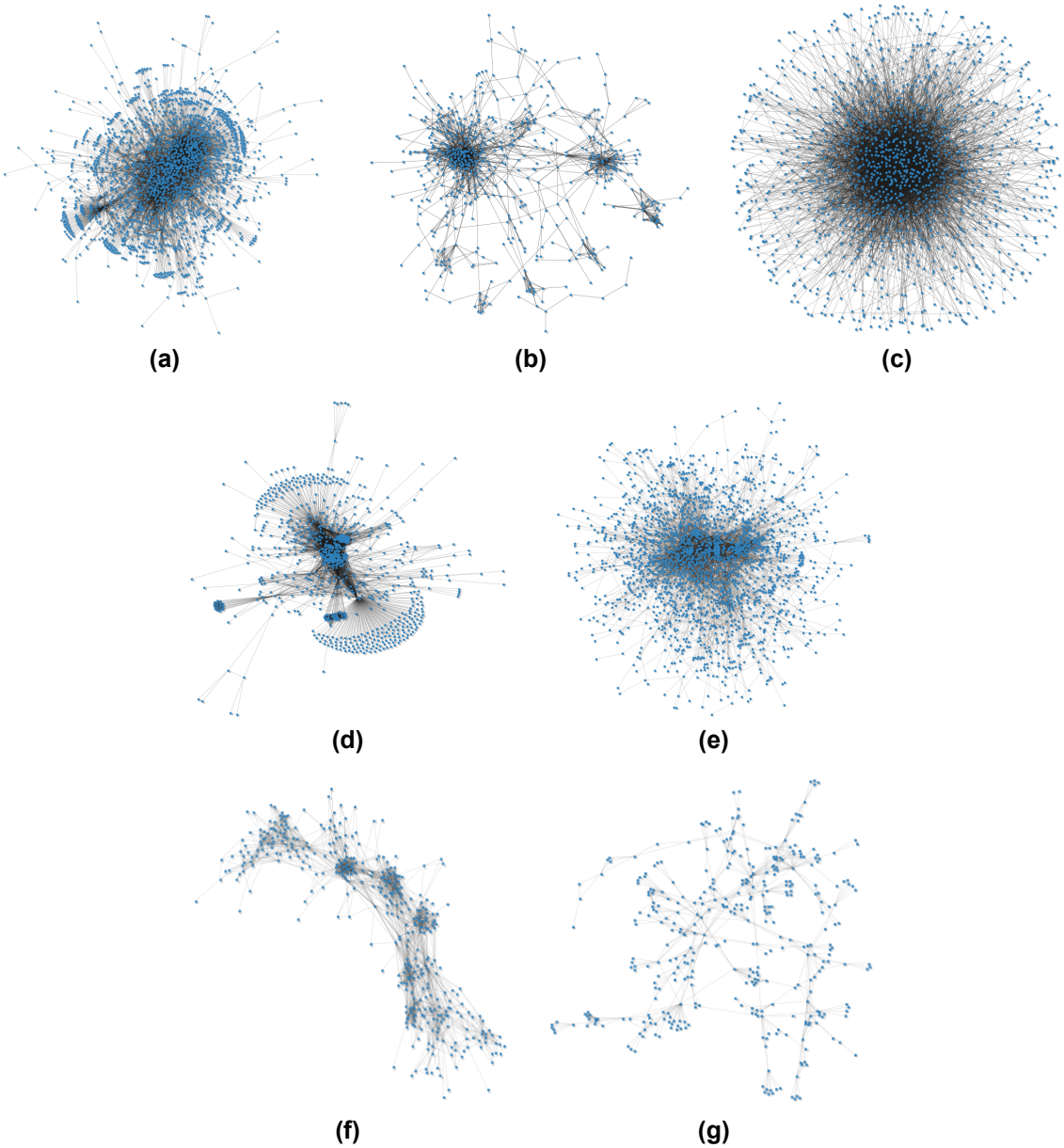



Figure 3.2: Visual depiction of the networks **(a)**: Protein Protein Interaction (BIO1), **(b)**: Facebook food pages (SOC1), **(c)**: Facebook messages (SOC2), **(d)**: DNC emails (COM1), **(e)**: Computer routers (COM2), **(f)**: Human contact (SOC3) and **(g)**: Citation network (CIT1).

## 3.2. Network Embedding Procedure

Figure 3.3 visualizes the pipeline used to produce the optimal embedding. The purpose of most of the steps in the pipeline is to determine the optimal set of hyperparameters $p$, $q$ and $d$ of the Node2Vec embedding framework. Given a network $G(V, E)$, the first step is to randomly remove $25\%$ of its links, while ensuring that the resulting network called $G_{train}$ remains connected. The removed links are denoted as *positive links*. Next, the same amount of *negative* links is sampled from the set $E^C$, which is the set of nonexistent links. Afterwards, Node2Vec is used to produce an embedding $E_{G_{train}}$ for $G_{train}$, which is then evaluated on its link prediction performance on the positive- and negative testing samples. To remove effects of the steps involving random sampling, the performance of the link prediction is averaged over $50$ repetitions (each network is split $5$ times and for each training network, $10$ embedding representations are produced). To determine the optimal parameters $p$, $q$ and $d$ in the Node2Vec algorithm, a grid search is performed over a range of values for each parameter: $p, q \in \{0.01, 0.25, 0.50, 1, 2, 4\}$ and $d \in \{32, 64, 128\}$. In the final step, the optimal parameters are used to embed the original network $G$.

The AUC score in the link prediction step is computed according to the method described in Section 2.2.4. First, each node pair (or link) in the positive- and negative test set is evaluated on the cosine similarity by using the obtained network embedding. Afterwards, a random positive- and negative link is sampled in order to evaluate whether the positive link had a higher score compared to the negative link. Repeating this comparison step for $n = 10000$ iterations, the AUC was computed using Equation 2.8. The AUC and the optimal parameters of Node2Vec for each network, is presented in Table 3.2. At the final step, these parameters are used to embed the full network.

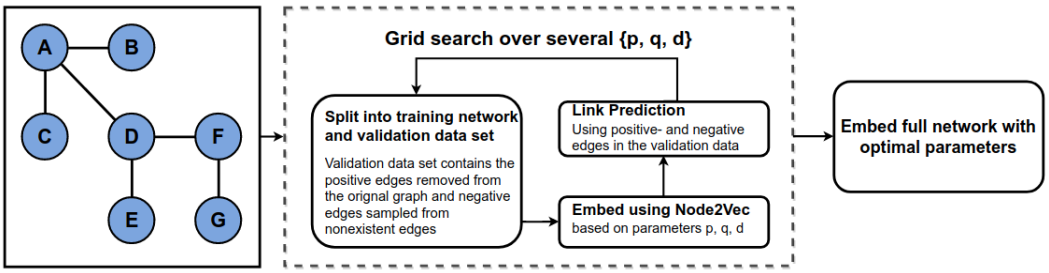



Figure 3.3: Pipeline denoting the steps taken in order to produce the optimal embedding for a network.

Table 3.2: Parameters in the Node2Vec algorithm used to produce the optimal embedding for each network.

| Network Type | Name | Notation | $p_{opt}$ | $q_{opt}$ | $d_{opt}$ | $AUC$ |
|---|---|---|---|---|---|---|
| **Biological** | Protein protein interaction | BIO1 | 1.0 | 0.01 | 32 | 0.71 |
| **Social** | FB pages food | SOC1 | 0.01 | 0.01 | 128 | 0.76 |
| **Social** | FB messages interaction | SOC2 | 2.0 | 0.01 | 32 | 0.78 |
| **Social** | Human contact | SOC3 | 0.50 | 0.25 | 32 | 0.94 |
| **Communication** | DNC emails | COM1 | 0.25 | 0.01 | 32 | 0.86 |
| **Communication** | Computer routers | COM2 | 1.0 | 0.01 | 32 | 0.93 |
| **Collaboration** | Citation (netscience) | CIT1 | 0.25 | 0.25 | 128 | 0.98 |

## 3.3. Node Influence

For each network, the influence $r_u$ of each node $u$ is computed using a stochastic SIR epidemic model. As explained in Section 2.3.4, the node influence is characterized by the expected value of the final epidemic size when the dynamic process terminates. Therefore, the node influence is averaged over $1000$ repetitions of a SIR process on a network for a given seed node. The main parameters used in the SIR model is the effective transmission rate $\tau = \frac{\beta}{\mu}$, where $\beta$ is the infection rate and $\mu$ the recovery rate. While $\mu$ is kept at a constant value of $1$, $\beta$ is chosen such that the maximum $r_u$ over all the nodes

approximates a value of $0.1$. Table 3.3 lists the values for these two parameters for every network.

Table 3.3: Effective transmission rate $\tau$ used in the SIR model for each network in order to ensure that $r_{u\ max} \approx 0.1$. $\tau_c$ denotes the numerical estimate of the epidemic threshold as described in Appendix A.

| Network | $\tau$ | $\tau_c$ |
|---|---|---|
| **BIO1** | 0.080 | 0.054 |
| **SOC1** | 0.110 | 0.080 |
| **SOC2** | 0.065 | 0.048 |
| **SOC3** | 0.090 | 0.070 |
| **COM1** | 0.020 | 0.015 |
| **COM2** | 0.095 | 0.060 |
| **CIT1** | 0.410 | 0.35 |

## 3.4. Prediction Framework

In this section the procedure used to obtain the node influence prediction model is described in multiple steps. As can be seen in Figure 2.7 in Section 2.3.3, the first step in this process is to choose a small subset of nodes as the training data. The choice of this subset with its prepossessing method is described in Section 3.4.1. Afterwards, the cross-validation strategy with its parameters are presented in Section 3.4.2 for each of the learning models. Finally, the evaluation metrics are discussed in Section 3.4.3.

### 3.4.1. Data Set

#### Splitting the data into Training- and Testing Set

In order to obtain a prediction model, two data types are needed: training- and testing data. In the case of this study, the training data refers to the ”small” subset $S$ with nodes from the network whose influence is known. The testing data refers to the set $V - S$, where the nodal influence information is not used in model training, but only in model testing. The default size of the training data is $10\%$ of the nodes in the network, while the testing data consists of the remaining $90\%$. In some experiments these may be different and will be specified accordingly. One important aspect is the method used to split the original network into the two datasets. As can be seen in Figure 3.4, the distribution for the node influence over all the nodes in the Protein protein interaction network, shows that most of the nodes have a very low influence. In contrast only few nodes have a large node influence. These are also the nodes of interest. Determining the set $S$ randomly is desired as in practice we cannot control how the data presents itself. In addition, the training data and the testing data should both have the same distribution. However, since the size of the network is relatively small, it can be the case that zero highly influential nodes will be sampled randomly. This is less of an issue when the networks are larger. Therefore, a stratified random sampling strategy is used to maintain the same distribution of the node influence in the training- and testing data. In this strategy, the nodes are divided into $b = 5$ bins, and from each bin the same proportion is sampled into the training data, depending on the size of the training data needed. This ensures that at least one sample node with a high influence will be present in the training data set. As mentioned before, in a practical setting, the distribution of the given training dataset cannot be controlled beforehand. Thus, in this work we explore two different scenarios: (1) splitting the data randomly and (2) using a stratified sampling strategy.

#### Data imbalance in Regression

Figure 3.4 presents an additional issue which should be accounted for during model training: data imbalance. It should be noted that this work is not heavily focused on the machine learning aspects. However, mentioning these topics are important as it will help to better position the observations. In the problem of data imbalance, the nodes for which the prediction framework should work the best during prediction, occurs in orders of magnitude lower quantity than the nodes for which the prediction is irrelevant. In machine learning a distinction is then made between the majority class and the minority class [45]. There are several methods that can alleviate the drawbacks of such an imbalanced dataset, some of which are:

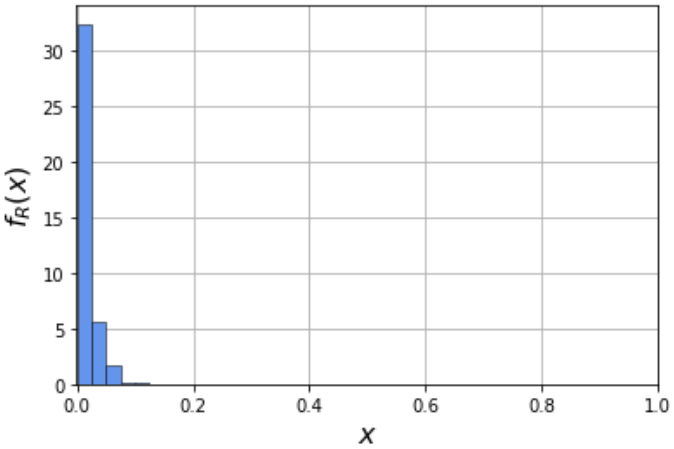



Figure 3.4: Probability density function $f_R(x)$ of the average prevalence $R$ of a random node. The histogram consists of 40 bins split in the interval [0, 1] with the same bin size. The probability density function $f_R(x)$ at a given bin $x$ then equals the fraction of data within each bin normalized by the bin size (1/40).

- **Data Sampling:** This is a preprocessing technique where the imbalance in the data is evened out by either(i) oversampling the minority data, and (ii) undersampling the majority data. In the first case, the less frequent nodes are either duplicated or synthetically generated by introducing some noise in the features and the target variable [26]. In the second case, the more frequent nodes which are not of interest are removed from the dataset. Methods also exits where both these approaches are combined in order to produce a well distributed data set such as SMOTER and SMOGN [5, 56].
- **Sample Weights:** This is a method which is applied during the model training step. Here, the minority nodes are weighted higher in comparison with the majority nodes. As a result, during model training or optimization, prediction errors on the minority data are amplified in the loss function, guiding the model to correctly learn the minority data.

During the initial model development phase a sample weighting scheme based approach as proposed by Torgo *et al.* in [55] had been utilized in order to address the class imbalance issue. However, as no significant improvement was observed in the node influence prediction performance, the results were omitted in this report.

#### Comparison with Baseline Features

One objective of this work is to not only investigate the utility of embedding features, but also to benchmark those against classical topology based metrics. As of recent, several studies have already been conducted on which individual or combinations of centrality metrics are needed to predict the node influence [7, 6]. A common conclusion among these two studies is that a classical network topology based feature vector should contain at least a local- and a global metric. As a result, three centrality metrics have been chosen: the Degree- ($d_u$), Closeness- ($c_u$) and Eigenvector Centrality ($x_u$). Here, the $d_u$ denotes a local metric, the $c_u$ a global metric and the $x_u$ a spectral (global) metric.

