ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3717

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : *Laboratoire de Recherche en Informatique et Télécommunications*
Discipline : *Sciences de l'ingénieur*
Spécialité : *Informatique et Télécommunications*

Présentée et soutenue le 02/12/2022 par :

JIBOUNI AYOUB

Link prediction using local structures, machine learning classifiers And node centralities in large scale complex networks

JURY

Moulay Driss RAHMANI	PES, Université Mohammed V de Rabat, Faculté des Sciences.	Président
AIT ABDELOUAHED Abdelkaher	PH, Université Chouaib Doukkali El Jadida, Faculté des Sciences.	Rapporteur/ Examinateur
DRISSI MALIANI Ahmed	PH, Université Mohammed V de Rabat, Faculté des Sciences.	Rapporteur/ Examinateur
Lahoucine BALLIHI	PH, Université Mohammed V de Rabat, Faculté des Sciences.	Rapporteur/ Examinateur
Yahya BENKAOUZ	PH, Université Mohammed V de Rabat, Faculté des Sciences.	Examinateur
BARA SAMIR	PA, Université Sultan Moulay Slimane, Ecole Supérieure de Technologie Khénifra.	Invité
LOTFI Dounia	PH, Université Mohammed V de Rabat, Faculté des Sciences.	Co-encadrant
HAMMOUCH Ahmed	PES, Université Mohammed V de Rabat, École Nationale supérieure d'Arts et Métiers.	Directeur de Thèse

Année Universitaire : 2021-2022

I would like to dedicate my thesis
to my beloved parents,
my sister, my wife and JOUD.

Acknowledgements

This thesis has been performed in **the Laboratory of Research in Computer Science and Telecommunication (LRIT)** of the Faculty of Sciences Rabat, Mohammed V University, under the supervision of **Mr.Ahmed HAMMOUCH**, Full Professor at ENSEM Rabat and the co-supervision of **Mme.Dounia LOTFI**, Qualified Professor at Faculty of sciences Rabat .

First of all, I would like to express my gratitude to my supervisor, **Mr.Ahmed HAMMOUCH**, who guided me throughout my studies and I would like to mention that it was a privilege and honour for me to have been his student. I am thankful to him for giving me a strong background to pursue my research in this field, for having fruitful discussions with him, as well as for his strong support and dedication during these years.

I would like to warmly thank Professor **Mme.Dounia LOTFI**, for her co-supervision, patience, and motivation. I want to thank her especially for letting me wide autonomy while providing appropriate advice, for having fruitful discussions and for her support. I hope to keep up our collaboration in the future.

I would like to convey my greatest honor and gratitude to **Mr.MOULAY DRISS RAHMANI**, Full Professor at Faculty of Sciences Rabat, for agreeing to chair the committee of my thesis.

I would also like to thank **Mr.AIT ABDELOUAHED ABDELKAHER**, Qualified Professor at Faculty of Sciences El Jadida, for his kind willingness to attend and judge this thesis.

I would like also to thank **Mr.DRISSI MALIANI AHMED**, Qualified Professor at Faculty of Sciences Rabat, for having agreed to report this thesis and for having taken time to thoroughly evaluate this work.

I am very fortunate and grateful to **Mr.LAHOUCIN BALIHI**, Qualified Professor at Faculty of Sciences Rabat, for having agreed to report this work and for his valuable comments that helped me to improve this manuscript.

I would like to thank **Mr.YAHYA BENKAOUZ**, Qualified Professor at Faculty of Sciences Rabat, for having agreed to report also this work and to participate in the committee. I sincerely thank him for his availability and his precious remarks in order to ameliorate my modest work.

I would also like to extend my warmest thanks to **Mr.BARA SAMIR**, Assistant Professor at Superior School of Technology Khenifra, for having taken the time to thoroughly evaluate this work.
I would like to express here my gratitudes to every person who gave me support and contributed to the elaboration of this work. Finally, my warmest thanks go to my lovely parents, sister, and wife for their endless support and encouragement throughout my entire life.

Abstract

The term "complex network" has entered modern usage, and it now refers to circles of friends, acquaintances, and any other type of relationship between entities. Due to their enormous structure and dynamic behavior, many real-world systems are modeled as complex networks. Graph theory provides practical tools for comprehending and analyzing their process and evolution. Many structural properties are used, such as a neighbor, paths, shortest paths, and node degree. Based on those features, we have proposed metrics to predict the upcoming links in a network. This problem is known as link prediction, where we predict the upcoming links at time t+1 from a known graph at time t. Metrics such as Jaccard, Common neighbors, and Local paths, have been proposed to solve the link prediction problem. However, it is still an active zone where the existing metrics lack accuracy or can not deal with large-scale networks. In this thesis, our primary motivation is to propose new metrics and models that provide accurate predictions in reasonable execution time. Then, we evaluate the proposed methods against state-of-the-art metrics using Area Under Curve. In addition, we use machine learning models to solve the link prediction problem as a classification task. All the proposed methods show good performance in different types of real-world datasets.

Keywords: Complex Networks, Graph theory, Social Network Analysis, Social Recommendation, Similarity Measures, Area Under Curve, Machine Learning, Regression Models, Network Behaviors, Link Prediction.

Résumé

Le terme “réseaux complexes” est utilisé pour décrire l'ensemble des relations d'amitié, connaissances, ou autres types de relations entre des entités. Ces réseaux se caractérisent par leurs évolutions en termes d'utilisateurs et en termes de relations crées entre ces derniers. La théorie de graphe donne des outils efficaces pour étudier et analyser ces réseaux. Beaucoup de notions comme la notion d'amis, chemins, le plus court chemin, degré d'un noeud, et autres sont d'origine de la théorie de graphe. Des exemples de méthodes et de mesures comme Jaccard, Adamic/Adar, Katz sont basées sur ces notions. L'un des problèmes les plus importants dans l'analyse des réseaux sociaux est la prédiction des liens. Dans ce problème, on cherche à trouver les liens qui se produisent dans le temps t+1 en se basant sur les liens disponibles au temps t. Pour résoudre ce problème, plusieurs métriques ont été développées comme le nombre d'amis en communs, le plus court chemin, coefficient de Jaccard et autres. Le but de cette thèse est de proposer de nouvelles mesures qui dépassent la précision des méthodes de l'état de l'art dans un temps d'exécution raisonnable. Nous avons également utilisé des méthodes d'apprentissage automatique pour résoudre le problème de prédiction de liens. Finalement, une variété d'expérimentations a montré l'efficacité et la précision de toutes les méthodes proposées des réseaux réels.

Mots clés: Réseaux complexes, théorie des graphes, analyse des réseaux sociaux, recommandation sociale, mesures de similarité, AUC, apprentissage automatique, modèles de régression, comportements des réseaux, prédiction des liens.

Résumé Détaillé

Les applications des réseaux sociaux sont multiples et leur impact dans notre vie actuelle est nettement visible. Ces réseaux jouent un rôle trés important dans notre vie, ils permettent d'une part, la communication entre tous les individus qui les constituent et d'autres part, la circulation des informations et des événements instantanément. Ce qui les rend trés dynamiques et favorise d'avantage leur évolution.

En effet, l'évolution des réseaux sociaux est fondée sur deux piliers : Le premier est l'ajout de nouveaux noeuds (utilisateurs dans le cas des réseaux sociaux). Le deuxième est l'ajout de nouveaux liens entre les noeuds déja existants. Cet ajout de liens entre utilisateurs pose une question fondamentale : "comment peut-on savoir les liens qui vont être ajoutés dans le futur ?", ou plus généralement étant donné un graphe $G = (V, E)$ dans le temps t avec V sa liste des noeuds et E la liste des liens entre ces noeuds, peut-on prédire les liens qui vont être ajoutés dans le temps $t+1$. Cette question, qui constitue l'objectif cette thèse, nécessite une étude approfondie et lie plusieurs sciences comme la théorie des graphes et la sociologie.

La prédiction des liens est l'un des problèmes les plus connus dans l'analyse des réseaux sociaux. Cette thèse s'étale sur cinq chapitres.

1. **Chapitre 1: Analyse des réseaux complexes.** Dans ce chapitre, nous présentons l'histoire des réseaux complexes, leurs applications, leurs types puis nous introduisons les concepts généraux relatives aux noeuds comme le degré d'un noeud, betweenness d'un noeud etc. Nous avons aussi introduit des mesures relatifs aux graphes comme le coefficient d'assortativité, coefficient de clustering, etc. Les mesures présentées dans ce chapitre sont appliquées dans les chapitres 3,4 et 5 sur des graphes réels.
2. **Chapitre 2: Les méthodes de prédiction des liens.** Dans ce chapitre, nous

avons introduit le problème de prédiction des liens dans les réseaux complexes, leurs applications, les mesures de similarité basée sur le voisinage, puis nous avons présenté les mesures globales les plus utilisées, nous avons introduit quelques mesures semi-locales, par la suite nous avons comparé les inconvénients des mesures existantes. Nous avons aussi définit la metrique d'évaluation qui nous permet de valider les résultats.

3. **Chapitre 3: Méthode de prédiction de lien basée sur la longueur du chemin entre une paire de noeuds non liés et leur degré.** Dans ce chapitre, nous avons introduit les inconvénients des mesures présentées dans le chapitre précédent, puis nous avons introduit une nouvelle mesure basée sur l'ensemble des chemins entre deux noeuds. Cette mesure donne des résultats prometteux par rapport aux indices de l'état de l'art. Pour tester la précision de notre mesure nous l'avons appliqué sur cinq bases de données. Notre mesure est paramétrable, nous avons étudié l'impact de la longueur de chemins sur la précision de la prédiction. Par la suite, nous avons utilisé les méthodes d'apprentissage automatique pour résoudre le problème de prédiction des liens.

4. **Chapitre 4: Les ressources moyennes reçues et les algorithmes d'apprentissage automatique pour améliorer les méthodes de prédiction de liens.** Dans ce chapitre, nous avons introduit une nouvelle mesure de prédiction des liens basée sur les ressources moyennes reçues depuis un noeud source, puis évaluer ses performances par rapport aux métriques existantes. Nous avons étudié la relation entre le paramètre utilisé par cette mesure et les caractéristiques des réseaux. Par la suite, nous avons utilisé notre méthode en combinaison avec les méthodes de l'état de l'art pour résoudre le problème de prédiction des liens.

5. **Chapitre 5: Prédiction de liens en utilisant "betweenness centrality" et le réseau de neurones.** Dans ce chapitre, nous avons introduit une nouvelle mesure basée sur la centralité de "betweenness" et les métriques de similarité locales, ce qui permet d'améliorer leurs précision. Nous avons appliqué les mesures sur cinq réseaux réels. Finalement, nous avons combiné nos mesures avec les mesures de l'état de l'art comme entrée pour le réseau de neurones. Les résultats prouvent que la méthode suivie augmente la précision et diminue le taux d'erreur.

Acronyms

SNA:Social Network Analysis
LP: Link Prediction
CN: Common Neighbors
Sa: Salton
JA: Jaccard
PD: Parameter Dependent
Sor: Sorensen
HP: Hub Promoted Index
HD: Hub Depressed Index
HN: Leicht-Holme-Newman index
AA: Adamic-Adar
RA: Resource Allocation
PA: Preferential Attachment
LPI:Local Paht index
SH: Common Neighbors Soundarjan and Hopcroft
WIC:Within-cluster inter-cluster index
MRR:Mean Received Ressources
CCPA:Common neighbor and Centrality based Parametrized Algorithm
PSI:Proposed Similarity Index
SVM:Support Vector Machine
LM:Local Mertrics
LSBC:Local Similarity Based Metrics
RELU: Rectified Linear Unit

List of Figures

List of Tables

List of publications

- Ayoub, J., Lotfi, D. & Hammouch, A. Link prediction using betweenness centrality and graph neural networks. Soc. Netw. Anal. Min. 13, 5 (2023). https://doi.org/10.1007/s13278-022-00999-1.

- Ayoub J, Lotfi D, Hammouch A. Mean Received Resources Meet Machine Learning Algorithms to Improve Link Prediction Methods. Information. 2022; 13(1):35. https://doi.org/10.3390/info13010035.

- Ayoub, J., Lotfi, D., El Marraki, M. et al. Accurate link prediction method based on path length between a pair of unlinked nodes and their degree. Soc. Netw. Anal. Min. 10, 9 (2020). https://doi.org/10.1007/s13278-019-0618-2.

- A. JIBOUNI, D. LOTFI, M. E. MARRAKI and A. HAMMOUCH, "A novel parameter free approach for link prediction," 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), 2018, pp. 1-6, doi: 10.1109/WINCOM.2018.8629586.

Contents

General Introduction

The study of complex networks, which are omnipresent in nature and society, has lately emerged as a significant subject of research in the multidisciplinary field of research that covers: physics, mathematics, computer science, sociology, biology, and economics. The elements of many real systems in these various scientific domains can be represented as complex networks. The edges represent relationships between the elements, and the nodes represent the elements themselves. These complex networks do not follow any pattern in their construction. Also, they are dynamically evolving in time, with thousands or millions of nodes and edges added and deleted. The World Wide Web(WWW), the Internet, social networks like Facebook, Instagram, transportation systems, communication networks, genetic networks, brain networks, and others are some well-known examples of complex networks. Regardless of their origins, those networks share a common property: the increase in the number of edges over time.