### 3.4.2. Model Training

Each of the models described in Section 2.3.5 contains several tuning parameters. In order to tune these parameters and to reduce the effects of over-fitting, a grid search is performed over every possible parameter configuration. At each step of the grid search 5 fold cross-validation is applied to determine the performance of the prediction model. Finally, the model is retrained on the parameter configuration which produced the best results. It should be noted that in this case, the cross-validation procedure also utilized the previously defined sample weights. All these procedures with data preprocessing and model training have been conducted using the scikit learn library in Python [40]. The model specific parameters are as follows:

- **Random Forest Regression:** #Decision trees (estimators) trained: [10, 20, 50, 80, 100]. The

loss is the mean absolute error.

- **Support Vector Regression:** Regularization parameter $\lambda \in$ [0.0001, 0.0005, 0.001, 0.01, 0.1, 1.0, 10, 20]. The kernels used: [polynomial with degree 2, radial basis function with auto scaling].
- **Ridge Regression (Linear model):** Regularization parameter $\lambda \in$ [0.0001, 0.0005, 0.001, 0.01, 0.1, 1.0, 10, 20].

### 3.4.3. Model Evaluation

Machine learning frameworks, in the context of regression are evaluated by computing the coefficient of determination ($r^2$):

$$r^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{3.1}$$

where $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$, $\bar{y}_i$ the predicted value and $y_i$ the ground truth for observation $i$. The intuition behind this performance metric is that it is a comparison between two models in terms of the residual error: (i) the learned regression model $f(x)$ on the training data, and (ii) a model that always outputs the expected value of $y$. Thus, the better the regression model, the less variance it will have where $r^2$ will be closer to $1$. While this may denote a good prediction performance, it is difficult to judge the prediction performance of a model where $r^2$ is lower. This is generally application specific. In addition, especially for imbalanced datasets, this metric may give an incorrect estimate on the prediction performance. While the frequently occurring samples may be correctly predicted and the infrequent samples incorrectly, the $r^2$ metric can still take a value close to 1. Therefore, other metrics that specifically measures the prediction performance on the infrequent samples are needed. In classification problems, the most suitable metrics are the precision, recall and the F1 score to assess the classification performance of the minority class in comparison with the majority class. For network analysis, a similar set of metrics exist that can quantify how well a model is capable of correctly predicting the top $f$ fraction of influential nodes (minority data):

- **Recognition rate $r(f)$:** The recognition rate makes use of two rankings: the top $fN$ nodes in the network according to the true nodal influence and the top $fN$ nodes as identified using the predicted nodal influence. The number of common nodes in both rankings denote the recognition rate:

$$r(f) = \frac{|R_f^T \cap R_f^P|}{|R_f^T|} \tag{3.2}$$

  , where $R_f^T$ represents the nodes in the top $f$ fraction as ranked by the true node influence and $R_f^P$ the nodes in the top $f$ fraction as ranked by the predicted node influence.

- **Precision function $p(f)$:** . The previous metric is useful for determining whether the model is capable of identifying the nodes with the highest influence, with respect to the true values. To get a measure how well the prediction itself is on each of those nodes the precision function $p(f)$ as defined in [6] can be used:

$$p(f) = \frac{I_{i \in R_f^P} r_i}{I_{i \in R_f^T} r_i} \tag{3.3}$$

  , where $r_i$ denotes the influence of node $i$ and $I(.)$, the mean. This metric quantifies the average prediction on the node influence in comparison with the true average node influence at the top $f$ fraction of nodes in the network. Therefore, in conjunction with the recognition rate, these two metrics can give a clear view on the prediction performance of a model on the top influential nodes.

The previous two metrics give insight on the prediction of minority samples only when used in tandem. In case the recognition rate is low, while the precision is higher, this may still denote sub-par

performance. Therefore, these metrics are commonly aggregated into a single metric called the F1 score:

$$F1(f) = \frac{2}{r(f)^{-1} + p(f)^{-1}} \tag{3.4}$$

# 4

# Results

This chapter contains the experimental results that are used to answer the research questions defined in Chapter 1. The experiments are divided into two groups (i) exploration of the embedding feature space, and (ii) evaluation of the node influence prediction framework. The aim of the first set of experiments is to study whether the network topology information is captured into the optimal network embedding (see Section 4.1). The second set of experiments is constructed to evaluate the performance of the prediction framework (see Section 4.2). Additionally, the effect of the network topology on the node influence prediction performance is also investigated in order to justify some of the observed results (Section 4.3).

## 4.1. Exploring the Nodal Embedding Vector

The analysis on the relation between the distance (shortest path- and resistance distance) of two nodes in topology and the proximity of the two nodes between their corresponding embedding vectors is divided into two sections. In the first subsection, a study is performed to investigate whether a correlation indeed exists between the distance in the network topology and the proximity between the corresponding embedding vectors for each node pair. In this case, the network embedding is obtained as the one which best preserves the network topology according to the link prediction task. Afterwards, the effects of the embedding parameters are investigated on the observed correlation patterns.

### 4.1.1. Correlation Study on Distance Metrics

#### Description

In the first experiment, an optimal embedding is created for each network. Afterwards, the shortest path distance (SPD) and the resistance distance (RD) have been calculated for each node pair in the network topology. In addition, the cosine similarity (CS) has also been computed for each node pair using the corresponding (normalized) embedding vectors of each node. A correlation study has been performed between each topological distance and the proximity in embedding space. The goal is to then identify how the distance metrics in the topology space correlate with the CS. Note, the terms ”similarity” and ”distance” are interchanged as they convey the same concept in the embedding space (a smaller angle between two embedding vectors translates into a closer proximity). There are two reasons why these topology based distance metrics are interesting:

- **Embedding of network topology:** As described in Section 2.2.2, an optimal embedding should at least be able to reconstruct the original network topology for it to be effective in network inference tasks. Therefore, in the embedding it is expected that node pairs with a shorter SPD in the topology, have a higher CS (due to the closer proximity between the embedding vectors of the node pairs). As a result, the correlation pattern could affirm the possibility of the embedding to preserve the network topology.
- **Embedding of shortest path distance:** A wide variety of classical topology based metrics utilizes the SPD between all possible node pairs (for example the closeness $c_u$ of a node $u$).

Table 4.1: Pearson correlation coefficient between the Cosine Similarity (CS), Shortest Path Distance (SPD) and Resistance Distance (RD) metrics of all possible node pairs in the network.

| Network | Pearson Correlation | | |
|---|---|---|---|
| | $p_{(SPD,\ CS)}$ | $p_{(SPD,\ RD)}$ | $p_{(RD,\ CS)}$ |
| **BIO1** | 0.40 | 0.67 | 0.16 |
| **SOC1** | 0.34 | 0.74 | 0.03 |
| **SOC2** | 0.43 | 0.71 | 0.19 |
| **SOC3** | 0.21 | 0.46 | 0.08 |
| **COM1** | 0.25 | 0.44 | 0.12 |
| **COM2** | 0.34 | 0.75 | 0.12 |
| **CIT1** | 0.58 | 0.78 | 0.36 |

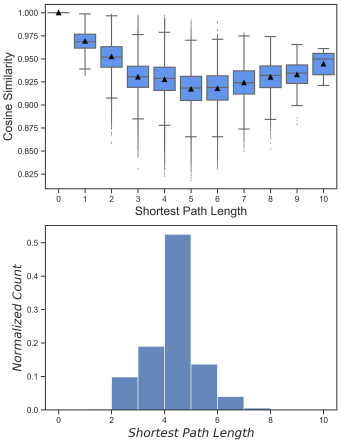
Investigating the correlation pattern may give insight whether the network embedding can be used to represent only the SPD of the node pairs in the original network.

### Observations

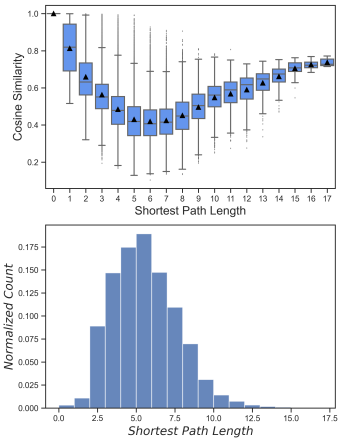
Table 4.1 presents the Pearson correlation coefficient $p(X, Y)$ between the metrics $X$ and $Y$, where $X, Y \in$ {Shortest Path Distance (SPD), Cosine Similarity (CS), Resistance Distance (RD)}. This test outputs a value in the range $\{-1, 1\}$ which denotes the degree of linearity between $X$ and $Y$. Three observations can be made. First, the SPD and CS are weakly correlated over most of the networks ($p(SPD,\ CS) \leq 0.4$). Second, the SPD and RD are moderately correlated in all networks. Lastly, the correlation between the CS and RD is negligible for all networks except for CIT1. Table B.1 in Appendix B also shows the Spearman correlation coefficient between the metrics in order to test whether a nonlinear relation exists or not. However, the same observations made on the basis of the Pearson correlation coefficient also hold for the correlation analysis using the Spearman correlation coefficient. While at first glance, these observations conclude that the network embedding hardly captures the distance in the network topology, more insight is obtained by consulting the (averaged) scatter plot between the various distance metrics. Figure 4.1 presents the SPD versus the CS of a node pair, where node pairs are grouped according to their SPD on the x-axis. It can be observed, if the SPD between a node pair increases in the network topology, then its similarity in the embedding space decreases at first, after which it starts to increase again. On the bottom row of each figure, the hopcount distribution is presented for all possible node pairs. It shows that the majority of the node pairs in the network have a relatively small hopcount. For such node pairs, the CS does decrease with increasing SPD. Therefore, the CS and SPD of a node pair tend to be negatively correlated. Figure 4.2 presents the relation between the SPD and RD of a node pair. As can be seen, the RD increases monotonically with the SPD in the network topology. The SPD between two node pairs is a distance measure which only considers a single (shortest) path between the two nodes. On the other hand, the RD also takes all other possible paths into account between the two nodes. It is based on the premise that the two nodes are effectively closer in distance if they are reachable via multiple paths. The previous observation shows that node pairs at increasing SPD are effectively harder to reach, despite the contribution of the multiple paths between these nodes.

### Analysis

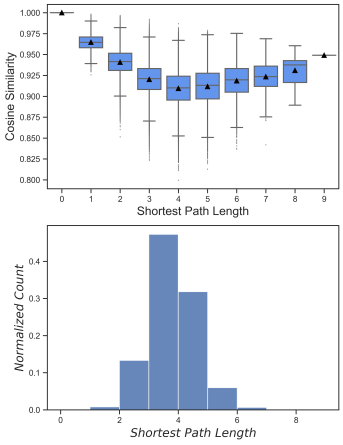
To make an attempt at the analysis of the previous observations, it is best to consider what the CS entails. As discussed in Section 2.2.3, the CS encodes the co-occurrence frequency of a node $v$ in the neighbourhood of a source node $u$, when performing random walks. The results show that nodes in the immediate neighbourhood are deemed less similar by the embedding, at increasing SPD. This indicates homophily, where nodes in the closer vicinity (or community) are highly connected and therefore similar [19]. On the other hand, node pairs where the SPD is much larger can also be embedded in closer proximity in the embedding space. At first, this suggests that Node2Vec is identifying similar nodes at a larger SPD. According to the work in [19], this result hints at node pairs at larger SPD to have a similar structural neighbourhood. However, based on the current results it is not clear whether this is indeed the case. Overall, the findings of this correlation analysis show that the distance between the majority of the node pairs in the topology is captured by the proximity in the embedding space.



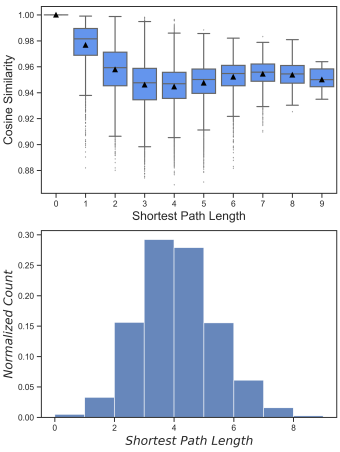

(a) Protein Protein Interaction (BIO1).
$E[D] = 5.79$
$E[H] = 3.84$



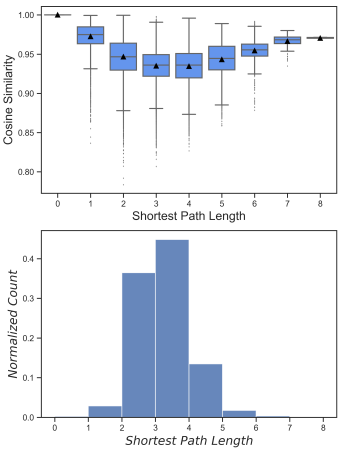

(b) Facebook pages food (SOC1).
$E[D] = 6.75$
$E[H] = 5.08$



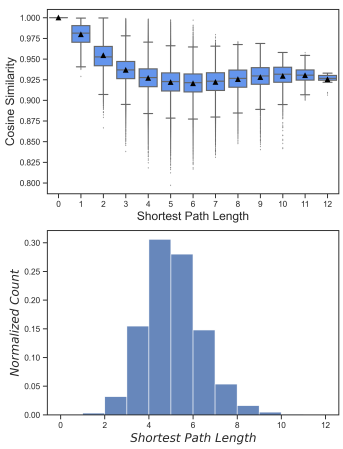

(c) Facebook messages interaction (SOC2).
$E[D] = 10.19$
$E[H] = 3.31$



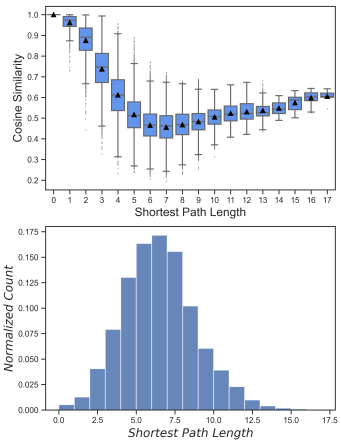

(d) Human contact (SOC3).
$E[D] = 13.49$
$E[H] = 3.63$




(e) DNC emails (COM1).
$E[D] = 24.46$
$E[H] = 2.76$

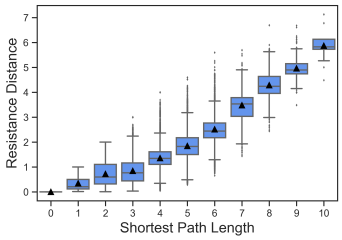



(f) Computer routers (COM2).
$E[D] = 6.28$
$E[H] = 4.61$

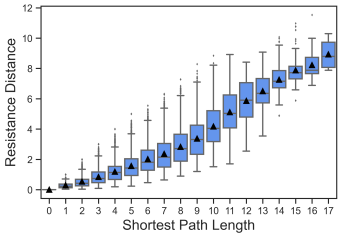



(g) Citation netscience (CIT1).
$E[D] = 4.82$
$E[H] = 6.04$

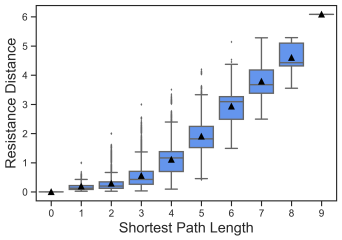
Figure 4.1: Cosine similarity versus the shortest path distance for all possible node pairs in the networks (Top row). The bottom row in each graph denotes the corresponding hopcount distribution of all possible node pairs.