Graph theory provides mathematical tools for modeling and analyzing complex networks. Euler, in 1735 introduced the problem of the Konigsburg bridges [Euler, 1956]; his solution is considered the first proof in graph theory. The last two decades have witnessed a huge interest in complex network analysis, also known as social network analysis (SNA). In this field of research, scientists try to solve several problems, such as community detection Fortunato [2010], network proper-

ties Boccaletti et al. [2006] and link prediction Lü and Zhou [2011], Kumar et al. [2020], Liben-Nowell and Kleinberg [2007]. The problem of link prediction in complex networks is known to be one of the most critical tasks of SNA. This problem has a wide area of applications, such as terrorist detection [Al Hasan et al., 2006] and fraud detection, it can be applied to detect communities, but its fundamental use is in product recommendation, E-commerce, and social networks. Solving this problem gives us a clear idea about the future interactions or choices that the users will have.

Because of its wide range of applications in various fields, the link prediction problem has received considerable attention from the scientific community. Many scientists propose to compute the similarity between users using metrics, such as common neighbor metric Newman [2001], Jaccard metric [Jaccard, 1901], preferential attachment metric Barabâsi et al. [2002] and so on, theses metrics relay on local information of the users like the set of their acquaintances. On the other hand, other scientists use information from all the graph [Katz, 1953] or a part of the graph such as local path [Lü et al., 2009]. However, these metrics are not efficient for real-world networks having a significant number of nodes and links because their computational complexity is very high. Therefore, most recent works have shown that the best way to solve this problem is using local metrics that provide low computational complexity and high accuracy.

The ultimate purpose of this dissertation is to investigate the link prediction problem by analyzing current metrics. Then, propose new metrics that are highly accurate. Finally, test the proposed metrics on real-world datasets from various fields.

Thesis Contributions and Outline

This thesis's novelty is to study the link prediction problem, then propose simple metrics that can provide accurate predictions in reasonable execution time. Afterward, we propose new metrics that can enhance the performance of the state-of-the-art metrics. Finally, we use machine learning and deep learning techniques to solve the link prediction problem as a classification task.

This thesis comprises five chapters: The first chapter introduces the basic concepts of graph theory and social network analysis. While chapter 2 studies the problem of link prediction and provides state of the art. Chapters 3, 4, and 5 present our main contributions to solving this problem using three different similarity metrics that outperform the existing link prediction metrics (chapter 2).

Chapter 1: Complex network analysis. In this chapter, we introduce the basic network structures. Then, we investigate the most common graph representations. We also used examples to clarify node influence and network definition. This chapter introduces the fundamental concepts of social network analysis. The majority of those ideas will be applied later.

Chapter 2: Link prediction in complex networks. In this chapter, we introduce the background and applications of link prediction. Then, we depict the different methods and techniques known in the literature to solve the link prediction problem in complex networks, apply these metrics on example datasets, and class the similarity-based methods. Then, we show the performance of these metrics in several real-world networks. In addition, we study the link prediction problem using machine learning and deep learning techniques. Finally, we introduce some other approaches.

Chapter 3: Accurate link prediction method based on path length between a pair of unlinked nodes and their degree. In this chapter, we proposed a new similarity index that relies on the number of paths between the source node and the destination node. Then, we apply our proposed metric to real-world datasets from different fields; the results show a considerable improvement (that reaches 10%) in AUC. Moreover, we study the influence of the path length on the algorithm performance. Then, we use the state-of-the-art metrics as features of classifiers of machine learning algorithms and compare the performance of those algorithms in both cases, with our metric as an additional feature and without our metric. Finally, to show the impact of our metric in the classification task, we have shown its importance for the random forest algorithm.

Chapter 4: Mean received resources meet machine learning algorithms to improve link prediction methods. This chapter presents a new parametrized metric for link prediction in complex networks. We based our measure on both mean received resources (MRR) and a local similarity measure (LM) from the state of the art. Besides, this metric is parametrized. As a result, the system/user can adjust the importance of the factor under consideration. The results show that the improved metrics have higher accuracy than the original metrics.

Chapter 5: Link prediction using betweenness centrality and graph neural networks. In this chapter, we present a novel method for improving state-of-the-art local similarity metrics by exploiting the betweenness centrality of unlinked nodes. The proposed metric is parameterized to control the local similarity measure's contribution. We tested the proposed metric method on eight datasets derived from real-world networks. Furthermore, we investigated the effect of the dumping

parameter on the AUC results and identified the best one for each dataset. We also demonstrate that all improved versions of local similarity metrics achieve a higher AUC value. Finally, we apply machine learning to classify links.

Chapter

1 Complex network analysis

1.1 Introduction

The term "complex network" has entered modern usage, and it now refers to circles of friends, acquaintances, and any other type of relationships between entities. Complex network analysis relies on a solid mathematical basis that enables researchers to represent, characterize and measure network features.

In this chapter, we introduce the history of complex networks. Then, we give some examples of real-world networks. We also define the network structure basics and the graph's representation and properties. Finally, we introduce the network description.

1.2 History of complex networks

According to Alexanderson [2006], Li [2014] **Graph theory** began in 1735, when Leonhard Euler [Euler, 1956] presented his solution to the Konigsberg bridge issue (see figure 1.1). The topic is historically significant because it demonstrates the power of graph theory by reducing complex real-world networks to a basic model. The close relationship between network structures and properties is usually the focus of graph theory research. Paul Erdos and Alfred Renyi, two Hungarian

mathematicians, successfully linked the concepts of graph theory with probability theory in the 1960s, establishing a new branch of graph theory called random graph theory(refer to Erdős et al. [1960]).

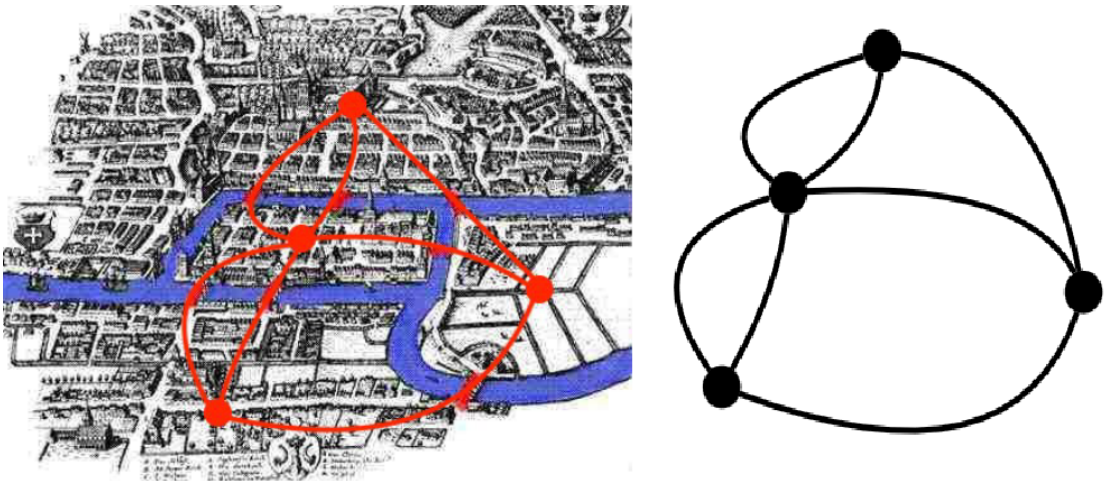


Figure 1.1: The real location of the seven bridges is depicted on a map of Konigsberg in Euler's time and a graph representation, with the river Pregel and the bridges being highlighted. Keep in mind that the challenge was to determine whether a path could be followed that traverses each bridge precisely once.

Euler introduced the Konigsberg Bridge problem in the first two paragraphs of his proof [Euler, 1956]. Euler states that he believes this problem is about geometry not founded on measurements and calculations and requires a new type of geometry, which Leibniz referred to as Geometry of Position. Euler sketched out the problem (see Figure 1) and labeled the seven different bridges as a, b, c, d, e, f, and g. In another paragraph, he poses the problem's general question: "Can one determine whether or not it is possible to cross each bridge exactly once?". At the end of his work, Euler found that:

- No route meeting the requirements can be discovered if an odd number of bridges connects more than two areas.

- The required travel can be completed if there are precisely two nodes with an odd number of edges. The travel should start in one of the two nodes.

- The necessary travel can be completed no matter where it starts if, at last, there is no area with an odd number of approach bridges. These guidelines resolve the issue that was previously raised.

Researchers have considered complex networks a helpful tool for representing real-world systems in nature and society in the last century. Some of the most important examples are the Internet, which results from the interconnection of routers and computers; the cell is a network of substances linked by chemical processes; and the www network. We can find these networks in various domains: biology, sociology, psychology, and computer science. The four main categories of complex networks are social, information, technology, and biological.

Definition 1.2.1. ***Social Network (SN)*** *[Cambridge, 2021] "is a website or computer program that allows people to communicate and share information on the internet using a computer or mobile phone." In this category of networks, nodes may represent groups of people, such as companies, students, and musicians (see Figure 1.2). Moreover, edges describe the communications between them. Social networks have referred to websites like Facebook and MySpace in the last decade.*

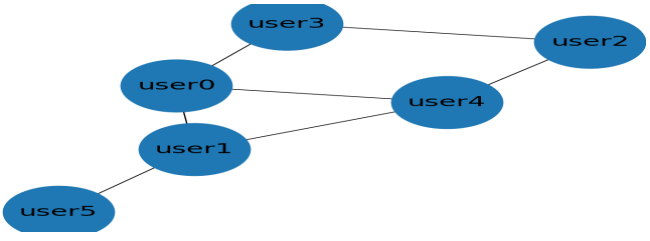


Figure 1.2: Example of social network

Definition 1.2.2. *McGraw-Hill [2003] has defined the* ***information network*** *as a service that offers a broad range of information services to users on a dial-up basis also known as a membership database. The giant information graph is the web (see Figure 1.3), where pages contain information and link to other pages by hyperlinks.*

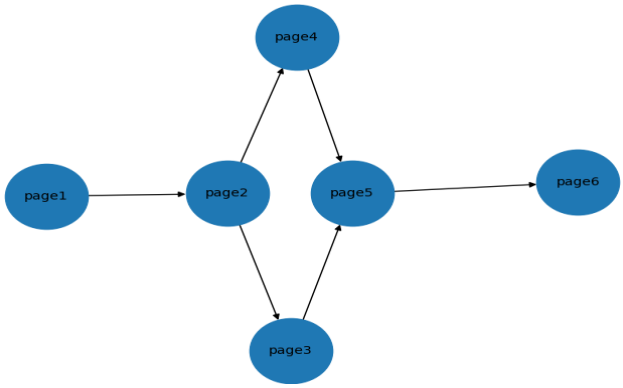


Figure 1.3: Example of information networks

Definition 1.2.3. *A* ***technological network*** *is the physical infrastructure networks (see Figure 1.4). Internet, the global network of electrical, optical, and wireless data connections that connect computers and other information systems, is the most well-known network. The power grid graph is a structure of high-voltage three-phase power lines that spans a country or a section of a country, and it constitutes one of the essential networks that we will use in the next chapters.*

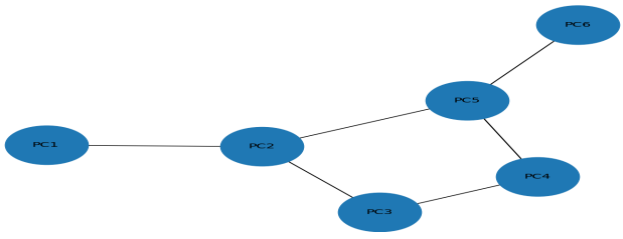


Figure 1.4: Example of technological networks

Definition 1.2.4. *Graphs can be used to model and evaluate* ***complex biological systems*** *(see Figure 1.5). For example, ecosystems can be described as networks of interaction between proteins of amino acids. A network of connected atoms, such as carbon, nitrogen, and oxygen, can also be represented as amino acids. A network's main components are nodes and edges. Edges represent the connection between nodes, while nodes represent the units in the network.*

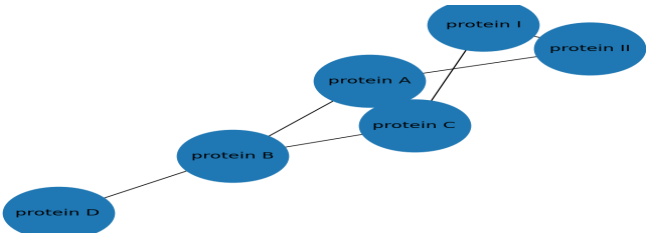


Figure 1.5: Example of biological networks

All these network categories were the subject of thousands of papers and numerous popular scientific books published on the modern theory of complex networks since the early 2000s. The construction and operation of complex networks, the modeling of real-world networks, and several network dynamic processes were all treated in these volumes [Barabasi, 2003, Dorogovtsev et al., 2003, Cohen and Havlin, 2010, Van Mieghem, 2010]. Several metrics were used to study and analyze complex networks. The number of nodes, the number of edges, the size of the giant component, the efficiency of the network [Latora and Marchiori, 2001] , the clustering coefficient [Watts and Strogatz, 1998], the assortative coefficient [Newman, 2002], the diameter of the network, the degree of heterogeneity [Snijders, 1981] are denoted by N, M, Nc, E, C, r, D and H respectively. These measurements help to understand the structure of complex networks and provide crucial insights into their dynamical and functional behavior. Furthermore, network science has been employed in a broader spectrum of challenges, leading us to investigate network

structure basics.

1.3 Basics of network structure

Graph theory provides essential tools to study and analyze complex network structures. A person can be represented as a node or vertex in a social network, and links or edges represent the relationships between nodes. For the rest of this thesis, we will use node or vertex interchangeably, as well as link, edge or tie. For example, a graph or network G is a set of nodes V and edges E, and we write $G(V, E)$. In this part, we will introduce the most commonly utilized types of networks.