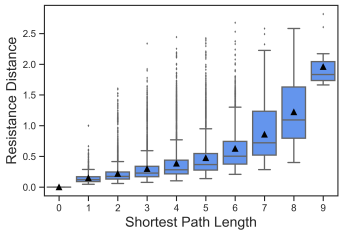

(a) Protein Protein Interaction (BIO1).



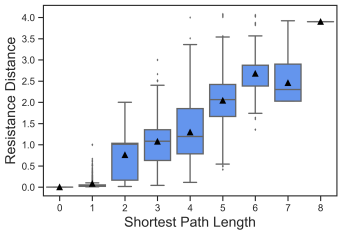

(b) Facebook pages food (SOC1).



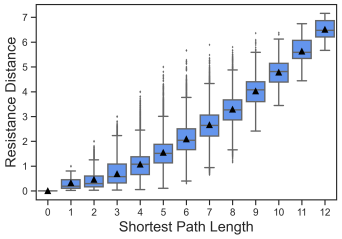

(c) Facebook messages interaction (SOC2).



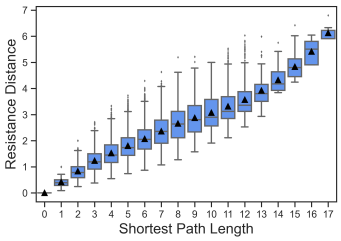

(d) Human contact (SOC3).




(e) DNC emails (COM1).




(f) Computer routers (COM2).




(g) Citation netscience (CIT1).

Figure 4.2: Shortest path distance versus the resistance distance for all possible node pairs in the networks.

### 4.1.2. Effect of Embedding Parameters

#### Description

The previous experiment has shown that the SPD of the majority of the node pairs in the network topology is preserved in the optimal embedding. However, it was also observed that for a minority of the node pairs in the network (where the SPD is larger than the average hopcount), the CS was relatively higher. In order to obtain the optimal embedding of the network using Node2Vec, the three tuning parameters $\{p, q \text{ and } d\}$ are optimized according to the link prediction task. This part of the experiment aims to investigate how these parameters affect the previously observed negative correlation between the CS and SPD of the node pairs. This is interesting due to the following two reasons:

- A parameter configuration could exist, which leads to an embedding in which the SPD and CS of the node pairs with a larger SPD, are negatively correlated as well. That is, the CS and the SPD exhibiting a monotonic correlation.
- For the node influence prediction task, the predictive models are trained where the optimal network embedding vectors for each node are used as features. The performance of these models could be improved by utilizing an embedding with a lower dimension, while ensuring that it still has preserved the observed negative correlation between the CS and SPD of the node pairs.

Therefore, in this experiment an embedding is produced on the given network for different values of $\{p, q, d\}$. Whenever, the effect of an individual tuning parameter is investigated, the other two parameters are set to the optimal values as listed in Table 3.2.

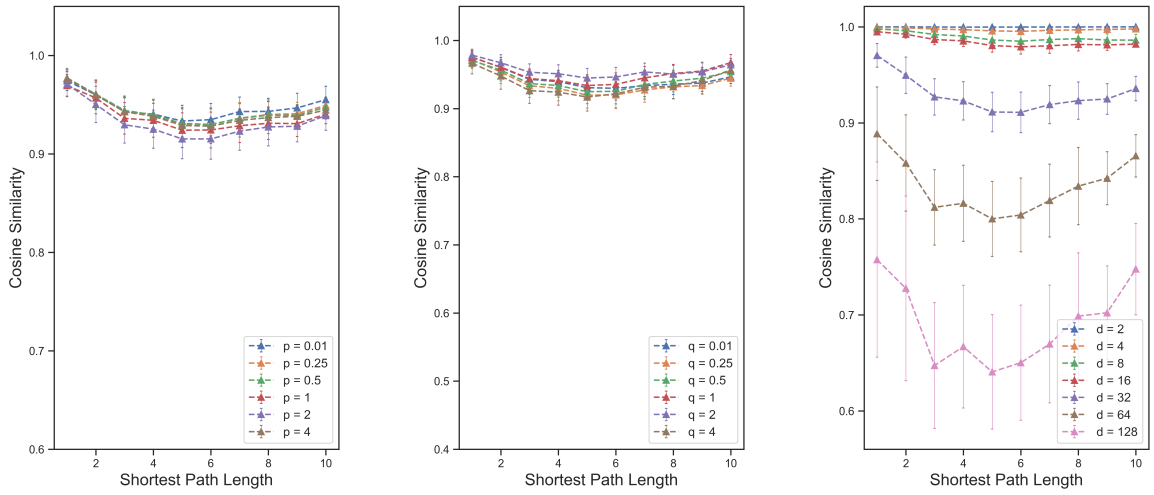
#### Observations

Figures 4.3-4.9 illustrate the CS versus the SPD for the node pairs in each of the networks in Table 3.1. Additionally, each sub figure in each graph depicts the individual effect of either $p$, $q$ or $d$ on the obtained correlation between the CS and the SPD of the node pairs. The following observations can be made:

- For the networks where $d = 32$ (BIO1, SOC2, COM1, COM2 and SOC3), individually varying the tuning parameters $p$ or $q$, hardly affects the previously observed negative correlation between the CS and SPD (as shown in Section 4.1.1).
- For the remaining two networks where $d = 128$ (SOC1 and CIT1), the majority of the node pairs still tend to have a negative correlation between their CS and SPD. However, the effect of the embedding parameters $p$ and $q$ are more prevalent.
- The embedding dimension $d$ has by far the largest effect on the correlation between the CS and the SPD of the node pairs. In general, when $d$ increases, nodes which are far apart in the network topology are deemed more similar in the embedding space. On the other hand, when $d$ decreases, so does the correlation between the CS and the SPD. When $d < 8$, most of the node pairs tend to have the same CS, denoting that the embedding is not able to distinguish the nodes in the network in terms of similarity.

#### Analysis

The previous set of observations show that individual tuning of the return parameter ($p$) and the in-out parameter ($q$) or both together, could result into a similar correlation between the CS and the SPD of the nodes, when the embedding dimension is not too large. When the embedding dimension is increased however, the negative correlation between the CS and the SPD of the node pairs becomes stronger. This could indicate that an embedding with a larger dimension is able to capture the distance in the network topology to a larger extent. While a network embedding with a lower dimension is favoured by supervised learning models, the results of this experiment show that an optimum value might exist, as an embedding with an extremely low dimension could not properly capture the SPD of the node pairs in the network topology.

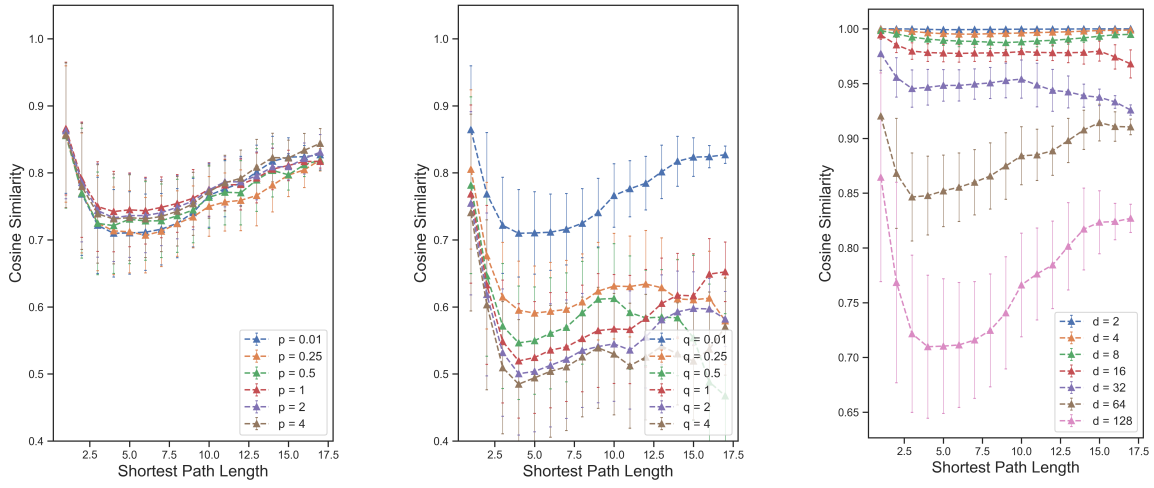



(a) Varying $p$ $q = 0.01$ and $d = 32$

(b) Varying $q$. $p = 1$ and $d = 32$

(c) Varying $d$. $p = 1$ and $q = 0.01$

Figure 4.3: Cosine similarity (CS) versus the shortest path distance (SPD) for all possible node pairs in the protein protein interaction network (BIO1). Whenever one of the parameters are varied, the other two are kept constant at the optimal values as listed in Table 3.2.

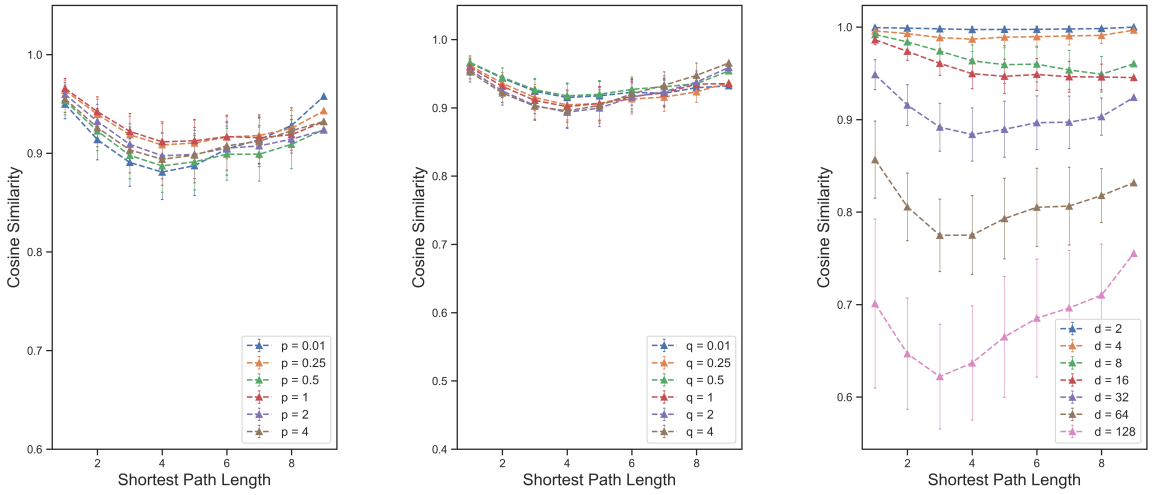



(a) Varying $p$ $q = 0.01$ and $d = 128$

(b) Varying $q$. $p = 0.01$ and $d = 128$

(c) Varying $d$. $p = 0.01$ and $q = 0.01$

Figure 4.4: Cosine similarity (CS) versus the shortest path distance (SPD) for all possible node pairs in the Facebook food pages network (SOC1). Whenever one of the parameters are varied, the other two are kept constant at the optimal values as listed in Table 3.2.

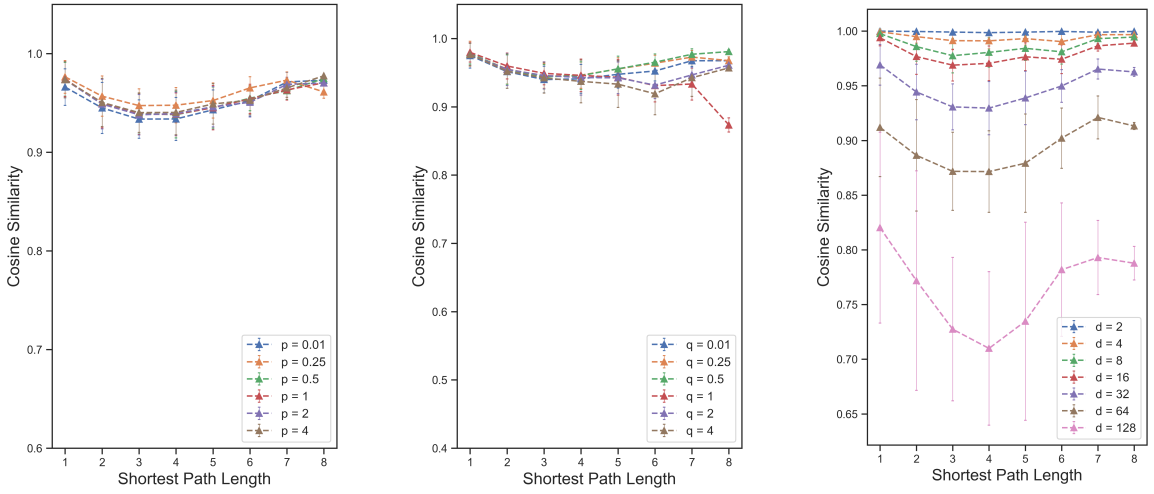



(a) Varying $p$
$q = 0.01$ and $d = 32$

(b) Varying $q$.
$p = 2$ and $d = 32$

(c) Varying $d$.
$p = 2$ and $q = 0.01$

Figure 4.5: Cosine similarity (CS) versus the shortest path distance (SPD) for all possible node pairs in the Facebook messages interaction network (SOC2). Whenever one of the parameters are varied, the other two are kept constant at the optimal values as listed in Table 3.2.