1.3.1 Simple graphs

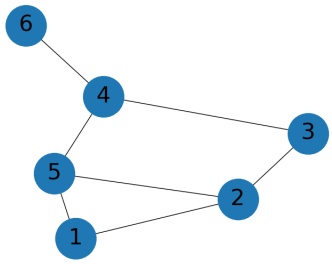


Figure 1.6: Example of simple graph

Definition 1.3.1. *A simple graph is an undirected, unweighted graph containing no loops or multiple edges (see Fig 1.6) (Note that in this thesis, we use only simple graphs). For instance, if node A links to node B, then the relation is considered in both directions.*

Example 1.3.1. *The most famous example of simple graphs is Facebook [mark zuckerberg], where the friendship relation is created once the invitation is accepted.*

1.3.2 Labled graphs

Definition 1.3.2. *In a labeled graph, it is possible to designate the edges. The label expresses something about the two people's relationship. It could be the name of the connection (for example, brother, mother, or father) or some other information about it.*

Example 1.3.2. *The labels in Figure 1.7 denote the type of relationship between a pair of nodes. For instance, nodes 2 and 3 are friends, and nodes 1 and 3 are neighbors.*

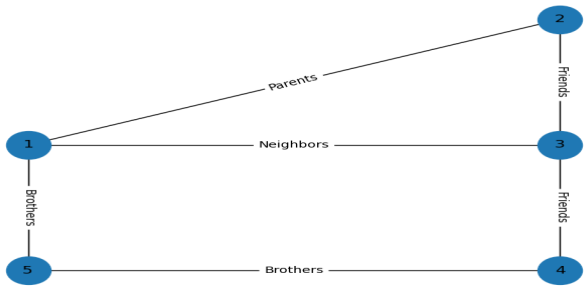


Figure 1.7: Example of labled graph

1.3.3 Weighted graphs

Definition 1.3.3. *In this type of graph, edges can be valued or weighted. The weight is a numerical value that represents numerical data about a relationship. This weight is frequently a relationship's strength, although it can come from various sources and implies multiple results. Weights can be displayed as numeric labels on the links.*

Example 1.3.3. *In Figure 1.8, the relation strength between nodes 1 and 3 is equal to 9.8, which is the heaviest relation in the network.*

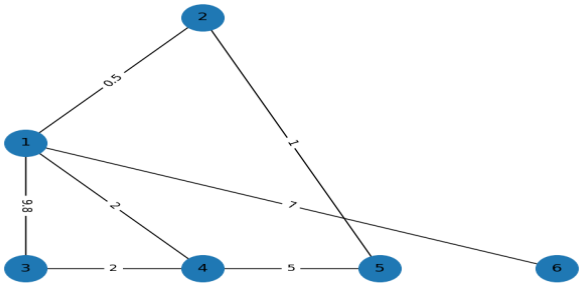


Figure 1.8: Example of weighted graph

1.3.4 Directed and undirected graphs

Definition 1.3.4. *An undirected edge denotes a tie between two nodes that is reciprocated. A directed edge defines a relationship between two nodes that is not necessarily reciprocated. The edge type determines whether the network is directed or undirected.*

Example 1.3.4. *For example, Instagram is a directed network where: Person 1 may follow Person 7 without being followed in return (see Figure 1.9).*

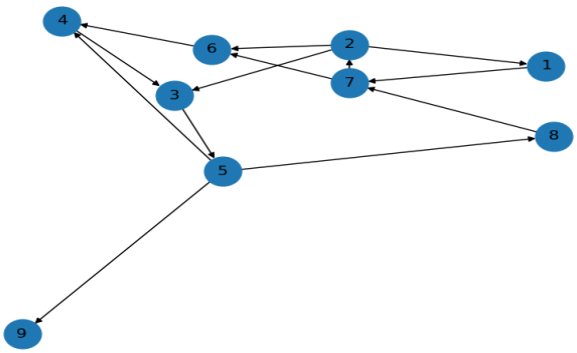


Figure 1.9: Example of directed graph

1.3.5 The six degrees of separation

Human society has a small-world aspect, according to the six degrees of separation [Milgram, 1967], where the author Stanley Milgram sent packages (300 packages) from the west coast of the US to a stockbroker in Boston. The experiment shows that the average paths of length 6 were enough to reach the stockbroker (about 64 packages reach the person in Boston). Another experiment was performed by some minor societies that had also conducted small-world experiments to check the accuracy. Casper Goffman defined the Erdos number, which describes the collaborative distance between a mathematician and Paul Erdos. With an average Erdos number of 4.65, practically everyone has an Erdos number of less than 8; that is to say, everyone can reach Erdos with a path of length less than 8 (we will discuss the concept of the path later in section 1.5). Erdos number is proved for even graphs of millions or billions of nodes and edges; for instance, a Facebook graph has a diameter of 8.

Definition 1.3.5. *A small-world network is a mathematical structure in which most nodes or users are not immediate neighbors with one another, but where any given node's neighbors are likely to be neighbors as well, and with only a few hops or steps may reach most nodes.*

We can describe a small-world network as a graph in which the typical distance L (the number of steps required) between two randomly selected nodes increases as a function of the network's logarithm, or in other words, $L \propto \log N$, whereas the global clustering coefficient is not insignificant.

1.4 Graph representation

In the previous section, networks are shown as graphs where nodes are depicted as circles and edges as lines connecting them. We can express networks in a variety

of ways. The adjacency list, adjacency matrix, and XML format are the most utilized representations of a network.

1.4.1 Adjacency list

An adjacency list, also known as an edge list, is one of the most basic and often used network representations. A list of connected nodes represents each network edge. We consider the graphs from the previous section and then provide their adjacency lists.
The first graph was simple (see Figure 1.6), where no loops nor weights are allowed. The edge list of this network is: $[(1,2),(1,5),(2,5),(2,3),(3,4),(4,5),(4,6)]$.

Remark 1.4.1. *Note that every node pair represents a link; for instance,* $(1,2)$ *is the link between nodes 1 and 2. The separation character may differ from one dataset to another, like space, tabulations, or even semicolons. We have to mention that the order of pair nodes has no impact on the final simple graph.*

In the case of weighted and labled graphs, an attribute is added, the weight or the label. Let's consider the graph in Figure 1.8 the edge list of the graph is: $[(1,2,0.5),(1,3,3),(1,4,2),(5,2,1),(4,5,5),(1,3,9.8),(4,3,2),(1,6,7)]$.

Remark 1.4.2. *Note that the order of edges has no effect on the final weighted graph.*

Next, we will consider a directed graph (see Figure 1.9). In the directed graph the edge list is:
$[(1,7),(7,2),(7,6),(2,1),(2,3),(2,6),(3,5),(6,4),(5,4),(5,8),(5,9),(4,3),(8,7)]$. In this graph, the order of the pair of nodes determines the edge direction; for instance, The edge from node 7 to node 1 differs from the edge from node 1 to node 7.

1.4.2 Adjacency matrix

Definition 1.4.1. *The adjacency matrix of a simple graph G (see Figure 1.6) is a square matrix (see matrix 1.2) of 0 and 1 that lists all the nodes in both row and columns, where 1 represents the relation between the nodes and 0 denotes no relation between the two nodes. The equation 1.1 defines the adjacency matrix of a simple graph:*

$$(A_{ij}) = \begin{cases} a_{i,j} = 1 & if\ (i,j) \in E \\ a_{i,j} = 0 & else \end{cases} \tag{1.1}$$

For instance, the first column represents node one, and the second column represents node 2. The same thing applies to the rows; if we consider node 1: this node has a relation with nodes 2 and 5, so the first column has two ones in the 2nd and fifth rows. Note that this matrix is symmetric (see matrix 1.2).

$$A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{1.2}$$

Definition 1.4.2. *In the case of weighted graph $a_{i,j}$ is equal to the weight of each edge, so the equation will be:*

$$(A_{ij}) = \begin{cases} a_{i,j} = w_{i,j} & if\ (i,j,w) \in E \\ a_{i,j} = 0 & else \end{cases} \tag{1.3}$$

If we consider the Figure 1.8, node 2 has two relations: the first is with node 1 (edge weight = 0.5) and the second with node 5 (edge weight= 1). Note that this matrix is symmetric.

$$A_G = \begin{pmatrix} 0 & 0.5 & 9.8 & 2 & 0 & 7 \\ 0.5 & 0 & 0 & 0 & 1 & 0 \\ 9.8 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 & 5 & 0 \\ 0 & 1 & 0 & 5 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.4}$$

In a directed graph (see Figure 1.9), the relationship between nodes is not necessarily mutual, then the adjacency matrix is not symmetric (see matrix 1.5). We apply here the same principle, the value 0 shows no relation, and the value 1 shows a connection between node x and y but not between y and x.(Note that we can extend this network with weights).

$$A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.5}$$

1.4.3 Laplacian matrix

Definition 1.4.3. *Another matrix form of a graph is based on the degree and adjacency of vertices, and it is helpful in social network analysis. Let G be a simple graph. The Laplacian matrix of G is denoted $L_G = D_G - A_G$, where D_G is the diagonal matrix of degrees, such that $deg(i)$ is the degree (number of adjacent links) of vertex i. L_G is defined as:*

$$(l_{ij}) = \begin{cases} deg(i) & if\ i = j \\ -1 & if\ i \neq j\ and\ (i,j) \in E \\ 0 & else \end{cases} \tag{1.6}$$

Let's consider the graph in Figure 1.6, the laplacian matrix is:

$$A_G = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \tag{1.7}$$

1.4.4 XML formats

Sometimes the graph is presented in the XML format. To express a data set in the XML format, we define tags like *node, name, and connection*. The following code 1.1 describes node one relationships that are nodes 2,3, and 4.

Listing 1.1: XML format

```
<node>
        <name>1</name>
        <connection>2</connection>
        <connection>3</connection>
        <connection>4</connection>
</node>
```

1.5 Paths in a graph

Definition 1.5.1. *The notion of the path has been studied in various books and papers [Barnes, 1969, Sedgewick, 2001]. According to [Golbeck, 2013], a path is a series of nodes that information can traverse following edges between them; the number of ties in a path defines its length.*

Let's consider the graph in Figure 1.10:

Table 1.1: Example of paths in a simple graph.

Source node	Destination node	Path	Length
1	6	2, 3, 4, 5	5
1	6	3, 4, 5	4
1	6	4, 5	3
1	6	5	2

However, the most critical paths are the shortest paths. In the case of our example, the shortest path is through node 5 (the path of length 2). In a graph, we can find one or multiple paths if the graph is **connected** (An entire graph is called connected if all pairs of nodes are connected [Golbeck, 2013]).

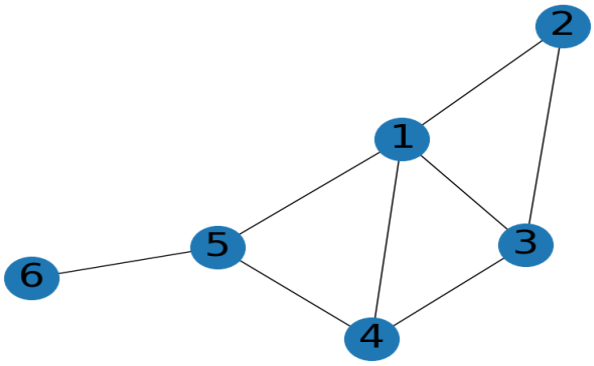


Figure 1.10: Example of paths between two nodes.

1.6 Node centrality

Researchers have proposed several measures to identify the essential and influential individuals in a network, such as a degree centrality, closeness, betweenness, and eigenvector centrality. Those measures also help to understand how networks will grow in the future. Note that we will use some of those measures in this thesis.

1.6.1 Degree centrality

Definition 1.6.1. *According to [Diestel, 2006, Freeman, 1978], the amount of edges connecting to a node determines its degree. The degree of a node in an undirected graph is just the total number of edges incident to it. The degree is measured in two ways in directed graphs: in-degree and out-degree. The amount of edges that enter the node determines the in-degree. In a directed graph, in-degrees are represented by edges with arrows pointing towards the node. The number of edges that starts from a node to adjacent nodes is known as the out-degree. The total degree for a node is calculated by adding the in-degree and out-degree.*

Example 1.6.1. *Considering the previous graph Figure 1.10 the degree of nodes are*

shown in the Table 1.2.

Table 1.2: Example of node degree in a simple graph.

node	*degree*
node 1	*4*
node 2	*2*
node 3	*3*
node 4	*3*
node 5	*3*
node 6	*1*

Example 1.6.2. *For directed graph (see Figure 1.9) the in degrees and out degrees are shown in Table 1.3:*

Table 1.3: Example of node degree in a directed graph.

Node	*In Degree*	*Out Degree*
1	*1*	*1*
7	*2*	*2*
2	*1*	*3*
3	*2*	*1*
6	*2*	*1*
5	*1*	*3*
4	*2*	*1*
8	*1*	*1*

Continued on next page

Table 1.3: Example of node degree in a directed graph. (Continued)

9	*1*	*0*

1.6.2 Closeness centrality

Definition 1.6.2. *Closeness centrality [Bavelas, 1950, Freeman, 1978] describes how close a node is to all the other nodes in the network. In other words, the closeness of a node u is the reciprocal of the average shortest path distance to u across all n-1 reachable nodes. Notice that d(v, u) is the shortest-path distance connecting v and u, and n refers to the number of nodes that can reach u. The closeness centrality is defined as:*

$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)} \tag{1.8}$$

Example 1.6.3. *If we consider the graph in Figure 1.6, the closeness centrality of the nodes are:*

Table 1.4: Example of closeness centrality of nodes in simple graph.

Node	*Closeness centrality*
1	*0.556*
2	*0.625*
5	*0.714*
4	*0.714*
3	*0.625*
6	*0.455*

1.6.3 Betweenness centrality

Definition 1.6.3. *The betweenness centrality can quantify the relevance of a node in a network. We select a pair of nodes and search all the shortest paths between them to compute the betweenness of a node v. Then, we determine what percentage of the shortest paths include node v. The proportion would be* $3 \div 5 = 0.6$ *if there were five shortest pathways between two nodes and three of them traveled through node v. This process is repeated for each pair of nodes in the network. The betweenness centrality of node v is obtained by adding the fractions we computed [Freeman, 1978].*

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{1.9}$$

Where σ_{st} *is the number of shortest paths from node s to node t and* $\sigma_{st}(v)$*, is the amount of those pathways that pass through v (not where v is an endpoint).*

Example 1.6.4. *We consider the graph in Figure 1.10, the betweeness centrality of nodes is given in Table 1.5.*

Table 1.5: The betweeness centrality of nodes in graph Figure 1.10.

Node	*Betweeness centrality*
1	*0.35*
2	*0.0*
3	*0.05*
5	*0.4*
4	*0.1*
6	*0.0*

1.6.4 Eigenvector centrality

Definition 1.6.4. *Eigenvector centrality [Bonacich, 1987] calculates a node's centrality based on its neighbors' centrality. The i-th element of the vector given by the equation is the eigenvector centrality for a node.*

$$Ax = \lambda x \tag{1.10}$$

A is the adjacency matrix of the graph G with eigenvalue λ .

Example 1.6.5. *If we consider the graph in Figure 1.10 the centrality of nodes is given in table 1.6.*

Table 1.6: The eigenvector centrality of nodes in graph Figure 1.10.

Node	*Eigenvector centrality*
1	*0.549*
2	*0.336*
3	*0.453*
5	*0.383*
4	*0.464*
6	*0.128*

1.7 Network description

We can use measurements such as density, degree distribution, connectedness, the average degree of nodes, dimension, clustering coefficient, assortativity coefficient, and degree of heterogeneity to describe a graph.

1.7.1 Average degree of network

Definition 1.7.1. *In a network, the degree of a node is the number of edges connected to it. The average degree of a network is the number of nodes divided by the total sum of their degrees.*

$$Average\ Degree(G) = \frac{number\ of\ edges}{number\ of\ nodes} = \frac{|E|}{|V|} \tag{1.11}$$

Example 1.7.1. *In Figure 1.10 the total number of connections is 8, and the network contains 6 nodes; as a result of that, the average degree of this graph is:* $\frac{8}{6} = 1.33$.

1.7.2 Diameter of a graph

Definition 1.7.2. *The diameter of a graph is simply the maximum of shortest paths or, in other words, "the longest shortest path" between any two vertices. The graph diameter represents the maximum number of vertices that should be traversed from one vertex to another.*

Example 1.7.2. *In Figure 1.10, the diameter of this graph is 3. The shortest paths between the nodes are given in table 1.7:*

Table 1.7: The diameter of the graph in Figure 1.10.

Source node	*Destination node*	*Shortest paths*
1	*2*	*[1,2]*
1	*3*	*[1,3]*
1	*5*	*[1,5]*
1	*4*	*[1,4]*

Continued on next page

Table 1.7: The diameter of the graph in Figure 1.10. (Continued)

1	*6*	*[1,5,6]*
2	*3*	*[2,3]*
2	*5*	*[2,1,5]*
2	*4*	*[2,1,4],[2,3,4]*
2	*6*	*[2,1,5,6]*
3	*5*	*[3,1,5],[3,4,5]*
3	*4*	*[3,4]*
3	*6*	*[3,1,5,6],[3,4,5,6]*
5	*6*	*[5,6]*
4	*5*	*[4,5]*
4	*6*	*[4,5,6]*
6	*1*	*[6,5,1]*
6	*2*	*[[6,5,1,2]]*
6	*3*	*[6,5,1,3],[6,5,4,3]*
6	*5*	*[6,5]*
6	*4*	*[6,5,4]*

From the table 1.7, we can conclude that the maximum length is 3 between nodes 2 and 6, between nodes 3 and 6.

1.7.3 Clustering coefficient

Definition 1.7.3. *The global clustering coefficient [Luce and Perry, 1949] is calculated using node triplets. A triplet is formed by three nodes linked by undirected*

links, either two (open triplet) or three (closed triplet). Consequently, a triangle graph has three closed triplets, one centered on each node (n.b. this means the three triplets in a triangle come from overlapping selections of nodes). The global clustering coefficient is the number of closed triplets (or 3 x triangles) divided by the total number of triplets. This metric expresses the degree to which the network is clustered.

$$C = \frac{3 \times number\ of\ triangles\ in\ the\ graph}{number\ of\ connected\ triplets} \tag{1.12}$$

As an equivalent to the global clustering coefficient is the average of the local clustering coefficients of all the nodes u, note that the clustering coefficient of a node u equals:

$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)} \tag{1.13}$$

where $T(u)$is the number of triangles through node u and $deg(u)$ is the degree of u. The average clustering coefficient is:

$$C = \frac{1}{n}\sum_{v \in G} c_v \tag{1.14}$$

Example 1.7.3. *If we consider the graph in Figure 1.6, the average clustering coficent is:*
Average clustering coefficient$(G) = \frac{1+0.33+0.33}{6} = 0.277$ *because* $c_1 = \frac{2*1}{2*1}$ *and* $c_2 = c_5 = \frac{2*1}{3*2}$ *finaly,* $c_4 = c_3 = c_6 = 0$

1.7.4 Assortativity coefficient

Definition 1.7.4. *According to [Noldus and Van Mieghem, 2015, Newman, 2003], assortativity is measured as a scalar value ranging from -1 to 1. If the nodes with high degrees are connected to nodes with high degrees and nodes with low degrees are connected to nodes with low degrees, then the graph is called assortative. On*

the other hand, if the higher degree nodes are linked to lower degree nodes and low degree nodes are linked to high degree nodes, the graph is disassortative. Assortativity gives information about a network's structure and its dynamic behavior and robustness, such as random or targeted attacks and virus propagation [Chang et al., 2020].

To sum up, assortativity evaluates the similarity of connections in a graph with regard to the node degree. Several applications can easily calculate this metric, like Networkx, which we will use in this thesis to implement all the metrics and measures[Hagberg et al., 2008]. For instance, the assortative coefficient of the graph in Figure 1.6 is -0.4736.

1.7.5 Degree of heterogeneity

Definition 1.7.5. *The degree of heterogeneity as defined in [Li et al., 2018, Zhou et al., 2009] is as follows:*

$$H = \frac{< k^2 >}{< k >^2} \tag{1.15}$$

Where $< k >$ denotes the average degree of the nodes, this measure provides information about the diversity in node degrees and the diversity in the network's structure. Note that new formulas have been proposed [Hu and Wang, 2008, Berahmand et al., 2018].

Example 1.7.4. *If we consider the graph in Figure 1.6 the degree of heterogeneity index is :* $H = 1.102$

1.7.6 Efficiency of network

Definition 1.7.6. *The concept of efficiency of a network [Latora and Marchiori, 2001], defines how efficiently a network exchanges information. The average effi-*

ciency of G can be defined as:

$$e = \frac{\sum_{i \neq j \in G} \epsilon_{ij}}{N(N-1)} = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{i,j}} \tag{1.16}$$

Where the efficiency ϵ_{ij} of the communication between vertex i and vertex j is inversely proportional to the shortest distance: $\epsilon_{ij} = \frac{1}{d_{ij}}$ $\forall i, j$. When there is no path between i and j, $d_{ij} = \infty$ and consistently $\epsilon_{ij} = 0$.

Example 1.7.5. *For the graph in Figure 1.6 the efficiency is: $e = 0.711$*

1.7.7 Density

Definition 1.7.7. *The density of a graph describes how well it is connected. It is a statistic that compares the number of edges in a network to the number of edges that could potentially exist.*

$$density(G) = \frac{number\ of\ edges}{number\ of\ possible\ edges} \tag{1.17}$$

Example 1.7.6. *Let's consider the example of a network in Figure 1.10. The density of the graph is $density(G) = 0.533$ because the number of edges in the graph is 8, and we can find the number of possible edges using the formulas:*
$number\ of\ possible\ edges = \frac{n \times (n-1)}{2}$ where n is the number of nodes in the graph. In this case, $n = 6$.

1.7.8 Degree distribution

Nodes are described by their degree. We can create a degree distribution to understand the degree for all the nodes in the network. This diagram depicts the number of nodes for each degree.

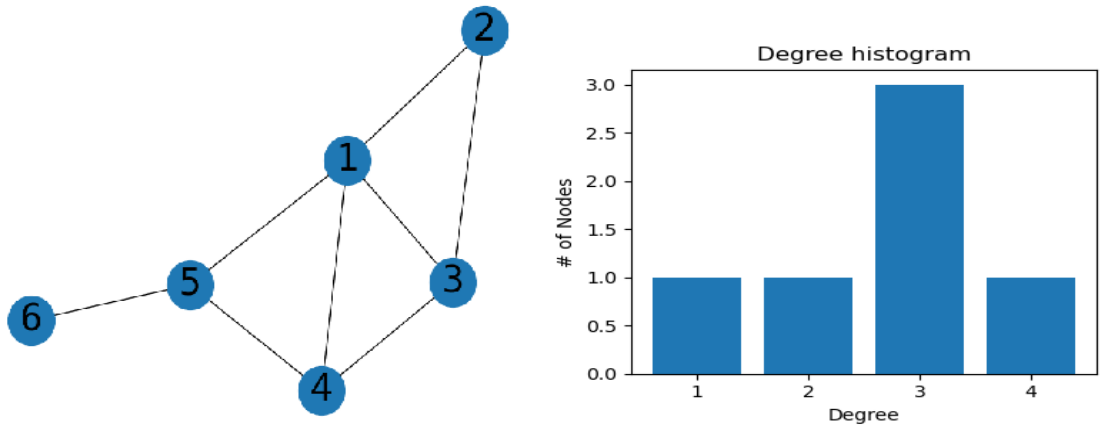


Figure 1.11: Degree distribution of a graph.

From Figure 1.11, we represent the degree in the X-axis, and Y-axis describes the number of nodes with a specific degree. In this Figure, there is only one node of degree 1, which is node 6, and also one node of degree 2, which is node 2; for degree 3, there are three nodes, namely nodes 3,4 and 5, and only one node of degree 4 which is node 1.

1.7.9 Connectivity

Connectivity [Golbeck, 2013], also known as cohesion, measures how the edges are distributed. Connectivity is the minimum number of nodes that would have to be removed to obtain a disconnected graph. That is to say, there is no longer a path from each node to every other node. Considering the graph in Figure 1.6, we need to delete only one node to make the graph disconnected.

1.8 Conclusion

This chapter is an elementary introduction to the field of complex networks. We analyze the relevance of topological properties of networks and their types, the

properties of a node, and the network's description measures. We also discussed the basic concepts of graph theory. Indeed, we can use graphs in several disciplines-for example, community detection, system recommendations, spam E-mail detection, and disease prediction. In the next chapter, we will review one of the most crucial active zones in complex network analysis: the link prediction problem.

Chapter

2 Link prediction in complex networks

2.1 Introduction

Since the origins of social and behavioural sciences, the link prediction problem has gotten much attention. Social networks, for example, Facebook, Twitter, Linkedin, and others, change over time as new connections appear in the graph. One of the most challenging tasks for these networks is to determine the best recommendations for users reliably. The primary goal of the link prediction issue, in terms of graph meaning, is to anticipate upcoming links given the current state of a graph. Link prediction techniques use various score functions, such as the Jaccard coefficient, Katz index, Adamic Adar metric, and others, to assess the likelihood of adding links to the network. Because of their simplicity and interpretability, these metrics are widely utilized in various applications. However, the majority of the measures are created for a particular topic. Social networks get quite vast when many types of links connect many individuals. Predicting those relationships is a difficult challenge since we must figure out how to make predictions that are as accurate as possible.

This chapter will define the notations and background needed to solve the link prediction problem. Then, we will introduce some examples of applications of link prediction. We introduce the similarity measures and shed light on other

probabilistic and maximum likelihood approaches.

2.2 Background and notations

The evolution of a social network is one of its most essential characteristics. We may look at its growth from two different perspectives. The first is the number of new users who join the network, and the second and most crucial is the network's link formation. We studied the link formation problem in the link prediction field. In recent years, social network analysis has received substantial attention because of its application in various domains, including recommendation systems, spam E-mail detection, and disease prediction.

In order to investigate and model a social network where entities are expressed as nodes and relationships between them are depicted as connections, social network analysis (SNA) utilizes methods from graph theory. Let $G(V, E)$ be a snapshot of a social network at time t. Link prediction methods and metrics [Liben-Nowell and Kleinberg, 2007] aim to accurately anticipate the ties that will be added to the graph at time t' from the edges in time t.