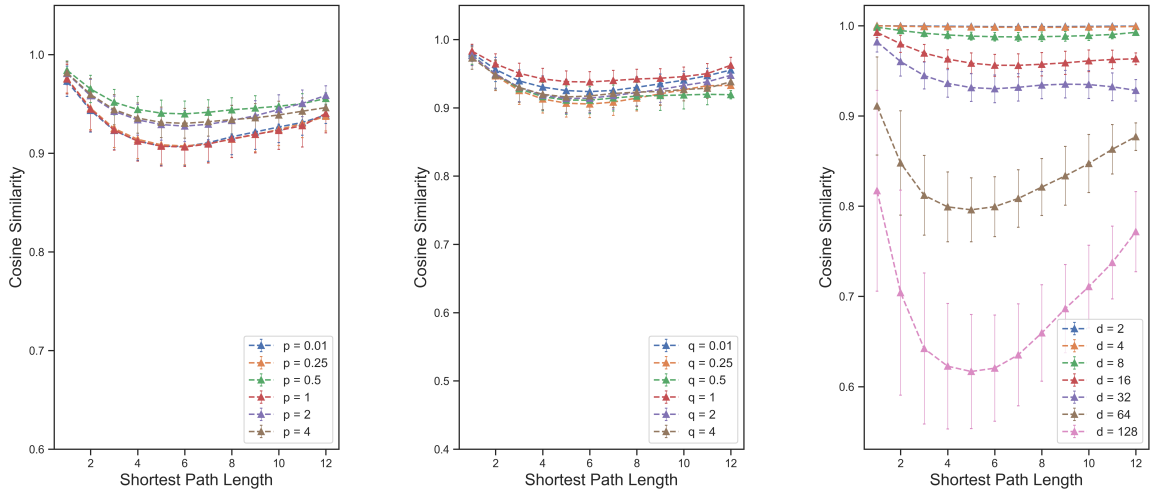



(a) Varying $p$
$q = 0.01$ and $d = 32$

(b) Varying $q$.
$p = 0.25$ and $d = 32$

(c) Varying $d$.
$p = 0.25$ and $q = 0.01$

Figure 4.6: Cosine similarity (CS) versus the shortest path distance (SPD) for all possible node pairs in the DNC emails network (COM1). Whenever one of the parameters are varied, the other two are kept constant at the optimal values as listed in Table 3.2.

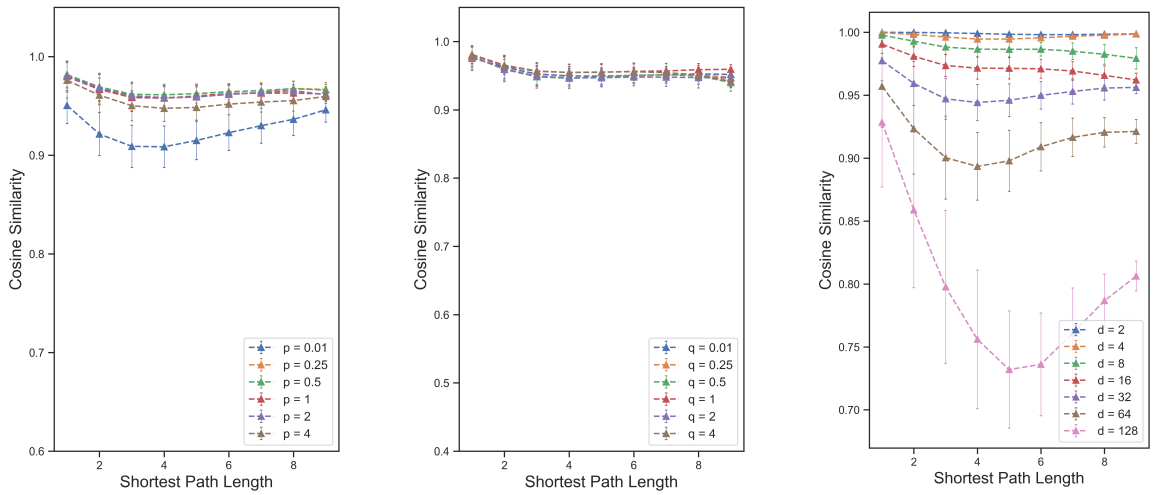



(a) Varying $p$
$q = 0.01$ and $d = 32$

(b) Varying $q$.
$p = 1$ and $d = 32$

(c) Varying $d$.
$p = 1$ and $q = 0.01$

Figure 4.7: Cosine similarity (CS) versus the shortest path distance (SPD) for all possible node pairs in the computer routers network (COM2). Whenever one of the parameters are varied, the other two are kept constant at the optimal values as listed in Table 3.2.

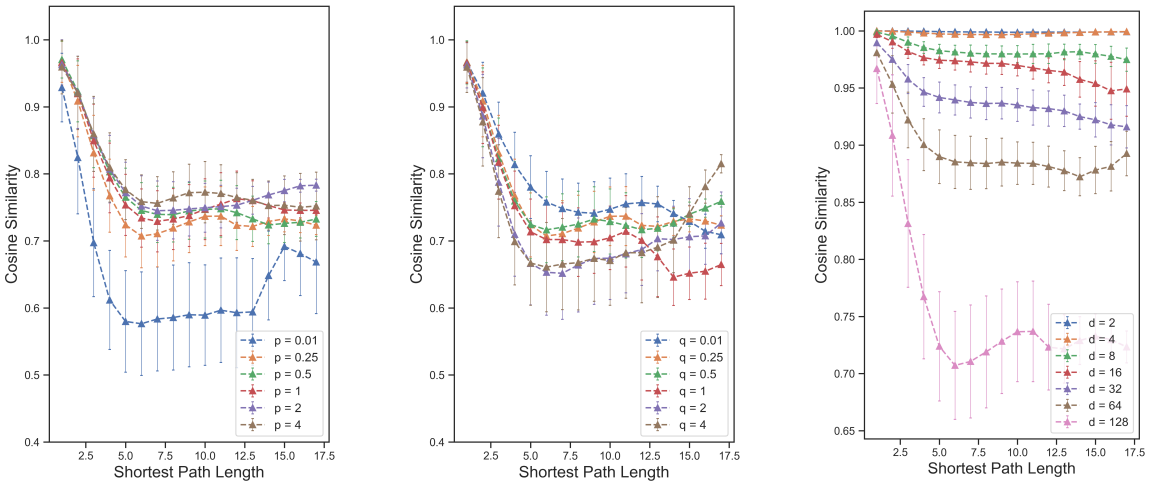



(a) Varying $p$.
$q = 0.25$ and $d = 32$.

(b) Varying $q$.
$p = 0.5$ and $d = 32$.

(c) Varying $d$.
$p = 0.5$ and $q = 0.25$.

Figure 4.8: Cosine similarity (CS) versus the shortest path distance (SPD) for all possible node pairs in the human contact network (SOC3). Whenever one of the parameters are varied, the other two are kept constant at the optimal values as listed in Table 3.2.




(a) Varying $p$.
$q = 0.25$ and $d = 128$.

(b) Varying $q$.
$p = 0.25$ and $d = 128$.

(c) Varying $d$.
$p = 0.25$ and $q = 0.25$.

Figure 4.9: Cosine similarity (CS) versus the shortest path distance (SPD) for all possible node pairs in the citation (netscience) network (CIT1). Whenever one of the parameters are varied, the other two are kept constant at the optimal values as listed in Table 3.2.

## 4.2. Framework Evaluation: Node Influence Prediction

In this section, the aim is to evaluate the performance of the Linear-, SVR and RFR regression models on the node influence prediction task. In order to achieve this, two different features are utilized: (1) classical network topology features and (2) network embedding based features. The analysis on this comparison is presented in Section 4.2.1). In the next step, the previous experiments have been repeated with two different splitting strategies in order to determine how the regression models would be affected by the manner in which the data is available in practice (see Section 4.2.2). In a subsequent experiment, the effect of reducing the embedding dimension is explored to determine whether this can improve the prediction performance (Section 4.2.3). In Section 4.2.4, the size of the training data set is increased even further to evaluate how much the network embedding based regression models can improve. Lastly, the network embedding- and classical topology based features are combined in order to determine scenarios where network embedding based features could prove to be the better solution (Section 4.2.5).

### 4.2.1. Comparison of Embedding and Topology features

#### Description

In this experiment, each model (Linear, SVR and RFR) is trained on a subset consisting of $10\%$ of the nodes in a network, using either topology features ($\vec{X}_{Top}$) or embedding features ($\vec{X}_{Emb}$). In the latter case, the features are based on the optimal network embedding as determined by the link prediction task. As mentioned in the previous chapter, the topology based features consists of the degree- ($d_u$), eigenvector- ($x_u$) and closeness centrality ($c_u$) metrics. The model training is repeated 50 times and the results for each performance metric, averaged. It should be noted that in this experiment the training dataset is obtained using stratified sampling (as described in Section 3.4). The main purpose of this experiment is to evaluate the performance of the network embedding based regression models with respect to those trained using the topology based features. An important criterion in the evaluation step is to determine how well the predictions are made on nodes with the highest influence. Therefore, the recognition rate $r(f)$, precision function $p(f)$ and the $F1(f)$ score are all computed at $f = 0.1$. In other words, the top $10\%$ of the influential nodes are investigated.

#### Observations

Table 4.2 presents the benchmarking results of the several prediction models on all the networks. For each combination of {model and network}, the performance metrics are depicted. Additionally, the best performing prediction model for each network is presented in bold for both types of features. As can be observed, the RFR model has the best overall performance over all networks when using the topology features. In contrast, the Linear model performs the worst overall. While nearly all the models perform well when using topology features, the RFR outperforms those on all four performance metrics. Strikingly, in the case of the Linear model, the $r^2$ metric can be negative while $F1$ is acceptable. This could indicate that the linear model under predicts the nodal influence. A second observation is that the SVR model exhibits the best performance when the experiments are conducted using network embedding features. However, it is still sub-par in comparison with the RFR model where topology features are used instead.

#### Analysis

The observations in this experiment regarding the topology based features is not surprising. the RFR model can generally interpolate the non-linear patterns well within the training data in contrast to for example the Linear model. In addition, when considering topology based features, the dimensionality of the feature vector is much lower compared to the size of the training dataset. Therefore, sufficient data is available for not only the RFR, but also the Linear and SVR models to generalize well, resulting into the best prediction performance overall. In the case of the network embedding based features, each feature vector has at least a network dimension of 32 (for the networks, SOC1 and CIT1 the dimension is 128). SVR models are known to work well in this setting where the number of features is large relative to the size of the training dataset. As a result, its performance is expected to be superior than the Linear- and RFR models. This experiment concludes the following finding: When the degree-, eigenvector- and closeness centrality metrics are available for each node in the network, the RFR model is the best choice in the node influence prediction task. In the opposite case, network

embedding features are best utilized by the SVR model.

Table 4.2: Prediction performance of the models trained on a training dataset, consisting of 10% of the nodes in each network. Here, stratified sampling has been used to obtain the training data. Performance indicators have been computed on the remaining 90% of the unused nodes and averaged over 50 repetitions. For each model and each network, a comparison is shown between the baseline features ($\vec{X}_{Top}$) and the network embedding features ($\vec{X}_{Emb}$). The network embedding is based on the optimal parameter configuration.

| | | **Linear** | | **SVR** | | **RFR** | |
|---|---|---|---|---|---|---|---|
| **Network** | **Metric** | $\vec{X}_{Top}$ | $\vec{X}_{Emb}$ | $\vec{X}_{Top}$ | $\vec{X}_{Emb}$ | $\vec{X}_{Top}$ | $\vec{X}_{Emb}$ |
| **BIO1** | $r^2$ | 0.01 | -0.64 | 0.77 | 0.50 | 0.95 | 0.01 |
| | $r(0.1)$ | 0.89 | 0.29 | 0.90 | 0.60 | 0.89 | 0.19 |
| | $p(0.1)$ | 0.55 | 0.29 | 0.91 | 0.77 | 0.95 | 0.56 |
| | $F1(0.1)$ | 0.68 | 0.28 | 0.90 | **0.68** | **0.92** | 0.28 |
| **SOC1** | $r^2$ | 0.22 | -0.46 | 0.85 | 0.29 | 0.98 | 0.27 |
| | $r(0.1)$ | 0.93 | 0.40 | 0.93 | 0.55 | 0.93 | 0.38 |
| | $p(0.1)$ | 0.62 | 0.43 | 0.84 | 0.46 | 0.95 | 0.58 |
| | $F1(0.1)$ | 0.74 | 0.41 | 0.88 | **0.50** | **0.94** | 0.45 |
| **SOC2** | $r^2$ | -0.19 | -0.99 | 0.89 | 0.45 | 0.98 | 0.05 |
| | $r(0.1)$ | 0.95 | 0.13 | 0.94 | 0.62 | 0.94 | 0.16 |
| | $p(0.1)$ | 0.63 | 0.29 | 0.89 | 0.71 | 0.97 | 0.59 |
| | $F1(0.1)$ | 0.76 | 0.17 | 0.91 | **0.66** | **0.95** | 0.25 |
| **SOC3** | $r^2$ | -1.21 | -1.56 | 0.84 | 0.52 | 0.87 | 0.38 |
| | $r(0.1)$ | 0.77 | 0.41 | 0.66 | 0.58 | 0.81 | 0.38 |
| | $p(0.1)$ | 0.52 | 0.38 | 0.87 | 0.76 | 0.93 | 0.73 |
| | $F1(0.1)$ | 0.62 | 0.39 | 0.75 | **0.66** | **0.86** | 0.49 |
| **COM1** | $r^2$ | 0.47 | 0.19 | 0.89 | 0.77 | 0.98 | 0.49 |
| | $r(0.1)$ | 0.84 | 0.62 | 0.82 | 0.77 | 0.83 | 0.51 |
| | $p(0.1)$ | 0.75 | 0.58 | 0.87 | 0.81 | 0.98 | 0.73 |
| | $F1(0.1)$ | 0.79 | 0.59 | 0.84 | **0.79** | **0.90** | 0.59 |
| **COM2** | $r^2$ | 0.34 | -0.28 | 0.84 | 0.41 | 0.96 | 0.13 |
| | $r(0.1)$ | 0.90 | 0.46 | 0.93 | 0.59 | 0.93 | 0.34 |
| | $p(0.1)$ | 0.64 | 0.34 | 0.90 | 0.76 | 0.96 | 0.60 |
| | $F1(0.1)$ | 0.75 | 0.39 | 0.91 | **0.66** | **0.94** | 0.43 |
| **CIT1** | $r^2$ | -2.10 | -2.60 | 0.42 | 0.34 | 0.75 | 0.29 |
| | $r(0.1)$ | 0.75 | 0.47 | 0.51 | 0.53 | 0.73 | 0.49 |
| | $p(0.1)$ | 0.58 | 0.38 | 0.76 | 0.70 | 0.83 | 0.71 |
| | $F1(0.1)$ | 0.65 | 0.41 | **0.60** | **0.60** | **0.77** | 0.57 |

### 4.2.2. Effect of Sampling Strategy

#### Description

To determine whether the sampling strategy used to obtain the training dataset did contribute to the prediction performance, another set of experiments is conducted where the training data is sampled randomly. As the previous analysis already established that the RFR model for topology features and SVR model for the embedding features were the top performing frameworks, only these two models are considered in this experiment for better clarity.

#### Observations

Table 4.3 presents the results for the comparison study. It mainly shows that for both the topology- and network embedding features, both sampling strategies have a similar effect on the prediction performance. While in the RFR model, the stratified sampling strategy show slight improvement in the $F1$ score (for the networks SOC1, CIT1 and COM1), it is negligible. In the case of the embedding features based SVR model, the stratified sampling strategy does on average produce slightly better prediction performance. However, even in this case the improvement is small enough such that it is negligible.