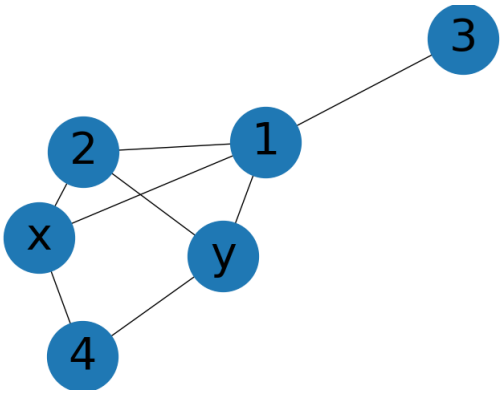


Figure 2.1: Graph example

Considering the network shown in Figure 2.1, for the rest of this thesis, we

will use the following terminology:

- $\Gamma(x)$ denotes the set of neighbours of node x. For example: $\Gamma(x) = \{1, 2, 5\}$ and $\Gamma(1) = \{2, 3, x, y\}$ etc.
- $|\Gamma(x)|$ represents the cardinality of the set.
  For example:$|\Gamma(x)| = 3$ and $|\Gamma(1)| = 4$
- A is the adjacency matrix of the graph $G(V, E)$ (see section 1.4.2).

2.3 Applications of link prediction

Link prediction is used in a wide variety of domains like:

1. System recommendations [Esslimani et al., 2011]: web personalization and information retrieval systems typically employ recommender systems. Collaborative filtering (CF) is the most commonly used recommendation technique. However, classic CF (CCF) systems only employ direct linkages and shared characteristics to represent user relationships. The authors present a novel densified behavioural network based on the collaborative filtering model (D-BNCF) based on the BNCF approach. BNCF analyzes navigational patterns to define user connections.

2. Spam Email detection [Sharma et al., 2021]: The popularity of social media has expanded due to the quick increase in the number of users. The expansion of social media and associated technologies has created an ecosystem in which individuals from all walks of life are linked across various social media platforms. At the same time, the fast rise of users on social media generates a threat to user data owing to cyber security branches and data theft. Hackers increasingly exploit spam emails to obtain data from social media accounts. They send spam emails with infected links, and when individuals click on the

links, their profiles' information is supplied to hackers. The authors Sharma et al. [2021] discuss the different procedures to secure the social media profile and present a spam filter using the decision tree model and other machine learning algorithms.

3. Disease prediction [Folino and Pizzuti, 2012]: In their research, the author aims to forecast the emergence of future illnesses based on people existing health conditions. A comorbidity network is developed, wherein nodes indicate diseases and edges denote the concurrent occurrence of two disorders in a patient. A ranked list of scores is created after applying similarity measures that assess the closeness of two nodes based simply on network topology. The stronger the connection score, the more probable it is that a relationship between the two illnesses will be identified. According to the authors Folino and Pizzuti [2012], the outcomes of the studies imply that the suggested approach may identify morbidities that a patient may acquire in the future.

4. Hyper link prediction [Adafre and de Rijke, 2005]: The authors' work addresses the issue of detecting missing hypertext links in Wikipedia. They offer a two-step method: first, they compute a cluster of very similar pages around a particular page and then discover candidate connections from those similar pages that may be absent on the given page. The key innovation is the LTRank method for detecting comparable pages, which ranks pages based on co-citation and page title information.

5. Protein-protein interactions (PPI) [Airoldi et al., 2006, Nasiri et al., 2021]: In these papers, the authors handle the protein-protein interaction forecasting issue as an LP problem in attributed networks and then estimate the connections between proteins in the PPI network using an attributed embedding technique. They especially seek to develop a customized form of Deepwalk

based on feature selection for tackling link prediction in protein-protein interactions, which would enhance both network structure and protein characteristics.

6. Hidden link among terrorists and their organizations [Al Hasan et al., 2006, Anil et al., 2015]: The authors analyze a massive dataset documenting multiple terrorist events worldwide and depict the collection as a heterogeneous network. Their paper aims to study the influence of different link prediction frameworks, including topic modelling, graph topology and graph kernels. They offer bipartite-based link prediction over topic feature connections, heterogeneous vertex proximity-based LP, and graph kernel techniques. Authors demonstrate that a bipartite strategy based on topic modelling produces equivalent (and occasionally better) results than node proximity and graph kernel.

2.4 Link prediction methods

Several link prediction algorithms have recently been proposed. According to [Kumar et al., 2020], these methods can be divided into similarity-based, probabilistic, and learning-based models. The similarity-based methods include three categories: local similarity metrics, global metrics, and semi-local (or quasi-local) metrics; The probabilistic and maximum likelihood metrics are divided into a hierarchical random graph, stochastic block model, local probabilistic model, probabilistic relational model, and exponential random graphs. Dimensionality reduction can also solve the link prediction problem with embedding models or matrix factorization models. The most used methods are similarity-based metrics because of their interpretability and simplicity.

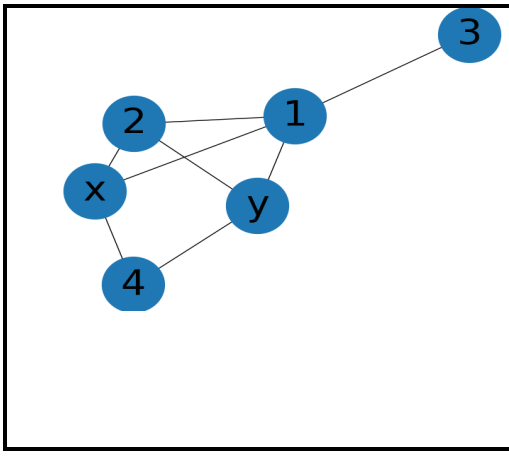
2.4.1 Similarity based metrics

2.4.1.1 Local similarity indices

Definition 2.4.1. *The preferential attachment metric is one of the most basic link prediction metrics. This metric was proposed by Barabâsi et al. [2002] and is based on the observation that some social networks follow a power-law distribution. PA refers to this phenomenon as "rich getting richer". As a consequence, the more friends a person has, the more new connections he or she will establish in the future. A new link between two nodes with a high degree is more likely to be formed in the graph. The PA metric is given by*

$$PA(x, y) = |\Gamma(x)|.|\Gamma(y)| \tag{2.1}$$

Example 2.4.1. *Let's consider the graph in Figure 2.1, the unconnected nodes are:* $\{('x', 3), ('x', 'y'), (1, 4), (2, 4), (2, 3)\}$. *The scores of the links are given in table 2.1.*



PA ('x', 3) = 3
PA ('x', 'y') = 9
PA (1, 4) = 8
PA (2, 4) = 6
PA (2, 3) = 3

Table 2.1: Example of PA algorithm.

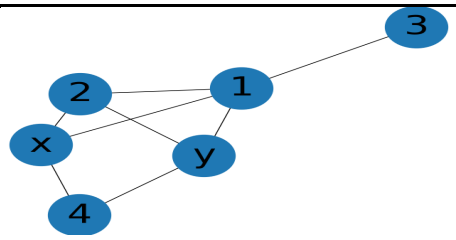
The degree of node x is three, and the degree of node y is also three then, $PA(x, y) = 3 \times 3 = 9$. *From Table 2.1, we can conclude that the edge between*

nodes x and y will be formed first, then the edge between nodes 1 and 4, followed by the edge between nodes 2 and 4. This metric performs the worst in most real datasets, as shown in the following chapters.

Definition 2.4.2. *The common neighbour (CN) measure [Newman, 2001] asserts that "if two vertices overlap numerous contacts, they are often more probably to meet". In a complex network, the chance that nodes x and z will be linked rises if node x is connected to node y and node z is connected to node y.*

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{2.2}$$

Example 2.4.2. *The common neighbour metric was applied to unconnected nodes for the graph in Figure 2.1. For instance, nodes x and 3 shares only one node, but nodes x and y share 3 common neighbours that are* $\{4, 2, 1\}$*, so the metric gives 3 as a score to that link which means that it will probably appear in future. The score of the other links are given as follows:*



CN(x , 3)= 1
CN(x , y)= 3
CN(1 , 4)= 2
CN(2 , 4)= 2
CN(2 , 3)= 1
CN(4 , 3)= 0
CN(3 , y)= 1

Table 2.2: Example of CN algorithm.

From Table 2.2, we can conclude that the link between x and y will be formed

first (the nodes x and y share three neighbours, as we have shown before), followed by the links $(1, 4)$ and $(2, 4)$. The nodes forming those edges share only two nodes. Based on the CN metric, nodes 4 and 3 can not be connected in the future.

Definition 2.4.3. *The authors in [Ravasz et al., 2002] have introduced hub promoted (HP) as the topological overlap between pair of nodes x and y normalized by the minimum of their degrees. The authors preferred relations between simple and hub nodes in this measurement, making it appropriate for hierarchical networks.*

$$Hub\ Promoted(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{min\{|\Gamma(x)|, |\Gamma(y)|\}} \tag{2.3}$$

Example 2.4.3. *In table 2.3, there are no hub nodes, the majority of links have a score of one.*

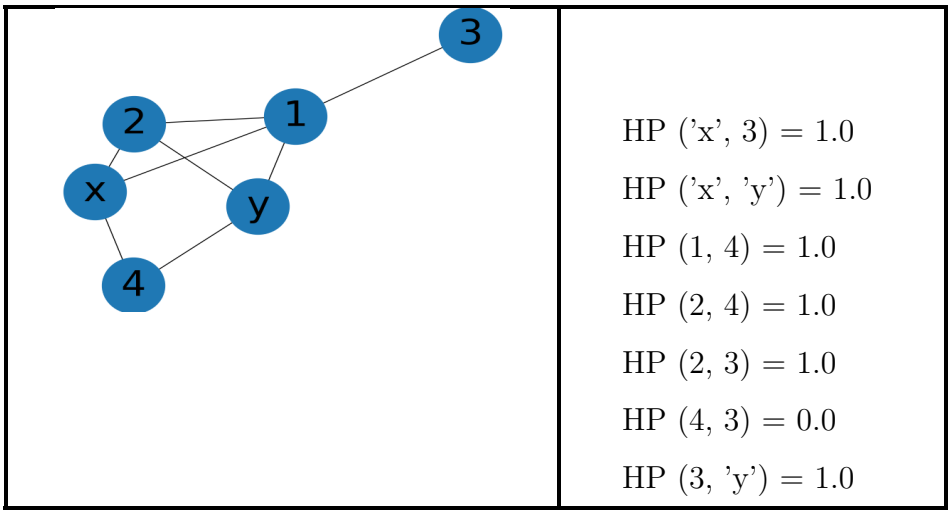
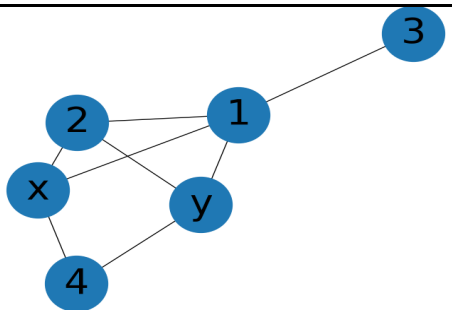


Table 2.3: Example of HP algorithm.

Definition 2.4.4. *Hub depressed (HD) [Ravasz et al., 2002]: is identical to Hub promoted. In HD, the authors considered the maximum degrees of the two nodes x and y. The Hub depressed serves the opposite purpose, allowing edge formation*

between simple nodes rather than nodes and hubs.

$$Hub\ Depressed(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{max\{|\Gamma(x)|, |\Gamma(y)|\}} \tag{2.4}$$

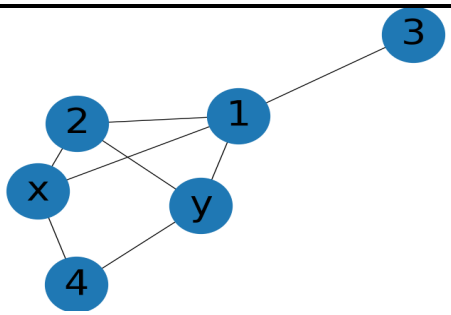


HD ('x', 3) = 0.333
HD ('x', 'y') = 1.0
HD (1, 4) = 0.5
HD (2, 4) = 0.667
HD (2, 3) = 0.333
HD (4, 3) = 0.0
HD (3, 'y') = 0.333

Table 2.4: Example of HD algorithm.

Definition 2.4.5. *Jaccard coefficient (JA) [Jaccard, 1901]: In order to evaluate the similarity between two sets, Jaccard has proposed equation 2.5, where* $\Gamma(x)$ *is the set of friends of the first node and* $\Gamma(y)$ *is the set of friends of y. This measure considers a normalization of the common neighbour similarity.*

$$Jaccard(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{2.5}$$



JA ('x', 3) = 0.333
JA ('x', 'y') = 1.0
JA (1, 4) = 0.5
JA (2, 4) = 0.667
JA (2, 3) = 0.333
JA (4, 3) = 0.0
JA (3, 'y') = 0.333

Table 2.5: Example of JA algorithm.

From Table 2.5, the link between nodes x and y will be formed first because $|\Gamma(x) \cap \Gamma(y)| = 3$ and $|\Gamma(x) \cup \Gamma(y)| = 3$ as a results of that, $JA(x, y) = 1$. The next edge that JA centrality will form is between nodes 2 and 4 with a probability of $JA(2, 4) = 0.667$. Similarly, for the other edges, with respect to the order of apparition. The link $(3, 4)$ will not be formed because $JA(3, 4) = 0$.

Definition 2.4.6. *Leicht-Holme-Nerman (LHN) assigns greater scores to edges connecting nodes with numerous shared common neighbours compared to the expected number of such neighbours [Leicht et al., 2006]. The following equation defines the LHN index:*

$$LHN(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)|.|\Gamma(y)|} \tag{2.6}$$

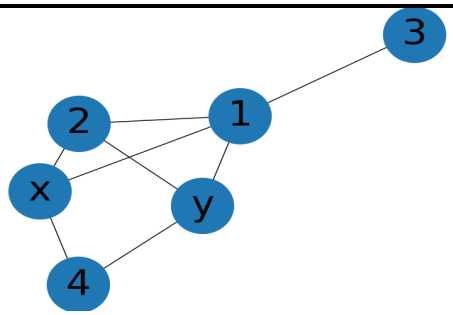
	LHN ('x', 3) = 0.333 LHN ('x', 'y') = 0.333 LHN (1, 4) = 0.25 LHN (2, 4) = 0.333 LHN (2, 3) = 0.333 LHN (4, 3) = 0.0 LHN (3, 'y') = 0.333

Table 2.6: Example of LHN algorithm.

Definition 2.4.7. *Parameter dependent (PD) [Zhu et al., 2012] claims that the use of the free parameter δ increases the link prediction algorithm's accuracy as shown in equation 2.7. Note that when $\alpha = 0$, the parameter-dependent metric is equal to CN, and when $\alpha = \frac{1}{2}$, the parameter-dependent metric is identical to the Salton index.*

$$PD(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{(|\Gamma(x)|.|\Gamma(y)|)^{\delta}} \tag{2.7}$$

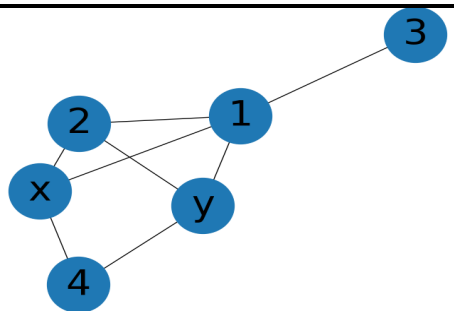
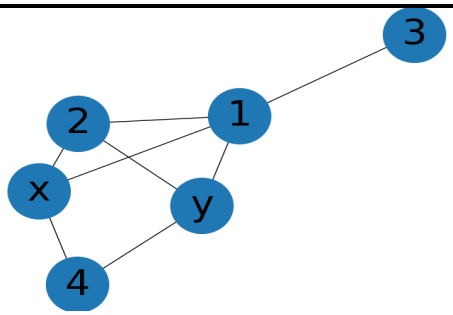
	PD ('x', 3) = 0.978 PD ('x', 'y') = 2.871 PD (1, 4) = 1.919 PD (2, 4) = 1.93 PD (2, 3) = 0.978 PD (4, 3) = 0.0 PD (3, 'y') = 0.978

Table 2.7: Example of PD algorithm.

Definition 2.4.8. *Adamic/Adar (AA)[Adamic and Adar, 2003]: This measurement assumes that some people who work with fewer individuals choose their connections more wisely than those who work with more people. As a result, a celebrity acquaintance who is a close friend is seen as less important. However, a shared friend with fewer contacts might be more significant when evaluating this resemblance. AA measure is based upon the Jaccard coefficient :*

$$AA(x,y) = \sum_{|z \in \Gamma(x) \cap \Gamma(y)|} \frac{1}{log|\Gamma(z)|} \tag{2.8}$$

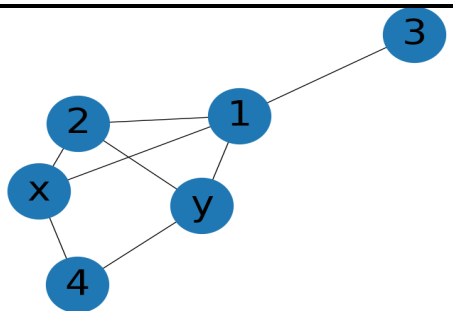


AA ('x', 3) = 0.721
AA ('x', 'y') = 3.074
AA (1, 4) = 1.82
AA (2, 4) = 1.82
AA (2, 3) = 0.721
AA (4, 3) = 0
AA (3, 'y') = 0.721

Table 2.8: Example of AA algorithm.

Definition 2.4.9. *Salton, or common cosine metric [Salton and McGill, 1983], used it first to calculate the similarity between two angles. Later, researchers used it to measure the similarity between two nodes x and y using the sets of their neighbours. This metric is given by:*

$$Salton(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)|.|\Gamma(y)|}} \tag{2.9}$$



salton ('x', 3) = 0.577
salton ('x', 'y') = 1.0
salton (1, 4) = 0.707
salton (2, 4) = 0.816
salton (2, 3) = 0.577
salton (4, 3) = 0.0
salton (3, 'y') = 0.577

Table 2.9: Example of Salton algorithm.

Definition 2.4.10. *Sorensen: In [Sorensen, 1948, Kumar et al., 2020], we may deduce from this definition that the Sorensen index is based on the same assumption as Jaccard and Adamic/Adar since the author considers the number of common neighbours and emphasizes that nodes with a lower degree will contribute more to link creation.*

$$Sorensen_{x,y} = \frac{2.|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|} \tag{2.10}$$

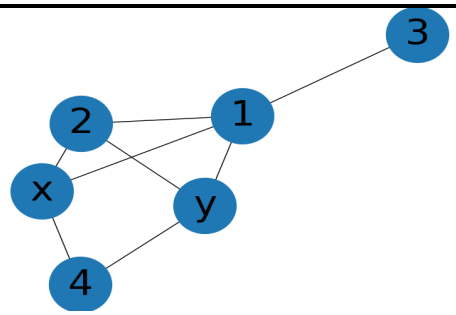
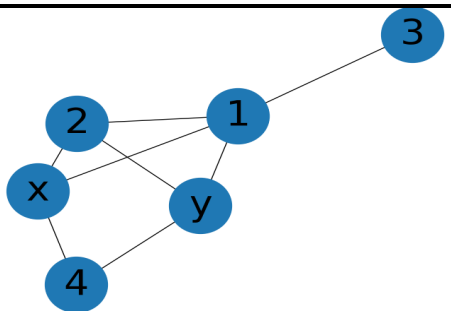
	sorensen ('x', 3) = 0.25 sorensen ('x', 'y') = 0.5 sorensen (1, 4) = 0.333 sorensen (2, 4) = 0.4 sorensen (2, 3) = 0.25 sorensen (4, 3) = 0.0 sorensen (3, 'y') = 0.25

Table 2.10: Example of Sorensen algorithm.

Definition 2.4.11. *Resource allocation (RA) [Zhou et al., 2009] heavily neglects the contributions of shared neighbours with high degrees. When the dataset is large, the AA and RA produce similar findings.*

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|} \tag{2.11}$$



RA ('x', 3) = 0.25
RA ('x', 'y') = 1.083
RA (1, 4) = 0.667
RA (2, 4) = 0.667
RA (2, 3) = 0.25
RA (4, 3) = 0
RA (3, 'y') = 0.25

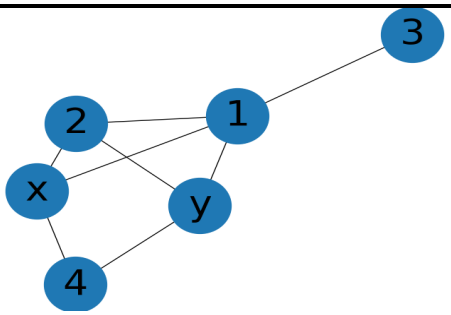
Table 2.11: Example of RA algorithm.

Definition 2.4.12. *According to [Soundarajan and Hopcroft, 2012], nodes that belong to the same community are more likely to be connected; this means that the similarity metrics that are presented previously could be extended by adding a new term that awards extra points to links constructed by nodes from the same community:*

$$CN_{SH}(x,y) = |\Gamma(x) \cap \Gamma(y)| + \sum_{w \in \Gamma(x) \cap \Gamma(y)} f(w) \tag{2.12}$$

Where $f(w) = 1$ if w belongs to the exact same community as x and y or 0 elsewhere and $\Gamma(x)$ signifies the set of acquaintances of x.

Example 2.4.4. *In the following example, we will add an extra piece of information to the graph using community detection; we will consider that nodes $\{x, y, 4\}$ belong to the same community 1 and nodes $\{1, 2, 3\}$ in the same community 2.*



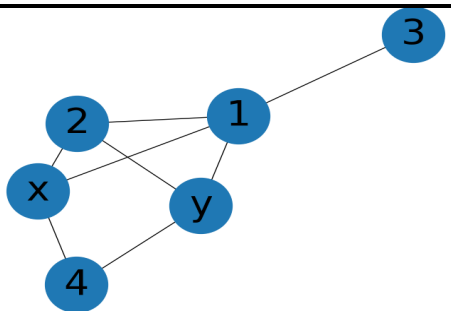
CNSH ('x', 3) = 1
CNSH ('x', 'y') = 4
CNSH (1, 4) = 2
CNSH (2, 4) = 2
CNSH (2, 3) = 2
CNSH (4, 3) = 0
CNSH (3, 'y') = 1

Table 2.12: Example of CN Soundarajan Hopcroft algorithm.

Definition 2.4.13. *The authors of WIC index [Valverde-Rebaza and Andrade Lopes, 2012] consider that for two nodes u and v, if a common neighbour w belongs to the same community as them, w is considered as a within-cluster common neighbour of u and v. Otherwise, it is regarded as an inter-cluster common neighbour of u and v. The ratio between the size of the set of within- and inter-cluster common neighbours are defined as the WIC measure.*

$$s_{x,y}^{WIC} = \frac{|\Lambda_{x,y}^{W}|}{|\Lambda_{x,y}^{IC}| + \delta} \tag{2.13}$$

Note that δ is a free parameter to prevent the devision by zero $\delta \approx 0$.



WIC ('x', 3) = 0
WIC ('x', 'y') = 0.5
WIC (1, 4) = 0
WIC (2, 4) = 0
WIC (2, 3) = 1000.0

Table 2.13: Example of WIC algorithm.

Definition 2.4.14. *The CAR index is based on the Common neighbour metric (see equation 2.2). The authors Cannistraci et al. [2013] consider that a link exists between two vertices if their shared neighbours are members of a local community. The CAT metric is defined as:*

$$CAR(x,y) = CN(x,y) \times LCL(x,y) = CN(x,y) \times \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{|\gamma(z)|}{2} \tag{2.14}$$

$LCL(x,y)$ identifies the number of local community ties defined as the connections among the shared neighbours of seed nodes x and y. $\gamma(z)$ represents the subset of neighbours of node z that are also common neighbours of x and y. Note that other variations of this metric exist for other similarity metrics like the CAR-based Adamic/Adar index, CAR-based Resource Allocation Index (CRA), etc.

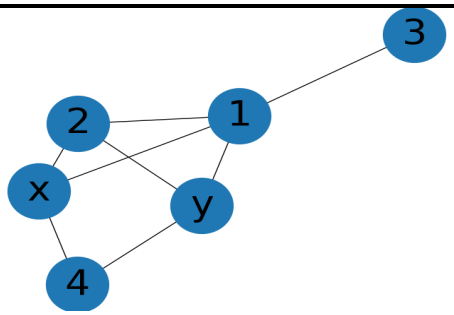
	CAR ('x', 3) = 0.0 CAR ('x', 'y') = 3.0 CAR (1, 4) = 0.0 CAR (2, 4) = 0.0 CAR (2, 3) = 0.0

Table 2.14: Example of CAR algorithm.

Remark 2.4.1. *From the above, we can sum up that the local metrics are easy to interpret and have a slight computational complexity. The presented metrics provide good results when applied to real-world datasets. However, there is always room to enhance their certainty or propose new metrics with higher accuracy.*

2.4.1.2 Global similarity indices

Global metrics use the entire topology of the network, and because of that, their computational time is very high. The most known global metrics are Katz, Shortest path, and SimRank. We will use the following graph to calculate the global metrics.

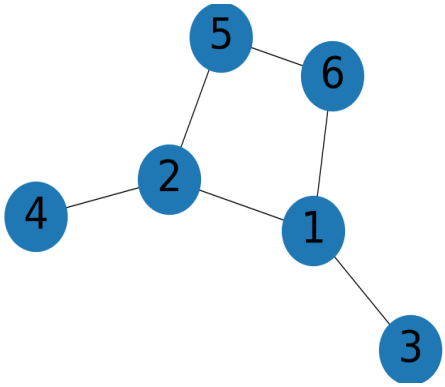


Figure 2.2: Graph example for Global metrics

Definition 2.4.15. *According to [Liben-Nowell and Kleinberg, 2007],* ***Shortest Path*** *can be seen as a metric that quantifies the strength of the relation between two nodes. If nodes x and y are close, the probability of their connection increases. To express this idea, the inverse relation between similarity and shortest path is captured by the equation 2.15.*

$$S(x, y) = -|d(x, y)| \tag{2.15}$$

Where d is the shortest path between nodes x and y and $|d(x, y)|$ is the length of this shortest path.

Example 2.4.5. *Lets consider the graph in Figure 2.2:*

Table 2.15: Shortest path algorithm applied to graph in Figure 2.2.

Source node	*Destination node*	*Score*
1	*4*	*-3*
1	*5*	*-3*

Continued on next page

Table 2.15: Shortest path algorithm applied to graph in Figure 2.2. (Continued)

2	*3*	*-3*
2	*6*	*-3*
3	*4*	*-4*
3	*5*	*-4*
3	*6*	*-3*
4	*5*	*-3*
4	*6*	*-4*

We can notice that this metric gives the same score for too many edges. This metric has low accuracy compared to the local metrics.

Definition 2.4.16. *Katz [Katz, 1953]: This measure may be seen as an improvement of the shortest path measure. It immediately combines all paths between x and y and penalizes lengthier paths by dumping factor. It is mathematically expressed as:*

$$S = (I - \beta A)^{-1} - I \tag{2.16}$$

Where I is the identity matrix of size $n \times n$ and A is the graph's adjacency matrix.