#### Analysis

The observations conclude that the stratified sampling strategy used to obtain the training data does not improve the best performing regression models significantly, when compared to the case where a random sampling strategy is used to produce the training dataset. This suggests that the current prediction framework is applicable in both scenarios. The added value of the stratified sampling strategy is that the highly influential nodes which occur in the minority would be sampled as well. The results of this experiment suggests that the random sampling strategy already achieves this effect.

Table 4.3: Prediction performance of the best regression models with random- and stratified based training data. Each training data set consists of 10% of the nodes in the original network and the testing data set, the remaining nodes. All the results shown are an average over 50 repetitions. The highest value for the $F1$ score is presented in bold.

| | | RFR ($\vec{X}_{Top}$) | | SVR ($\vec{X}_{Emb}$) | |
|---|---|---|---|---|---|
| **Network** | **Metric** | **Random** | **Stratified** | **Random** | **Stratified** |
| **BIO1** | $r^2$ | 0.94 | 0.95 | 0.43 | 0.50 |
| | $r(0.1)$ | 0.89 | 0.89 | 0.58 | 0.60 |
| | $p(0.1)$ | 0.95 | 0.95 | 0.69 | 0.77 |
| | $F1(0.1)$ | **0.92** | **0.92** | 0.62 | **0.68** |
| **SOC1** | $r^2$ | 0.96 | 0.98 | 0.18 | 0.29 |
| | $r(0.1)$ | 0.93 | 0.93 | 0.49 | 0.55 |
| | $p(0.1)$ | 0.91 | 0.95 | 0.38 | 0.46 |
| | $F1(0.1)$ | 0.92 | **0.94** | 0.41 | **0.50** |
| **SOC2** | $r^2$ | 0.98 | 0.98 | 0.50 | 0.45 |
| | $r(0.1)$ | 0.93 | 0.94 | 0.66 | 0.62 |
| | $p(0.1)$ | 0.97 | 0.97 | 0.71 | 0.71 |
| | $F1(0.1)$ | **0.95** | **0.95** | **0.68** | 0.66 |
| **SOC3** | $r^2$ | 0.87 | 0.87 | 0.51 | 0.52 |
| | $r(0.1)$ | 0.81 | 0.81 | 0.58 | 0.58 |
| | $p(0.1)$ | 0.94 | 0.93 | 0.77 | 0.76 |
| | $F1(0.1)$ | **0.87** | 0.86 | **0.66** | **0.66** |
| **COM1** | $r^2$ | 0.99 | 0.98 | 0.76 | 0.77 |
| | $r(0.1)$ | 0.83 | 0.83 | 0.77 | 0.77 |
| | $p(0.1)$ | 0.97 | 0.98 | 0.80 | 0.81 |
| | $F1(0.1)$ | 0.89 | **0.90** | 0.78 | **0.79** |
| **COM2** | $r^2$ | 0.96 | 0.96 | 0.41 | 0.41 |
| | $r(0.1)$ | 0.93 | 0.93 | 0.59 | 0.59 |
| | $p(0.1)$ | 0.96 | 0.96 | 0.72 | 0.76 |
| | $F1(0.1)$ | **0.94** | **0.94** | 0.64 | **0.66** |
| **CIT1** | $r^2$ | 0.73 | 0.75 | 0.31 | 0.34 |
| | $r(0.1)$ | 0.72 | 0.73 | 0.52 | 0.53 |
| | $p(0.1)$ | 0.82 | 0.83 | 0.72 | 0.70 |
| | $F1(0.1)$ | 0.76 | **0.77** | 0.59 | **0.60** |

### 4.2.3. Effect of Embedding Dimension

#### Description

In this experiment, the network embedding dimension has been varied between $d \in \{2, 4, 8, 16, 32, 64, 128\}$. For each $d$ a separate network embedding has been produced, while keeping the other tuning parameters $p$ and $q$ the same as in the optimal setting. Ideally these would have to be tuned as well at different embedding dimensions, however due to the high run time complexity and the negligible effects of these parameters, this idea was put on hold. As usual, the SVR model is trained on 10% of the nodes in the network for 50 repetitions and the results have been averaged. The aim of this experiment is evaluate whether the prediction performance of the current best prediction model (the SVR framework) can be improved even further by decreasing the embedding dimension.

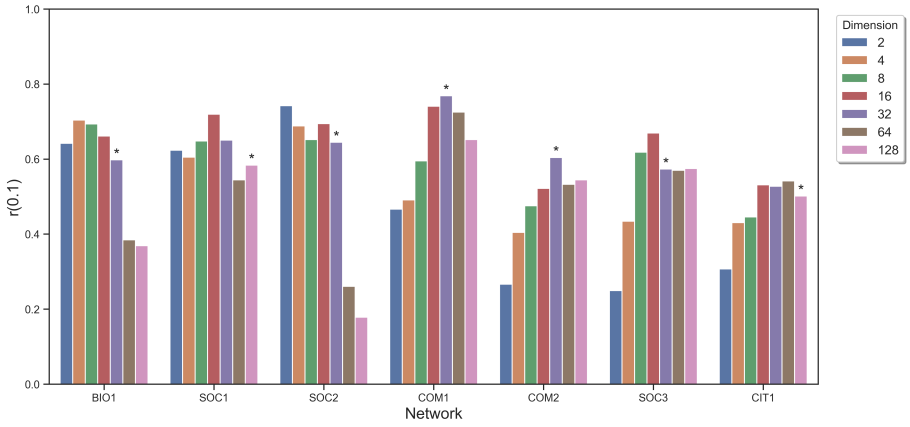
#### Observations

Figure 4.10 shows the recognition rate $r(0.1)$, precision $p(0.1)$ and the F1 score $F1(0.1)$ for the cases with a different embedding dimension. The following observations can be made:

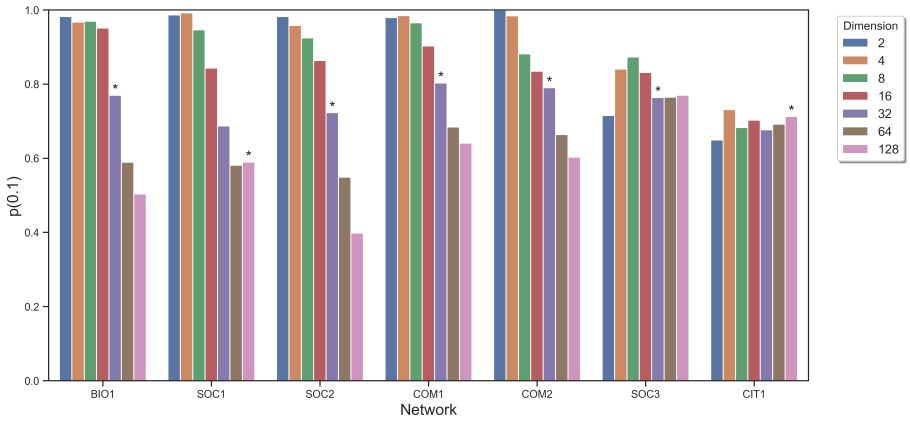
- For the networks {BIO1, SOC1, SOC2}, decreasing $d$ will slightly improve the recognition rate, while for the remaining networks it has the opposite effect. The optimal embedding of the networks COM1, COM2 and CIT1 (in the figure marked by an asterisk) exhibits the highest recognition rate and decreasing the embedding dimension has a negative effect on the prediction performance. There seems to be an optimum value for the embedding dimension.
- The precision metric $p(0.1)$ consistently increases as $d$ decreases for all the networks except SOC3 and CIT1. The $F1(0.1)$ score, which is the harmonic mean of the previous two measures, therefore displays two behaviours: (1) for the networks {BIO1, SOC1, SOC2, COM1 and SOC3} lowering the dimension to $d = 4$, $d = 16$, $d = 2$, $d = 16$ and $d = 16$ respectively, will result in the best prediction performance and (2) changing the embedding dimension for the COM2 and CIT1 networks does not necessarily lead to an improvement in the nodal influence prediction.

#### Analysis

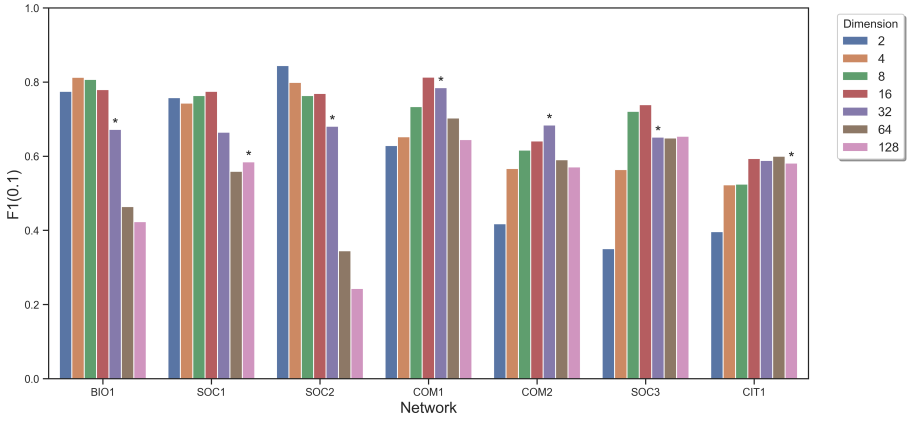
Considering the previous observations, two phenomena can be identified. Decreasing the dimension of the network embedding results into a prediction model with a better prediction performance. This is expected as for every supervised learning model, when the size of the training dataset is small, reducing the number of features may increase the prediction performance. This can be seen on the improvement in the precision. However, the results suggest that the optimization of the network embedding dimension is network specific.




(a) $r(0.1)$




(b) $p(0.1)$




(c) $F1(0.1)$

Figure 4.10: Performance measurements for different values of embedding dimension. Each value shown is averaged over 50 runs, where the SVR model has been used for training and prediction. As in the previous cases, the training data set comprised of 10% of the network nodes.

### 4.2.4. Increasing the Training Data

#### Description

The effect of increasing the training data set is investigated in this experiment. From classical machine learning theory, a larger training dataset will positively affect its prediction performance in nearly all cases. The goal of this experiment is to evaluate how much better the model will perform when given more data and whether the initial size of $10\%$ had been chosen appropriately. Here, only the results for the SVR model using embedding features is shown as this has been deemed the best performing prediction model in the previous analysis. As presented in Section 4.2.1, the RFR model based on topology features already exhibits near perfect prediction performance. Therefore, in this experiment the topology based features have been omitted. As usual, all experiments are conducted using the optimal embedding parameters and the results averaged over 50 runs.

#### Observations

Figure 4.11 shows the $F1(0.1)$ score of the SVR prediction model with increasing training data. As expected, when the model is trained on more data, it better predicts the influence of the top 10% of the most influential nodes. However, relative to the default training data size of 10% of the nodes in the network, when the training data size is doubled, only a slight increase in the $F1(0.1)$ score is observed. For all the networks except SOC2 and CIT1, this increase is $\leq 10\%$. On the other hand, the coefficient of determination ($r^2$) has a larger increase when the training data set is doubled (see Figure 4.12), suggesting that a larger training dataset affects the prediction of the low influential nodes positively to a larger extent in comparison with the high influential nodes in the minority.

#### Analysis

The previous observation suggests that more data positively affects the predictions of the nodal influence. The training data set consisting of $10\%$ of the nodes in the network is sufficient, as doubling the training dataset produces minimal improvement. This result is also expected as the SVR model works exceptionally well in a setting with a low quantity of training data while the number of features is large.

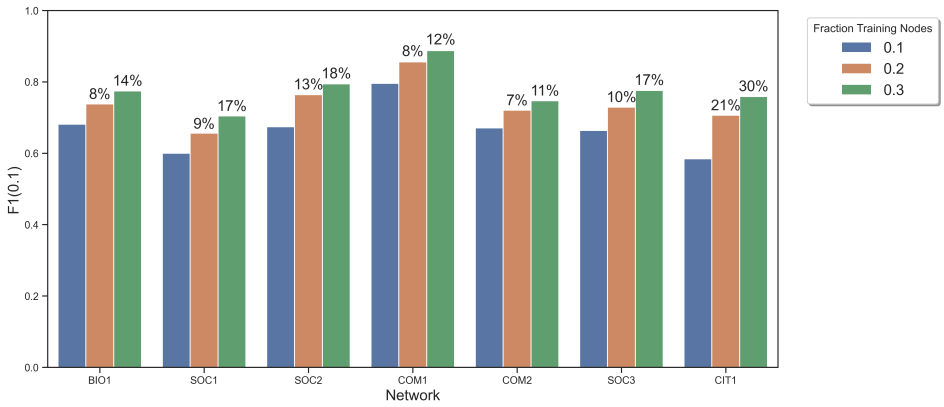



Figure 4.11: F1(0.1) measure for the SVR model, where the size of the training data is increased by 10% and 20%. Each embedding is generated using the optimal parameters in Node2Vec. The percentage on top of each bar represents the increase relative to the default case.

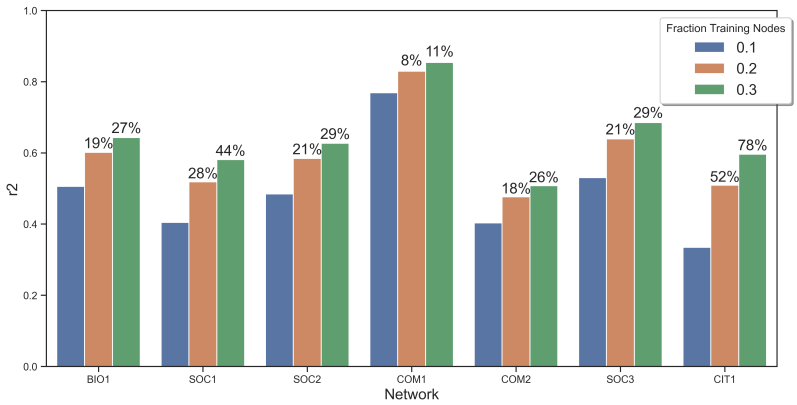



Figure 4.12: $r^2$ score for the SVR model, where the size of the training data is increased by 10% and 20%. Each embedding is generated using the optimal parameters in Node2Vec. The percentage on top of each bar represents the increase relative to the default case.

### 4.2.5. Effect of combining the embedding- and topology features

#### Description

In Section 4.2.3 it was found that the network embedding based SVR prediction model could further be optimized by decreasing the embedding dimension. In this experiment, a second optimization step is investigated where topology features are concatenated with the network embedding based features. Therefore, the SVR model has been re-trained on different feature sets, where in each case the embedding vector has been combined with the degree-, closeness- and eigenvector centrality metric. Note that first individual contributions are investigated and afterwards the performance improvement by adding all centrality metrics. All results in this case has been generated with the optimized embedding dimension for each network. Furthermore, the prediction performance of the RFR model with only topology based (individual) features have been presented as well in order to better position the benefit of using the network embedding based features.