Example 2.4.6. *Using the graph in Figure 2.2 and equation 2.16, the probability of links is given in the following table:*

Table 2.16: Katz algorithm applied to graph in Figure 2.2.

Source node	*Destination node*	*Score*

Continued on next page

Table 2.16: Katz algorithm applied to graph in Figure 2.2. (Continued)

1	*4*	*0.00010006003101570795*
1	*5*	*0.0002001100560283143*
2	*3*	*0.00010006003101570795*
2	*6*	*0.00020011005602831432*
3	*4*	*1.0006003101570794e-06*
3	*5*	*2.001100560283143e-06*
3	*6*	*0.00010005002501260637*
4	*5*	*0.00010005002501260637*
4	*6*	*2.001100560283143e-06*

Definition 2.4.17. *Cosine based on the inverse of laplacian matrix L^+ [Fouss et al., 2007, Kumar et al., 2020]: In spectral graph theory, the Laplacian matrix is often utilized as an alternate representation of networks. This matrix can be expressed as $L = D - A$, where D is the diagonal matrix containing the degrees of each node and A is the graph's adjacency matrix. The pseudo-inverse of the laplacian matrix is denoted as L^+, and each element of this matrix represents the similarity score between the two nodes x and y:*

$$S(x, y) = \frac{L^+_{x,y}}{\sqrt{L^+_{x,x} L^+_{y,y}}} \tag{2.17}$$

Example 2.4.7. *Using the graph in Figure 2.2 and equation 2.17, the probability of links are given in the following table:*

Table 2.17: Cosine based on the inverse of laplacian matrix algorithm applied to the graph in Figure 2.2.

First node	*Second node*	*Score*
1	*4*	*-0.4527165970184822*
1	*5*	*-0.3167213152453097*
2	*3*	*-0.45271659701848177*
2	*6*	*-0.31672131524530894*
3	*4*	*-0.42446043165467606*
3	*5*	*-0.4248533580175029*
3	*6*	*-0.2383323715707942*
4	*5*	*-0.23833237157079298*
4	*6*	*-0.42485335801750235*

Definition 2.4.18. *Average Commute time (ACT): In [Liu and Lü, 2010], the authors have defined ACT as the average number of movements/steps needed by a random walker to arrive at the final node y and return to the starting node x. The average commute time equation is defined as:*

$$S(x,y) = \frac{1}{l^+_{xx} + l^+_{yy} - 2l^+_{xy}} \tag{2.18}$$

l^+_{xy} *denotes the* (x,y) *entry of the matrix* L^+.

Example 2.4.8. *Using the graph in Figure 2.2 and equation 2.18, the probability of links are given in the following table:*

Table 2.18: Average Commute time algorithm applied to the graph in Figure 2.2.

First node	*Second node*	*Score*
1	*4*	*0.5714285714285712*
1	*5*	*0.9999999999999991*
2	*3*	*0.5714285714285711*
2	*6*	*1.0*
3	*4*	*0.36363636363636354*
3	*5*	*0.49999999999999933*
3	*6*	*0.571428571428571*
4	*5*	*0.5714285714285718*
4	*6*	*0.5*

Definition 2.4.19. *SimRank [Jeh and Widom, 2002] is a measure for comparing structural contexts and visualizing object-to-object connections. It is not domain-specific and may be utilized in both directed and undirected systems.*
The idea behind this metric is that two items are similar if they are connected to other similar objects. SimRank measures the time it takes for two random walkers to meet each other from two distinct starting positions. The measure is calculated in a recursive manner using the following equation:

$$\begin{cases} \frac{\alpha}{k_x k_y} \sum_{i=1}^{k_x} \sum_{j=1}^{k_y} S(\Gamma_i(x), \Gamma_j(y)) & \text{if } x \neq y \\ 1 & \text{otherwise} \end{cases} \tag{2.19}$$

α is a free parameter between 0 and 1. Note that the computational complexity of this measure is very high so it can not be applied to a small networks.

Example 2.4.9. *Using the graph in Figure 2.2 and equation 2.19, the probability of links are given in the following table:*

Table 2.19: SimRank algorithm applied to the graph in Figure 2.2.

First node	*Second node*	*Score*
1	*4*	*0.7541126042067663*
1	*5*	*0.759778799360726*
2	*3*	*0.7541126042067663*
2	*6*	*0.759778799360726*
3	*4*	*0.0*
3	*5*	*0.0*
3	*6*	*0.791860884907265*
4	*5*	*0.791860884907265*
4	*6*	*0.0*

2.4.1.3 Quasi-local similarity indices

Quasi-local metrics came as a trade-off between local metrics known by their low complexity and global metrics known by their high complexity. These metrics are as efficient as local metrics but use the entire topology of the network. However, the complexity of quasi-local is still below the global metrics (The complexity of all metrics is discussed in [Kumar et al., 2020]). Next, we will introduce the most known quasi-local metric and apply it to the graph in Figure 2.2.

Definition 2.4.20. *Local path [Lü et al., 2009] is a modification of the Katz index. LP only takes pathways of lengths two and three into account. This index minimizes*

the impact of the paths of length three. Local path is given by:

$$S^{LP}(x, y) = A^2 + \epsilon A^3 \tag{2.20}$$

Where A^2 denotes the number of paths of length 2 between x and y, A^3 equals the different paths of length 3 between x and y. ϵ represents a free parameter in the range $[0, 1]$. If $\epsilon = 0$, this metric gives the same results as CN in equation 2.2.

Example 2.4.10. *Table 2.20 show the results of applying local path equation 2.20 to the graph in Figure 2.2.*

Table 2.20: Local Path algorithm applied to the graph in Figure 2.2.

First node	*Second node*	*Score*
1	*4*	*0.05*
1	*5*	*1.0*
2	*3*	*1.0*
2	*6*	*0.05*
3	*6*	*0.02*
3	*4*	*1.0*
3	*5*	*0.01*
6	*4*	*0.04*
4	*5*	*0.02*

2.4.2 Link prediction using machine learning

Many works have been proposed to solve link prediction using machine learning and deep learning techniques such as [Li and Chen, 2013, Gu and Chen, 2016], where the authors take advantage of network topological properties and attribute

data. We can also treat the link prediction problem as a learning-based model. The issue is framed as a supervised classification model, where edges that are part of the graph can be labeled as 1, while links that are not a part of the graph can be marked as 0 or -1.

$$l_{(x,y)} = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & else \end{cases} \tag{2.21}$$

A similarity metric of an edge or a node's characteristics could represent an edge feature; this enables us to use machine learning techniques like KNN or logistic regression and random forest (refer to [Raschka and Mirjalili]). To sum up, we should apply the following steps to convert the link prediction issue to a binary classification task:

1. Graph construction, the goal of this step is to build the graph from the data available.

2. Graph kernel construction: In this stage, we need to extract the properties of edges and nodes, then label them into positive and negative edges as shown in equation 2.21.

3. Applying models in this step, we apply models like KNN, random forest, SVM, etc.

4. Using the structural characteristics of the edges and the nodes, we may predict the label of new edges in a graph.

2.4.3 Other approaches

In this section, we will introduce other techniques based on several principles. Still, the major drawback of those approaches is their high complexity and the amount

of information required to run the algorithms. For these reasons, we will only refer to them.

In addition to specific guidelines and parameters determined by the observed structure's maximum likelihood, probabilistic models also use organizing aspects of the network structure. After that, we can compute the probability of any missing link using previously defined rules and parameters. The maximum likelihood approaches have a clear disadvantage; they are much longer. A well-constructed algorithm can handle very small-sized networks in an acceptable duration but cannot deal with massive online networks with many nodes and links. Moreover, the maximum likelihood approaches are not the most precise compared to local metrics or even quasi-local and global metrics that provide more accurate results (refer to [Lü and Zhou, 2011] for more detail about those algorithms). **Link prediction using dimensionality reduction**, according to [Pecli et al., 2015], the authors have used the machine learning technique with dimensionality reduction to solve this problem. The experimental process was divided into three major stages: preprocessing, followed by configuration, and then evaluation. The first stage prepares the dataset for the subsequent phases; In the second stage, the authors define the settings for the dimensionality reduction strategy and the classification algorithm. The final stage trains and evaluates the learning model across the reduced datasets. Note that the features that have been passed to the classifiers were local similarity metrics and node attributes that have been proposed in chapter 1 and chapter 2.

2.5 Evaluation metric

Consider the graph $G(V, E)$, where V is the set of nodes, and E is the set of edges. G is an undirected and unweighted where loops and multiple edges are not allowed. We define two subsets from E based on the graph $G(V, E)$, where E_{train} is the training set and E_{test} is the testing set. 90 percent of the links are considered

in the training set, and the rest 10 percent are considered as a test set.

$$\begin{cases} E_{train} \cup E_{test} &= E \\ E_{train} \cap E_{test} &= \emptyset \end{cases} \tag{2.22}$$

The Area under the receiver operating characteristic curve (AUC) is used to evaluate a link prediction metric. The probability that the score of an edge selected randomly from the E_{test} is higher than that of an edge randomly chosen from the $\bar{E}$ is defined as AUC. Let suppose that e_1 is a link from the E_{test} and e_2 is a non-existing link randomly selected from $\bar{E}$; First, we calculate the score of e_1 as s_{e1} and e_2 as s_{e2} using a similarity-based metric that have been introduced previously in this chapter, then, we compare the two obtained scores s_{e1} and s_{e2}. The following formula gives the Area under the receiver operating characteristic curve:

$$AUC = \frac{N^{'} + 0.5 * N^{''}}{N} \tag{2.23}$$

1. N denotes the number of independent comparisons.
2. $N^{'}$ is the number of times the missing link s_{e1} has higher similarity score than s_{e2}. In other words, the number of times $s_{e2} < s_{e1}$
3. $N^{''}$ is the number of times the missing link and the non-existent link have the same similarity score $s_{e2} = s_{e1}$.

The studies Zhu et al. [2012], Hanley and McNeil [1982] show that the accuracy should be about 0.5 if the scores are produced using an independent and identical distribution. As a result, the accuracy value that is more than 0.5 indicates how much better the algorithm performs than random chance. Because the link prediction problem can be modeled as a binary classification task [Al Hasan et al., 2006], the majority of the evaluation metrics (refer to [Lichtnwalter and Chawla,

2012] for more detail), such as precision, F-score, applies to any binary classification task. We can also apply those evaluation metrics to link prediction, but because the extreme imbalance denotes this problem, we use AUC. As a result, the confusion matrix can be used to express the evaluation of a binary classification problem with two classes [Schütze et al., 2008, Ayoub et al., 2020, 2022].

2.6 Conclusion

This chapter is a fundamental introduction to the field of link prediction. We present the relevant metrics and methods for solving the link prediction problem. We have introduced the local metrics that rely on neighbors of nodes. Then, we presented global and semi-local metrics and gave some examples of each type. Besides, we introduced other methodologies, such as probabilistic methods and dimensionality reduction, and then demonstrated their shortcomings. We have also defined the evaluation criterion used in the link prediction field. In the next chapter, we will introduce our first work [Ayoub et al., 2020] in which we propose a new similarity index based on path length between a pair of unlinked nodes and their degree. This metric has a significant advantage over state-of-the-art metrics.

Chapter

3 Accurate link prediction method based on path length between a pair of unlinked nodes and their degree

3.1 Introduction

We can infer from the preceding chapter that the link prediction problem has attracted much attention since the birth of social and behavioral sciences. To address this issue, numerous metrics have been suggested. Due to their readability and simplicity, local metrics are frequently employed in many applications. The majority of them, nevertheless, are constructed to a specific domain. The complexity of the global and quasi-local measures is considerable, and they might need a lot of RAM. Predicting the forthcoming links is, therefore, still challenging because we need to figure out how to make forecasts as accurate as possible.

We introduce our new measure in this chapter, based on the paths and degrees of nodes between the source and destination nodes. We employed simple topological properties that are easy to compute and efficient in solving the link prediction field. Furthermore, we analyze the effect and influence of path length on method precision and show that the proposed technique offers better recommendations when path lengths 2 and 3 are used. Then, using the area under the curve, we compare 13 state-of-the-art approaches to the suggested method in terms of prediction performance. The results from five social network examples demonstrate the efficacy of the proposed method in generating reliable recommendations. Fur-

thermore, to address the link prediction problem as a binary classification task, we explore machine learning approaches such as logistic regression and artificial neural network, as well as K-nearest neighbors and decision trees. Also, we used support vector machine and random forest. The findings support the proposed metric's ability to improve accuracy significantly.

One of the most notable features of a social network is its evolution. This expansion can be perceived from two distinct perspectives: The first is the number of new nodes entering the network, and the second is the construction of links or edges between users. This latter is investigated in the link prediction. The link prediction has received considerable attention in the last five years because of its application in several domains, such as spam E-mail detection [Chen et al., 2005], disease prediction [Folino and Pizzuti, 2012], system recommendations [Esslimani et al., 2011]. In order to fully understand and model a social network where individuals are modeled as nodes and interactions between them are represented by edges, as we have seen in the previous chapter, SNA employs methods from graph theory (refer to chapter 1 and chapter 2). In Link Prediction (LP), we aim to properly estimate the edges that will be added to the network between time t and a specified future time t' given a snapshot of a social network at time t (Note that $t < t'$)[Liben-Nowell and Kleinberg, 2007].