#### Observations

Table 4.4 presents the prediction performance (F1(0.1) metric) for the SVR model with the combined features (network embedding- and topology based), while Table 4.5 the results for the RFR model with only (individual) topology absed features. The following observations can be made:

- **Network embedding features + degree centrality:** For all networks, adding the degree centrality to the network embedding based features results into a better F1(0.1) score for all the networks in comparison with using only the network embedding based features. However, as can be seen in Table 4.5, using only the degree centrality as a single feature in the RFR model will result into a prediction performance which is on par with the previous case (the difference between the F1(0.1) score of both methods is small).
- **Network embedding features + eigenvector centrality:** An improvement in prediction performance is achieved when compared to the case with only network embedding based features in the SVR model. Table 4.5 shows that the eigenvector centrality without the addition of the network embedding based features is a slightly better predictor for all the networks except BIO1.
- **Network embedding features + closeness centrality:** In general the addition of the closeness centrality metric improves the F1(0.1) score slightly. In contrast to the previous two topology metrics, this feature combination predicts the nodal influence of the top nodes better in comparison with the RFR model where only the eigenvector centrality metric has been used.
- **Network embedding features + all centrality metrics:** As in the previous cases, adding all topology based features to the network embedding based SVR model does lead to an improvement in the influence prediction of the top 10% of the most influential nodes. However, for all the networks except COM2 and CIT1, adding individual topology features can achieve

Table 4.4: Overview of the F1(0.1) scores for the SVR model where embedding- and topology features have been combined. The following features have been combined with the embedding vectors: Degree Centrality (DC), Closeness Centrality (CC) and Eigenvector Centrality (EC). In the last column, all three centrality features have been added to the embedding vector of each node. Results are shown for the SVR model and as usual averaged over 50 repetitions.

| F1(0.1) | SVR | | | | |
|---|---|---|---|---|---|
| | $\vec{X}_{Emb}$ | $\vec{X}_{Emb+DC}$ | $\vec{X}_{Emb+EC}$ | $\vec{X}_{Emb+CC}$ | $\vec{X}_{Emb+all\ Top}$ |
| **BIO1** | 0.79 | 0.83 | 0.88 | **0.91** | 0.87 |
| **SOC1** | 0.75 | 0.84 | **0.85** | 0.84 | **0.85** |
| **SOC2** | 0.83 | 0.88 | 0.90 | **0.91** | 0.89 |
| **SOC3** | 0.72 | **0.79** | 0.73 | 0.75 | 0.77 |
| **COM1** | 0.81 | 0.83 | **0.85** | 0.81 | 0.84 |
| **COM2** | 0.67 | 0.82 | 0.78 | 0.83 | **0.85** |
| **CIT1** | 0.56 | 0.63 | 0.59 | 0.60 | **0.65** |

Table 4.5: Overview of the F1(0.1) scores for the RFR model where individual topology features have been have been used in the prediction framework: Degree Centrality (DC), Closeness Centrality (CC) and Eigenvector Centrality (EC). In the last column, all three centrality features have been concatenated. Results are averaged over 50 repetitions.

| F1(0.1) | RFR | | | |
|---|---|---|---|---|
| | $\vec{X}_{DC}$ | $\vec{X}_{EC}$ | $\vec{X}_{CC}$ | $\vec{X}_{all\ Top}$ |
| **BIO1** | 0.78 | 0.83 | 0.89 | **0.92** |
| **SOC1** | 0.76 | **0.95** | 0.80 | 0.94 |
| **SOC2** | 0.90 | 0.94 | 0.91 | **0.95** |
| **SOC3** | 0.75 | 0.77 | 0.42 | **0.86** |
| **COM1** | 0.88 | 0.89 | 0.77 | **0.90** |
| **COM2** | 0.80 | 0.87 | 0.91 | **0.94** |
| **CIT1** | 0.48 | 0.66 | 0.55 | **0.77** |

the same result. Therefore, it is not necessary to add all three centrality metrics to the network embedding based features. When comparing the prediction capability of the two feature sets $\vec{X}_{Emb+all\ Top}$ and $\vec{X}_{all\ Top}$, the latter is better when used in the RFR model. More specifically, when all three topology based features are available, then the network embedding does not add additional benefit to the prediction framework.

**Analysis**

This experiment denotes that the prediction model should be constructed based on the availability of network topology based features. When the DC, EC and CC are all available, network embedding information will not enhance the prediction of the nodal influence. Network embedding is beneficial to the current prediction task in the following case: either single or none of the topology based centrality metrics are available. When only one of either DC, EC or CC metrics are available, it is best to add it to the network embedding based features in the SVR model.

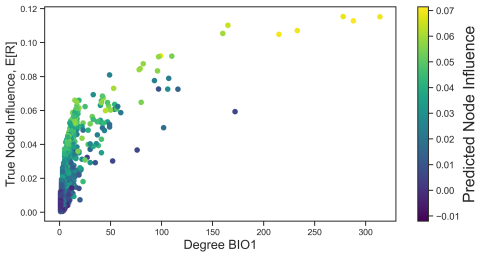
## 4.3. Network Topology based Analysis

One of the broad objectives in network science regarding network embedding algorithms (and also partly of this research project) is being able to relate which information of the network topology is captured in the embedding vectors of each node. In this section, a brief exploratory analysis is conducted on this matter. Therefore, a brief qualitative comparison is made between the network embedding and the network topology on its ability to predict the influence of each node. More specifically, an investigation is conducted on possible correlations between the individual topology based centrality metrics and the network embedding information by means of the node influence prediction task.

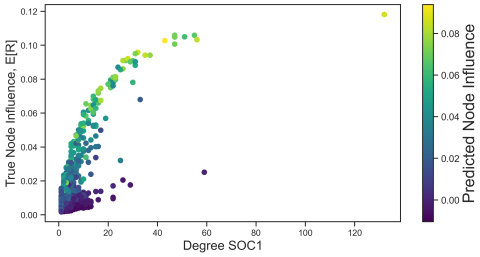
Figures 4.13, 4.14 and 4.15 each show the effect of the degree-, eigenvector- and closeness cen-

trality metrics on the true nodal influence, respectively for each network. Additionally, each node is color coded with the predicted nodal influence by the optimal network embedding based SVR model. It should be noted that in this case, the embedding dimension is also reduced for a better prediction performance, according to the results in Section 4.2.3.

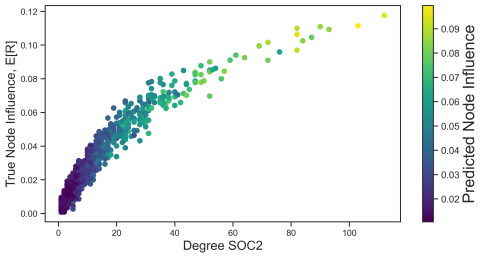
The results show that in general each individual centrality metric correlates well with the true nodal influence: if the centrality metric of a node has a higher value, it will have a larger influence in the network. However, these findings are not consistent for some networks. For the networks such as SOC3 and CIT1 it is clear that the individual centrality metrics are not a good predictor for the true nodal influence. This can be seen by the larger variation in the true nodal influence at any specific value of either the degree-, eigenvector- or closeness centrality metrics (x-axis). This observation is consistent with the quantitative results in the previous section, where the F1(0.1) score for the nodal influence prediction using the RFR model is lower in comparison with the other networks. The same conclusion holds for the nodal influence prediction using the network embedding based features. In the CIT1 network, it is apparent that even the network embedding based SVR model is not able to properly quantify the top influential nodes. For the networks, where the individual centrality metrics are good predictors (for example SOC1 and SOC2), the network embedding based SVR model also performs well. This hints at the possibility that the predictive power of the network embedding based features are also correlated with the predictive power of the topology based features in regards to the nodal influence.



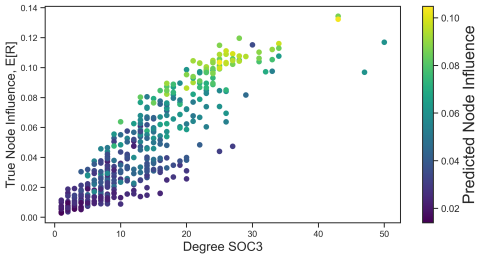

(a) Protein Protein Interaction (BIO1).



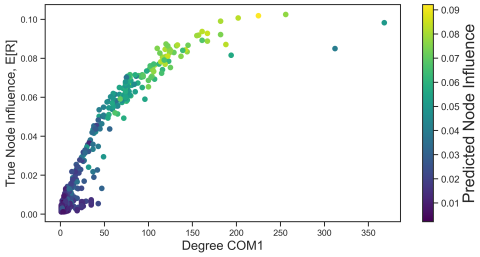

(b) Facebook pages food (SOC1).



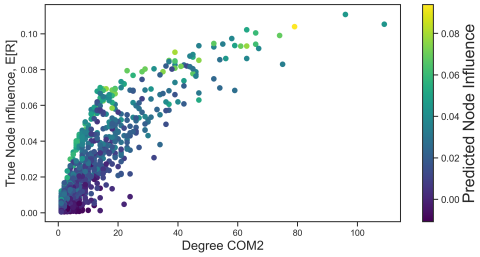

(c) Facebook messages interaction (SOC2).



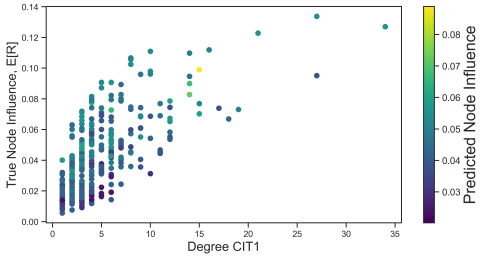

(d) Human contact (SOC3).




(e) DNC emails (COM1).

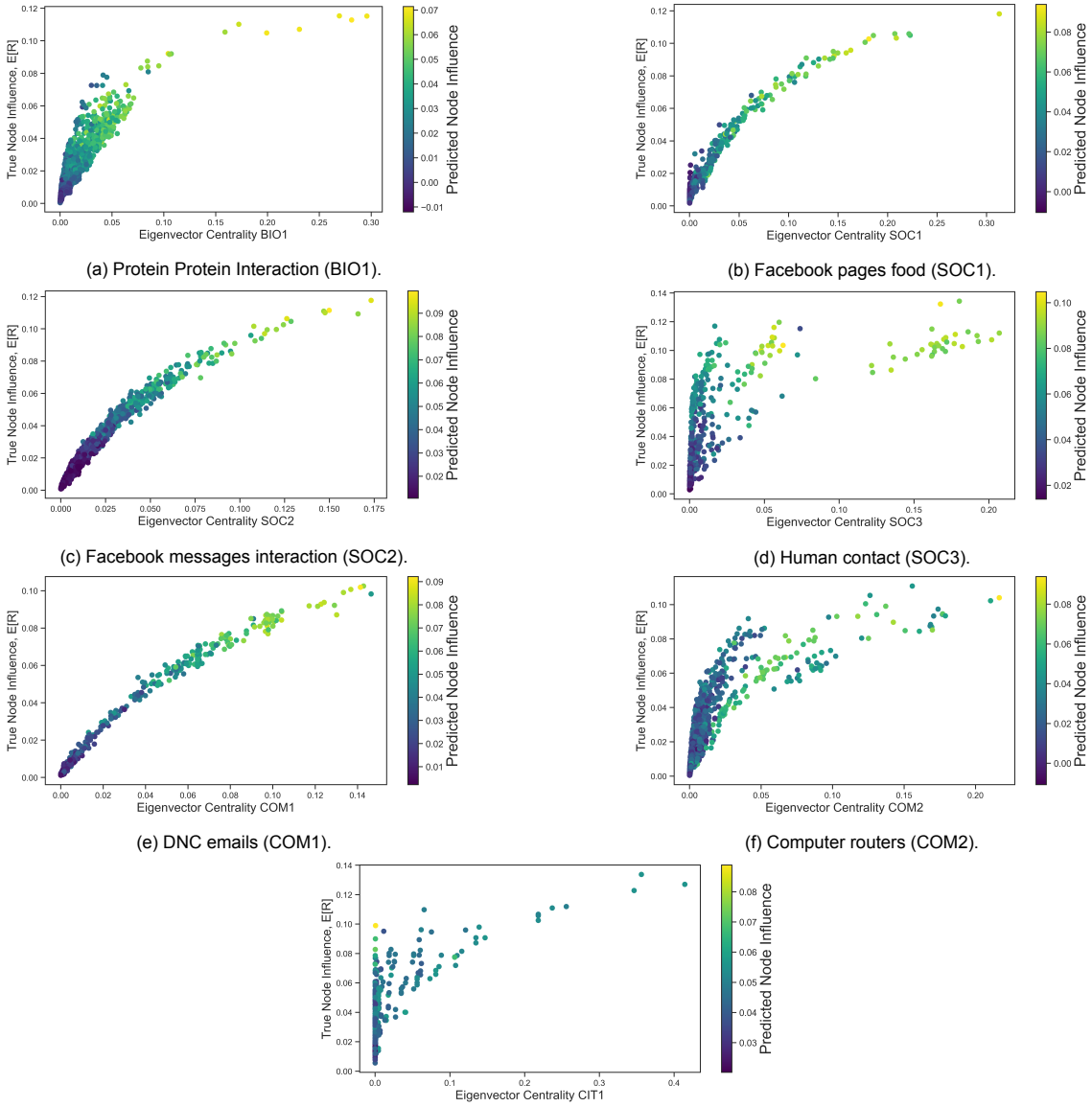



(f) Computer routers (COM2).




(g) Citation netscience (CIT1).

Figure 4.13: Scatter plot between the true nodal influence (final epidemic size) versus the degree centrality. The color coding on each point (node) represents the predicted nodal influence by the best prediction model (SVR) based on embedding features only.




(a) Protein Protein Interaction (BIO1).

(b) Facebook pages food (SOC1).

(c) Facebook messages interaction (SOC2).

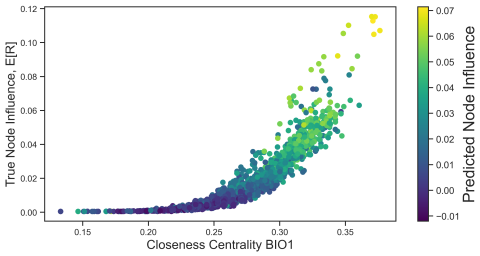
(d) Human contact (SOC3).

(e) DNC emails (COM1).

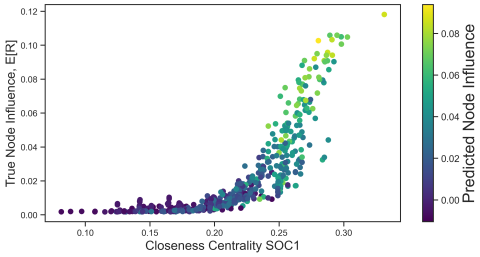
(f) Computer routers (COM2).

(g) Citation netscience (CIT1).

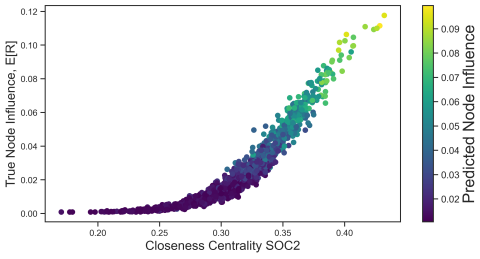
Figure 4.14: Scatter plot between the true nodal influence (final epidemic size) versus the eigenvector centrality. The color coding on each point (node) represents the predicted nodal influence by the best prediction model (SVR) based on embedding features only.



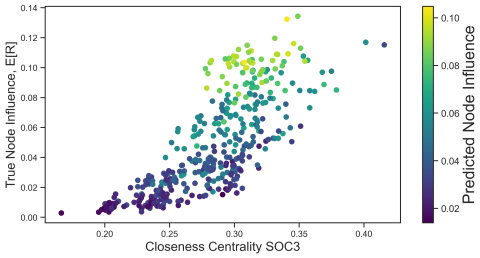

(a) Protein Protein Interaction (BIO1).



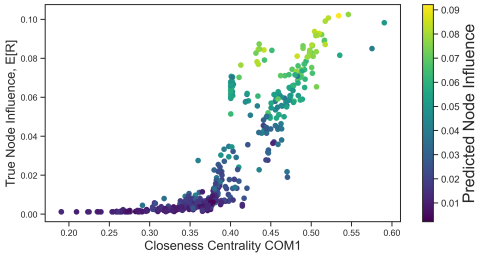

(b) Facebook pages food (SOC1).



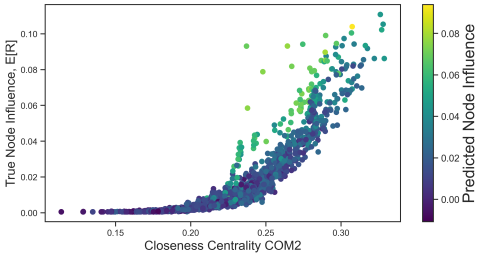

(c) Facebook messages interaction (SOC2).



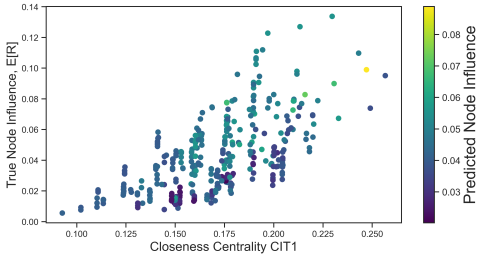

(d) Human contact (SOC3).




(e) DNC emails (COM1).




(f) Computer routers (COM2).




(g) Citation netscience (CIT1).

Figure 4.15: Scatter plot between the true nodal influence (final epidemic size) versus the closeness centrality. The color coding on each point (node) represents the predicted nodal influence by the best prediction model (SVR) based on embedding features only.

# 5

# Discussion & Conclusion

In this chapter a brief discussion is given of the experimental results presented in the previous chapter. For a convenient overview it is divided into two parts: (1) Investigation of the relation between the network embedding and the network topology, and (2) The application of the network embedding features in order to predict the nodal influence. In both topics, a reflection is given on the assumptions and the underlying methods which might aid into explaining the observed results. Finally, the conclusion is presented where the research questions are answered.

## 5.1. Discussion: Relation between Network Embedding and Network Topology

The main objective of this project was to utilize network embedding based features to predict the nodal influence of the nodes in a given network. In this case it was assumed: (1) that the network topology was known and (2) the nodal influence for a relatively small subset of its nodes were known. In a preliminary exploratory study, an investigation was performed on whether the embedding features of a node correlated with its proximity in the network topology. The intuition behind this investigation has been that nodes in close proximity in the network topology have a similar epidemic influence. Therefore, the first set of exploratory experiments aimed to confirm whether the optimal network embedding did indeed capture the distance between node pairs in the network topology. In order to achieve this feat, two similarity (distance) metrics have been compared for every node pair: the shortest path distance (SPD) and the cosine similarity (CS). In addition to the various proven advantages of network embedding methods in machine learning applications, the findings of this experiment highlight yet another advantage of these class of algorithms [12]. In particular, it has been demonstrated that the CS between the embedding vectors of any two nodes is indicative of how far apart the nodes are located in the network itself. However, the findings also suggest that for a minority of the node pairs where the SPD is relatively larger, the CS tends to be higher as well. While these results are promising, several limitations can be identified with the current approach, which need further exploration:

- In order to unravel the relation between the network embedding and the network topology, only the SPD and CS distance metrics have been explored in this study. In the context of network topology, definitions of distance measures other then the SPD may yield different results.
- The observed correlation pattern between the SPD and CS in this experiment is limited to network embedding based features which are optimized on the link prediction task. Optimizing the network embedding on other tasks can lead to embedding features with a better predictive capability for the nodal influence. For example, one could directly optimized the embedding parameters $p$ and $q$ in Node2Vec, based on the performance of the predictive models.
- Lastly, the found relation between the SPD and CS may be explained and be specific to the random walk based embedding methods, where local- and global neighbourhood information is preserved in the embedding features. Non-random walk based embedding methods may yield different correlation patterns when applying the same correlation analysis as in this work.

## 5.2. Discussion: Node Influence Prediction using Network Embedding

In the second part of this project, the extent to which a set of classical prediction models (Linear, RFR and SVR) were able to utilize the information of the optimal structure preserving network embedding were investigated. In particular the size of the training data set was kept at 10% of the total number of nodes in the network in order to reflect a real world scenario with an ongoing epidemic spreading process. In order to position the prediction capability of the network embedding based features, the results have been compared to a baseline model. This baseline model utilized a feature vector for each node with three network topology metrics: the degree-, eigenvector and closeness centrality metrics.

Among the three prediction models used, it was found that network embedding based features were best utilized by the SVR model, while the topology based metrics by the RFR model. This finding affirmed that the dimensionality of the feature vector played a key role in the prediction performance: the SVR model excels when the number of features are large, thus being the most suitable for the network embedding based features. Bearing these aspects in mind, the comparison of the prediction results based on the F1(0.1) score revealed that the best network embedding based prediction framework did not perform on par in comparison with the baseline (RFR model with all three centrality metrics). This result was expected due to the large dimensionality of the network embedding features ($> 32$ for all networks). When the embedding dimension was optimized (reduced) in the SVR model, the prediction performance increased for all networks except for the author citation network (CIT1) and the computer router network (COM2). However, even after this optimization step the RFR model utilizing the degree-, eigenvector and closeness centrality metrics has been proven to be superior to the SVR model utilizing the network embedding based features. This observation is not surprising as work conducted in [6] shows that a combination of a local- and global based topology metric is sufficient to achieve a near perfect prediction of the nodal influence.

In spite of the previous finding, network embedding based features does have its niche utility. This is highlighted by the outcome of the set of experiments where both topology- and embedding based features have been combined. When the closeness centrality metric is available, the addition of the network embedding based features will result into a slightly better prediction performance according to the F1(0.1) score. This observation also holds for the degree- and eigenvector centrality as well, but only for specific networks. Due to the limited number of networks investigated in this research project, the exact reason for this latter observation was not investigated. It is hypothesized that the homogeneity of the degree distribution of a network may close the gap of the prediction performance between network embedding- and topology based features. Another argument can be made on why the network embedding based features might be preferable to the classical topology based metrics: scalability. While the networks investigated in this project contained at most in the order of $10^3$ nodes, real world networks can span millions of nodes. In this case, network embedding features can be obtained more efficiently in comparison with for example the closeness centrality, where all possible pairwise shortest paths need to be computed. The degree centrality in contrast to the closeness- and eigenvector centrality is a metric that can be obtained relatively efficiently as long as the complete network structure is known. When this metric is used in combination with the network embedding based features, an improvement in the prediction performance is observed. Therefore, it is recommended to always incorporate the degree centrality in the SVR model using the network embedding based features.

It is worthwhile to note that the prediction results of this work should be taken into consideration with the following factors and assumptions:

- As the focus of this work was to evaluate the utility of the network embedding based features in the node influence prediction task, little attention has been paid to optimize the machine learning methods (Linear-, SVR- and RFR models). As described in Section 3.4.1 the training dataset was processed based on two factors: (1) the method of splitting the nodes into validation- and training data and (2) class imbalance. While the experimental results demonstrated that there was a negligible difference between the random- and stratified sampling strategy, the class imbalance issue was not tackled in this work as the main focus was not to dive too much into the machine learning aspects. It is believed that in a future work this aspect should be investigated in detail, as this might produce better network embedding based prediction models.

- The true nodal influence of each node has been obtained by means of a SIR spreading process. Here, the infection- and recovery rates were chosen such that the maximum prevalence level of the SIR process was approximately 10%. The current findings in this work is based on this parameter setting. Different values for these parameters may paint a better picture on the scenarios where network embedding based features are preferable in the node influence prediction task.

## 5.3. Conclusion

The main objective of this thesis project was to investigate whether a structure preserving embedding of a network could be used to predict the epidemic influence of its nodes. In order to tackle this challenge, the following master research question was defined:

**How can network embedding based features be utilized in order to predict the information diffusion capability (influence) of a node in a network?** In order to answer this main question, a set of sub research questions in Chapter 1 were defined. These sub research questions with their corresponding answers are as follows:

1. **How is the proximity between the nodes in the network topology captured within in the network embedding, which optimally preserves the network structure?**
   As a first step, the optimal network embedding which best preserved the structure of the network was obtained by optimizing it according to the link prediction task. Subsequently, the correlation analysis affirmed that for any two arbitrary nodes in the network, its distance (or similarity) in the network embedding space reflects the shortest path distance in the network topology. More specifically, the closer two nodes are in a network, the higher the similarity in the network embedding space. As a consequence, the observations concluded that network embedding based features are a good candidate in predictive models for the nodal influence, as nodes closer in proximity in the network topology are expected to have a similar nodal influence.

2. **In the presence of limited training data, how effective are network embedding based features in contrast to the classical network topology based features in the node influence prediction task?**
   To answer this research question, the main set of experiments focused on contrasting the node influence prediction performance of three machine learning models (Linear, SVR and RFR) with two sets of features: network embedding based and classical topology based (degree-, eigenvector- and closeness centrality). It was found that the RFR model utilizing all three classical centrality metrics outperformed the best network embedding based SVR model. Further optimization of the network embedding dimension did indeed result into better prediction performance over several networks, however even in this case, the SVR model did not outperform the baseline RFR model. Thus, it is concluded that network embedding in the presence (availability) of all the classical degree-, eigenvector- and closeness centrality metrics, is not recommended to be used in the node influence prediction task.

3. **Does the incorporation of the network topology based features into the network embedding based prediction models improve the prediction of the nodal information diffusion capability?**
   In order to answer this research question, the best network embedding based models were further enhanced by (1) individual centrality metrics and (2) all three centrality metrics. The experiments on the node influence prediction showed that in comparison to the original SVR model with network embedding based features, adding either individual or all centrality metrics would enhance the prediction performance. However, in the case where all centrality metrics are available, the RFR model utilizing only the centrality metrics as features is the best model. Here, it is not recommended to use network embedding based features. The network embedding based features should only be utilized when a single centrality metric is available (either degree-, eigenvector or closeness centrality).

# 6

# Future Work

The findings in this work demonstrate that there is still room for improvement in both sets of experiments: analysis of the network embedding features and optimizing the predictive models for the node influence prediction task.

## 6.1. Relation between Network Embedding and Network Topology

In order to investigate and affirm that the network embedding indeed was suitable to be used in the node influence prediction task, a correlation analysis was performed between the pairwise shortest path distance (SPD) and cosine similarity (CS). Another experiment that could have been conducted was to perform the same correlation analysis between two nodal pairwise metrics: (1) the shortest path distance between the nodes in the network topology and (2) the difference between the influence of the pairwise nodes. This analysis could demonstrate that nodes which are closer in the network topology have a similar influence.

The network embedding used to generate the results in this work was optimized on the link prediction task such that it best preserved the network topology. In order to achieve this the AUC metric was used to assess the link prediction performance. One question that arose was: *Does optimizing the embedding on the link prediction task indeed result in a network embedding best suitable for the node influence prediction task?* It is expected that nodes which are closer proximity in the network topology will have a similar influence. As a result, optimizing the embedding to better preserve the local community structures might yield different results. Lastly, the experiments in this work is based on a single network embedding method: Node2Vec. As surveyed in [12] a plethora of network embedding methods exist that could potentially be more suitable for the node influence prediction task.

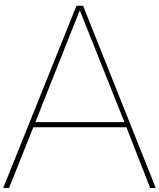
## 6.2. Node Influence Prediction using Network Embedding

Chapter 5 lists several limitations that should be considered when analyzing the results regarding the node influence prediction performance when using both network embedding- and topology based features. Here, the performance of the predictive models could be optimized even further by addressing the following aspects:

- In this work, little attention was paid to optimize the predictive models on machine learning aspects. Therefore, the three predictive models were implemented with default parameter settings in accordance with the state of the art. In a next study, these learning models could be further improved by tackling the class imbalance problem.

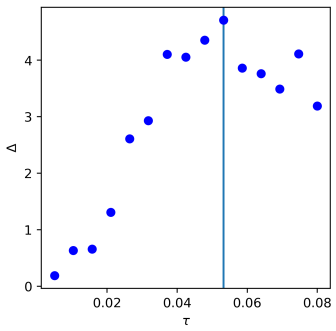
- The learning models (Linear, SVR and RFR) applied in this work belong the class of *supervised learning methods*. In this case, it is assumed that the features and nodal influence of a subset of the nodes in the network are known. Since the full network structure is also available, the relational structure of the neighbouring nodes can also be utilized during model training using *semi-supervised learning methods*.

- In this work only seven networks have been used in the node influence prediction task. The prediction performance analysis can be extended even further by analyzing more networks with different properties.
- In order to obtain the nodal influence for each node in the network, the SIR spreading process was simulated at a specific infection- and recovery rate. Here, it was ensured that the maximal nodal influence remained $\approx 0.1$. As shown in Table 3.3 in Section 3.3, in some cases the effective transmission rate would be higher than the epidemic threshold. Another interesting parameter setting where the current experiments should be conducted is the case where the effective transmission rate is closer to the epidemic threshold.

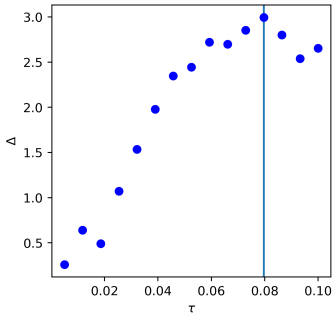


# Epidemic Threshold

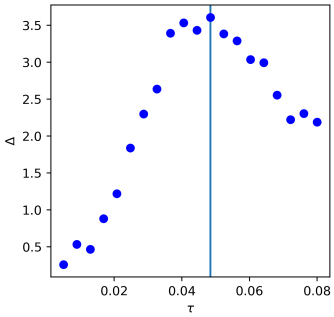
The epidemic threshold $\tau_c$ in the case when a SIR epidemic process unfolds on a network is determined numerically using the method outlined in [6] and [50].



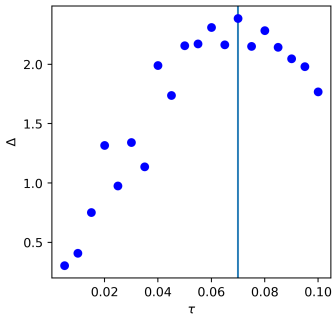

(a) Protein Protein Interaction (BIO1)



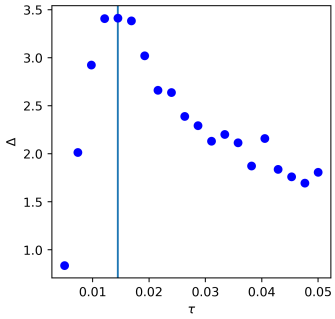

(b) Facebook pages food (SOC1)



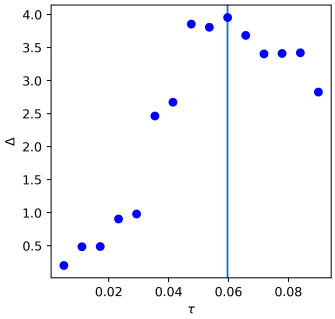

(c) Facebook messages interaction (SOC2)



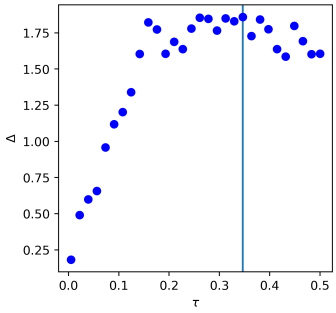

(d) Human contact (SOC3)




(e) DNC emails (COM1)




(f) Computer routers (COM2)




(g) Citation netscience (CIT1)

Figure A.1: Numerical estimate of the epidemic threshold ($\tau_c$) for the networks investigated in this work. The vertical line denotes the value of the effective transmission rate approximately equal to the epidemic threshold. The horizontal x-axis denotes the effective transmission rate at which the epidemic variability Δ has been computed according to the method described in [6] and [50].

# B

# Correlation Analysis

Table B.1: SPearman correlation coefficient between the Cosine Similarity (CS), Shortest Path Distance (SPD) and Resistance Distance (RD) metrics of all possible node pairs in the network.

| Network | SPearman Correlation | | |
|---|---|---|---|
| | $p_{(SPD,\ CS)}$ | $p_{(SPD,\ RD)}$ | $p_{(RD,\ CS)}$ |
| **BIO1** | 0.36 | 0.64 | 0.19 |
| **SOC1** | 0.38 | 0.73 | 0.00 |
| **SOC2** | 0.42 | 0.71 | 0.29 |
| **SOC3** | 0.18 | 0.55 | 0.10 |
| **COM1** | 0.25 | 0.46 | 0.21 |
| **COM2** | 0.32 | 0.73 | 0.15 |
| **CIT1** | 0.53 | 0.76 | 0.24 |

# Bibliography


[1] Albert-László Barabási. “Network science”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1987 (2013), p. 20120375.

[2] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. “Complex networks: Structure and dynamics”. In: *Physics reports* 424.4-5 (2006), pp. 175–308.

[3] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*. Vol. 290. Macmillan London, 1976.

[4] Peer Bork, Lars J Jensen, Christian Von Mering, Arun K Ramani, Insuk Lee, and Edward M Marcotte. “Protein interaction networks from yeast to human”. In: *Current opinion in structural biology* 14.3 (2004), pp. 292–299.

[5] Paula Branco, Luís Torgo, and Rita P Ribeiro. “SMOGN: a pre-processing approach for imbalanced regression”. In: *First international workshop on learning with imbalanced domains: Theory and applications*. PMLR. 2017, pp. 36–50.

[6] Doina Bucur. “Top influencers can be identified universally by combining classical centralities”. In: *Scientific reports* 10.1 (2020), pp. 1–14.

[7] Doina Bucur and Petter Holme. “Beyond ranking nodes: Predicting epidemic outbreak sizes by network centralities”. In: *PLOS Computational Biology* 16.7 (2020), e1008052.

[8] Alberto Ceria, Klemens Köstler, Rommy Gobardhan, and Huijuan Wang. “Modeling airport congestion contagion by heterogeneous SIS epidemic spreading on airline networks”. In: *Plos one* 16.1 (2021), e0245043.

[9] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. “Machine learning on graphs: A model and comprehensive taxonomy”. In: *arXiv preprint arXiv:2005.03675* (2020).

[10] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. “Graph representation learning: a survey”. In: *APSIPA Transactions on Signal and Information Processing* 9 (2020).

[11] Haochen Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “A tutorial on network embeddings”. In: *arXiv preprint arXiv:1808.02590* (2018).

[12] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. “A survey on network embedding”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.5 (2018), pp. 833–852.

[13] Narsingh Deo. *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2017.

[14] Ernesto Estrada. *The structure of complex networks: theory and applications*. Oxford University Press, 2012.

[15] Rob M. Ewing, Peter Chu, Fred Elisma, Hongyan Li, Paul Taylor, Shane Climie, Linda McBroom-Cerajewski, Mark D. Robinson, Liam O’Connor, Michael Li, Rod Taylor, Moyez Dharsee, Yuen Ho, Adrian Heilbut, Lynda Moore, Shudong Zhang, Olga Ornatsky, Yury V. Bukhman, Martin Ethier, Yinglun Sheng, Julian Vasilescu, Mohamed Abu-Farha, Jean-Philippe P. Lambert, Henry S. Duewel, Ian I. Stewart, Bonnie Kuehl, Kelly Hogue, Karen Colwill, Katharine Gladwish, Brenda Muskat, Robert Kinach, Sally-Lin L. Adams, Michael F. Moran, Gregg B. Morin, Thodoros Topaloglou, and Daniel Figeys. “Large-scale Mapping of Human Protein–Protein Interactions by Mass Spectrometry”. In: *Molecular Systems Biol.* 3 (2007).

[16] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[17] Shuai Gao, Jun Ma, Zhumin Chen, Guanghui Wang, and Changming Xing. “Ranking the spreading ability of nodes in complex networks based on local structure”. In: *Physica A: Statistical Mechanics and its Applications* 403 (2014), pp. 130–147.

[18] Palash Goyal and Emilio Ferrara. “Graph embedding techniques, applications, and performance: A survey”. In: *Knowledge-Based Systems* 151 (2018), pp. 78–94.

[19] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864.

[20] Nesrine Hafiene, Wafa Karoui, and Lotfi BEN ROMDHANE. “Influential nodes detection in dynamic social networks: A Survey”. In: *Expert Systems with Applications* (2020), p. 113642.

[21] Javier Martın Hernández and Piet Van Mieghem. “Classification of graph metrics”. In: *Delft University of Technology: Mekelweg, The Netherlands* (2011), pp. 1–20.

[22] *Human proteins (Figeys) network dataset – KONECT*. Oct. 2017. URL: `http://konect.cc/networks/maayan-figeys`.

[23] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. “What’s in a crowd? Analysis of face-to-face behavioral networks”. In: *Journal of theoretical biology* 271.1 (2011), pp. 166–180.

[24] István Z Kiss, Joel C Miller, Péter L Simon, et al. “Mathematics of epidemics on networks”. In: *Cham: Springer* 598 (2017).

[25] Douglas J Klein and Milan Randić. “Resistance distance”. In: *Journal of mathematical chemistry* 12.1 (1993), pp. 81–95.

[26] Max Kuhn and Kjell Johnson. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019.

[27] Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex networks: principles, methods and applications*. Cambridge University Press, 2017.

[28] Jure Leskovec. *Graph Representation Learning*. `http://snap.stanford.edu/class/cs224w-2018/handouts/09-node2vec.pdf`. Accessed: 2021–19-06. 2018.

[29] C Li, H Wang, W De Haan, CJ Stam, and Piet Van Mieghem. “The correlation of metrics in complex networks with applications in functional brain networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2011.11 (2011), P11018.

[30] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. “Deepcas: An end-to-end predictor of information cascades”. In: *Proceedings of the 26th international conference on World Wide Web*. 2017, pp. 577–586.

[31] Cong Li, Qian Li, Piet Van Mieghem, H Eugene Stanley, and Huijuan Wang. “Correlation between centrality metrics and their application to the opinion model”. In: *The European Physical Journal B* 88.3 (2015), pp. 1–13.

[32] David Liben-Nowell and Jon Kleinberg. “The link-prediction problem for social networks”. In: *Journal of the American society for information science and technology* 58.7 (2007), pp. 1019–1031.

[33] Linyuan Lü, Duanbing Chen, Xiao-Long Ren, Qian-Ming Zhang, Yi-Cheng Zhang, and Tao Zhou. “Vital nodes identification in complex networks”. In: *Physics Reports* 650 (2016), pp. 1–63.

[34] Linyuan Lü and Tao Zhou. “Link prediction in complex networks: A survey”. In: *Physica A: statistical mechanics and its applications* 390.6 (2011), pp. 1150–1170.

[35] Abdul Majeed and Ibtisam Rauf. “Graph Theory: A Comprehensive Survey about Graph Theory Applications in Computer Science and Social Networks”. In: *Inventions* 5.1 (2020), p. 10.

[36] Sebastian Mežnar, Nada Lavrač, and Blaž Škrlj. “Transfer Learning for Node Regression Applied to Spreading Prediction”. In: *arXiv preprint arXiv:2104.00088* (2021).

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[38] T. Opsahl and P. Panzarasa. “Clustering in weighted networks”. In: *Social networks* 31.2 (2009), pp. 155–163.

[39] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. “Epidemic processes in complex networks”. In: *Reviews of modern physics* 87.3 (2015), p. 925.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[41] Sen Pei and Hernán A Makse. “Spreading dynamics in complex networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2013.12 (2013), P12002.

[42] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “Deepwalk: Online learning of social representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[43] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. “Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec”. In: *Proceedings of the eleventh ACM international conference on web search and data mining*. 2018, pp. 459–467.

[44] Bo Qu and Huijuan Wang. “SIS epidemic spreading with heterogeneous infection rates”. In: *IEEE Transactions on Network Science and Engineering* 4.3 (2017), pp. 177–186.

[45] Rita P Ribeiro and Nuno Moniz. “Imbalanced regression and extreme value prediction”. In: *Machine Learning* 109.9 (2020), pp. 1803–1835.

[46] Francisco A Rodrigues, Thomas Peron, Colm Connaughton, Jurgen Kurths, and Yamir Moreno. “A machine learning approach to predicting dynamical observables from network structure”. In: *arXiv preprint arXiv:1910.00544* (2019).

[47] Ryan A. Rossi and Nesreen K. Ahmed. “The Network Data Repository with Interactive Graph Analytics and Visualization”. In: *AAAI*. 2015. URL: `http://networkrepository.com`.

[48] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. “GEMSEC: Graph Embedding with Self Clustering”. In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2019*. ACM. 2019, pp. 65–72.

[49] Bernhard Scholkopf, Peter L Bartlett, Alex J Smola, and Robert Williamson. “Shrinking the tube: a new support vector regression algorithm”. In: *Advances in neural information processing systems* (1999), pp. 330–336.

[50] Panpan Shu, Wei Wang, Ming Tang, and Younghae Do. “Numerical identification of epidemic thresholds for susceptible-infected-recovered model on finite-size networks”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.6 (2015), p. 063104.

[51] Alex J Smola and Bernhard Schölkopf. “A tutorial on support vector regression”. In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[52] N. Spring, R. Mahajan, and D. Wetherall. “Measuring ISP topologies with Rocketfuel”. In: *SIGCOMM*. Vol. 32. 4. 2002, pp. 133–145.

[53] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. “Line: Large-scale information network embedding”. In: *Proceedings of the 24th international conference on world wide web*. 2015, pp. 1067–1077.

[54] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Academic Press, 2009.

[55] Luis Torgo and Rita Ribeiro. “Precision and recall for regression”. In: *International Conference on Discovery Science*. Springer. 2009, pp. 332–346.

[56] Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. “Smote for regression”. In: *Portuguese conference on artificial intelligence*. Springer. 2013, pp. 378–389.

[57] Maddalena Torricelli, Márton Karsai, and Laetitia Gauvin. “weg2vec: Event embedding for temporal networks”. In: *Scientific reports* 10.1 (2020), pp. 1–11.

[58] Piet Van Mieghem. *Graph spectra for complex networks*. Cambridge University Press, 2010.

[59] Daixin Wang, Peng Cui, and Wenwu Zhu. “Structural deep network embedding”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 1225–1234.

[60] Huijuan Wang, Qian Li, Gregorio D'Agostino, Shlomo Havlin, H Eugene Stanley, and Piet Van Mieghem. "Effect of the interconnected network structure on the epidemic threshold". In: *Physical Review E* 88.2 (2013), p. 022801.

[61] Xiangrong Wang, Yakup Koç, Robert E Kooij, and Piet Van Mieghem. "A network approach for power grid robustness against cascading failures". In: *2015 7th international workshop on reliable networks design and modeling (RNDM)*. IEEE. 2015, pp. 208–214.

[62] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. "Graph embedding and extensions: A general framework for dimensionality reduction". In: *IEEE transactions on pattern analysis and machine intelligence* 29.1 (2006), pp. 40–51.

[63] Xiu-Xiu Zhan, Ziyu Li, Naoki Masuda, Petter Holme, and Huijuan Wang. "Susceptible-infected-spreading-based network embedding in static and temporal networks". In: *EPJ Data Science* 9.1 (2020), p. 30.

[64] Zi-Ke Zhang, Chuang Liu, Xiu-Xiu Zhan, Xin Lu, Chu-Xu Zhang, and Yi-Cheng Zhang. "Dynamics of information diffusion and its applications on complex networks". In: *Physics Reports* 651 (2016), pp. 1–34.

[65] Gouheng Zhao, Peng Jia, Cheng Huang, Anmin Zhou, and Yong Fang. "A machine learning based framework for identifying influential nodes in complex networks". In: *IEEE Access* 8 (2020), pp. 65462–65471.