The authors in different papers and surveys [Al Hasan et al., 2006, Lü and Zhou, 2011, Wang et al., 2015, Martínez et al., 2016] have labeled three main groups of solutions to the link prediction problem; two of those categories (namely maximum likelihood methods and probabilistic methods) could not process big graphs that contains a considerable number of edges due to the high complexity of those algorithms. The most used techniques are similarity-based methods that compute the likeness/similarity between two users (in this solution category, the

more two people are similar, the higher the probability of their connection will be).

In the present chapter, we will present a new similarity metric [Jibouni et al., 2018, Ayoub et al., 2020] our proposed index is based on a network's local properties, primarily the node degrees, as well as a semi-local features, the path length. The proposed score is unique because it considers pathways of various lengths. We suppose that a node can communicate with another node through various paths. Also, we consider that the path length significantly influences the interactions between nodes. If two nodes are close together, they can be rapidly linked, but if they are far apart, one of them is unlikely to cross the other. The degrees of these laters also influence how much information is carried from source to destination nodes, so using the path length alone to judge how similar two nodes do not produce good results.

In order to reduce the contribution of high-degree vertices, we consider the degree of the source and destination vertices (nodes with numerous friends). Because of their high degree, those nodes have less role in developing friendships; the quantity of information intercepted reduces as the degree of source, destination, and intermediate nodes grow.

Moreover, since many state-of-the-art methods have been presented to approach a given issue/dataset, we evaluated the proposed measure on five datasets from distinct domains. We evaluate the proposed approach and 13 other link prediction algorithms using simulations. Our proposed metric is parametrized. For that reason, we have tested different values of l; the simulation shows that when restricting the path depth to $l = 2$ and $l = 3$, the introduced measure provides greater accuracy than the present similarity-based measures. We can infer that the use of pieces of information from the second and third-degree neighbors has a positive impact on the accuracy. After that, we used the methodology presented in chapter 2, indicating that link prediction can be solved as a classification task.

We have utilized several machine learning and deep learning algorithms to classify links such as K-nearest neighbors and Artificial Neural networks. The results indicate that the effectiveness of our parameter-free technique reaches a maximum improvement of 40%.

3.2 The proposed metric

3.2.1 Definition

This section presents our new index for expressing the proximity or similarity of two nodes. Let two vertices, x, and y, in a simple graph G. The probability of the future connection between the two nodes is given by $J(x, y)$; their similarity value determines this probability at present. Our proposed measure is a parameter-free metric that relies on the number of paths connecting the two nodes, x, and y. In a social network, **the future connection probability increases if the nodes x and y have numerous paths**. Let us assume that vertice x has k units to distribute among his acquaintances. Then each friend will receive $\frac{k}{|\Gamma(x)|}$, suggesting that the bigger the degree /number of acquaintances on the graph, the fewer the number of units they will obtain from the node x. As a result, **our suggested metric diminishes the role of nodes with a high degree.** Occasionally, some nodes have degree 0; to prevent the division by zero, we add 1 to the denominator. We define $J(x, y)$ measure between two nodes x and y as:

$$J(x,y) = \frac{|path_{x,y}^{length \leq l}|}{|\Gamma(x)| + |\Gamma(y)| + 1} \tag{3.1}$$

where

- $|path_{x,y}^{length \leq l}|$ denotes the amount of paths connecting x and y note that the

$length \leq l$. We are using only simple paths in our metric; That is to say, paths such as $1- > 3- > 4- > 3- > 1$ are not allowed.

- The maximum path length connecting the two nodes x and y is l.
- The number of neighbors of a node x is $|\Gamma(x)|$.

3.2.2 Example

Lets consider the graph in Figure 2.2 and calculate the J score to the edge $(6, 4)$: the number of the path of length less than 3 is two (namely $[6, 1, 2, 4], [6, 5, 2, 4]$) the degree of node 6 is 2 and the degree of node 4 is 1, then, the score according to equation 3.1 is:$\frac{2}{2+1+1} = 0.5$. On the other hand, $CN(6, 4) = 0$ because nodes 4 and 6 do not share any common neighbors.
Next, we will apply the CN metric and our proposed metric to the edges from $\bar{E}$.

Table 3.1: Comparison of scores between CN metric equation 2.2 and J metric equation 3.1.

CN scores	J scores
$(1, 4) = 1$	$(1, 4) = 0.2$
$(1, 5) = 2$	$(1, 5) = 0.33$
$(2, 3) = 1$	$(2, 3) = 0.2$
$(2, 6) = 2$	$(2, 6) = 0.33$
$(3, 6) = 1$	$(3, 6) = 0.25$
$(3, 4) = 0$	$(3, 4) = 0.33$
$(3, 5) = 0$	$(3, 5) = 0.5$
$(6, 4) = 0$	$(6, 4) = 0.5$

Continued on next page

Table 3.1: Comparison of scores between CN metric equation 2.2 and J metric equation 3.1. (Continued)

$(4, 5) = 1$	$(4, 5) = 0.25$

3.2.3 Distribution of scores

In Figure 3.1 and Figure 3.2, we used CN on Power Grid and les miserable datasets to determine the proportion of scores. The x-axis describes the scores assigned to all the edges. The y-axis illustrates the number of edges with a specific score.
In Figure 3.3 and Figure 3.4, we used our proposed metric on Power Grid and les miserable dataset to determine the proportion of scores. The x-axis describes the scores assigned to all the links. The y-axis illustrates the number of edges with a specific score. The parameter-free metric produces highly realistic results regarding the proportion of scores by distinguishing the relationships.

Unlike our suggested metric, the common neighbor index gives the score zero to most of the edges (see Figure3.1 and Figure3.2). This is known as the degenerate state problem [Zhou et al., 2009] and [Shao and Xu, 2016]; it can be solved using a parameter-free metric in which the percentage of scores are balanced, and the ties are much more distinct.

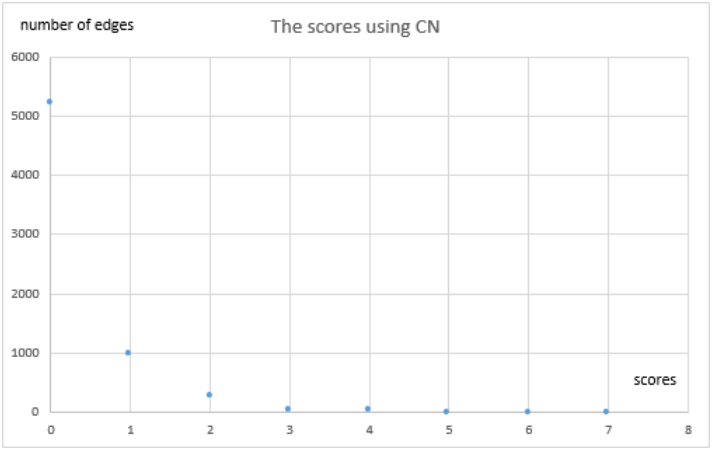


Figure 3.1: The proportion of scores in the power grid dataset using CN.

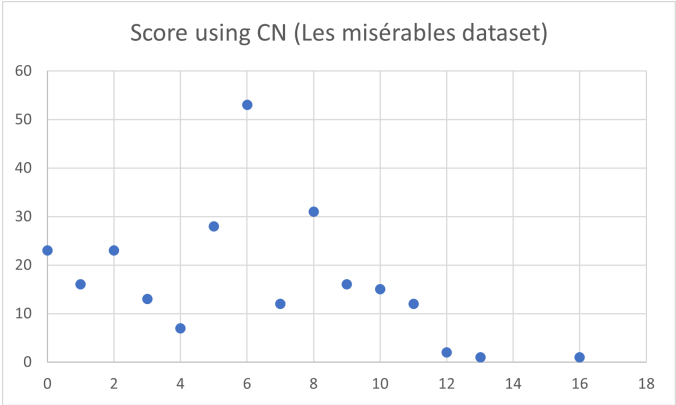


Figure 3.2: The proportion of scores in Les Miserables dataset using CN.

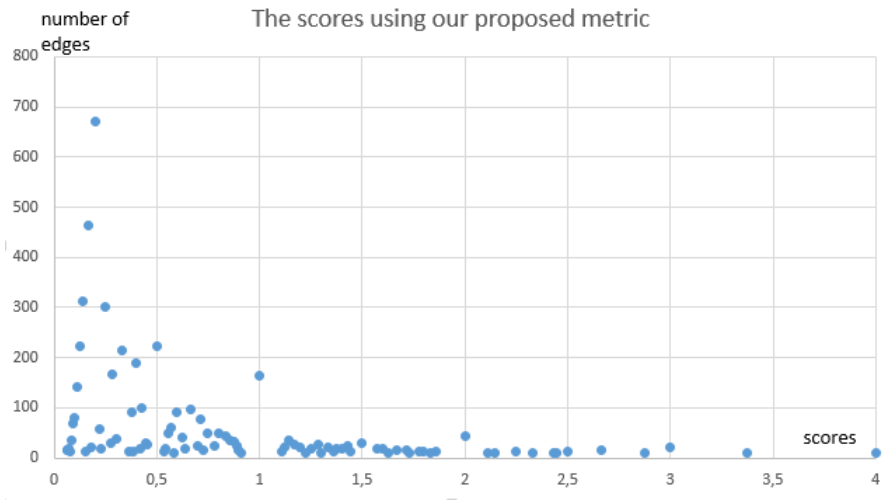


Figure 3.3: The proportion of scores in powergrid dataset after using the proposed algorithm.

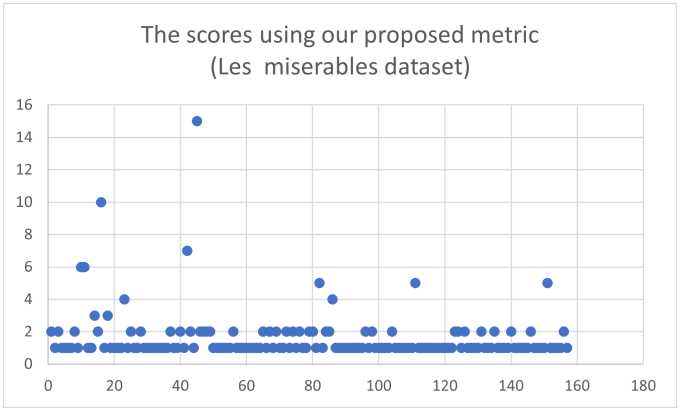


Figure 3.4: [The proportion of scores in Les miserable dataset after using the proposed algorithm.

3.2.4 Algorithm

The following algorithm explains the proposed methodology for evaluating a similarity-based link prediction metric.

Algorithm 1 Evaluation of link prediction method using the AUC

Require: Graph $G(V, E)$, algorithm
Ensure: AUC
Split our graph into $G_{train}(V, E_{train})$ *and* $G_{test}(V, E_{test})$ ▷ 10 is the number of indepndant comparisons
$N \leftarrow |E|/10$
$N' \leftarrow 0$
$N'' \leftarrow 0$
$AUC \leftarrow 0$
$i \leftarrow 0$
while $i < N$ **do**
$Edge1 \leftarrow$ *select Randomly an edge from* E_{train}
$Edge2 \leftarrow$ *select Randomly an edge from* E_{test} ▷ Edge1[0] is the first node and Edge1[1] is the second node
$Score1 \leftarrow S(Edge1[0], Edge1[1])$
$Score2 \leftarrow S(Edge2[0], Edge2[1])$
$If\ score1 > score2:$
$N' \leftarrow N' + 1$
$Else\ if\ score1 = score2:$
$N'' \leftarrow N'' + 1$
end while
$Return(N' + 0.5 * N'')/N$

This algorithm receives the graph $G(V, E)$ as input, then we split the graph into training graph G_{train} and test graph G_{test}. We consider N' to be the number of times the score of an edge from G_{test} is greater than that of an edge from $\bar{G}$. N'' is the number of times the score of an edge from G_{test} equals an edge from $\bar{G} = (V, \bar{E})$. As defined in previous chapter: $AUC = (N' + 0.5 * N'')/N$

3.3 Experimental results

3.3.1 Datasets

We have tested five datasets from various fields in order to confirm the accuracy of our measure against the state of the art metrics. The networks are: Facebook

[Leskovec and Mcauley, 2012], Enron [Leskovec et al., 2009, Klimt and Yang, 2004], Power Grid [Kunegis, 2013, Watts and Strogatz, 1998], Dolphins [Lusseau et al., 2003], Football [Girvan and Newman, 2002]. The description of the networks used is given in table 3.2, the features of the graphs (Refer to Chapter 1) are: the number of nodes. N, M is the number of edges, N_c is the size of the giant component and the number of components, E is the network efficiency [Latora and Marchiori, 2001], C is the clustering coefficient [Watts and Strogatz, 1998] (The nodes with degree 1 are excluded from the calculation) and r is the assortative coefficient [Newman, 2002] , D is the graph diameter, and H is the degree of heterogeneity [Snijders, 1981].

Table 3.2: Networks descriptions.

Networks	Enron	Grid	Facebook	Football	Dolphins
N	36692	4941	4039	115	62
M	183831	6594	88234	613	159
Nc	33696/1065	4941/1	4039/1	115/1	62/1
E	0,221	0,063	0,306	0,450	0,379
C	0,715	0,106	0,617	0,403	0,302
r	-0,11	0,003	0,06	0,162	-0,043
D	11	46	8	4	8
Averge Degree	10,0202	2,6691	43,6910	10,6609	5,1290
H	13,979	1,45	2,439	1,006	1,326

3.3.2 The influence of the path depth l

The effect of the path length l on the performance of the suggested approach is depicted in Figure3.5. We ran experiments on the five datasets using different path lengths from 0 to 10 with a step of 0.1; for every step, we calculated the AUC as an evaluation metric.

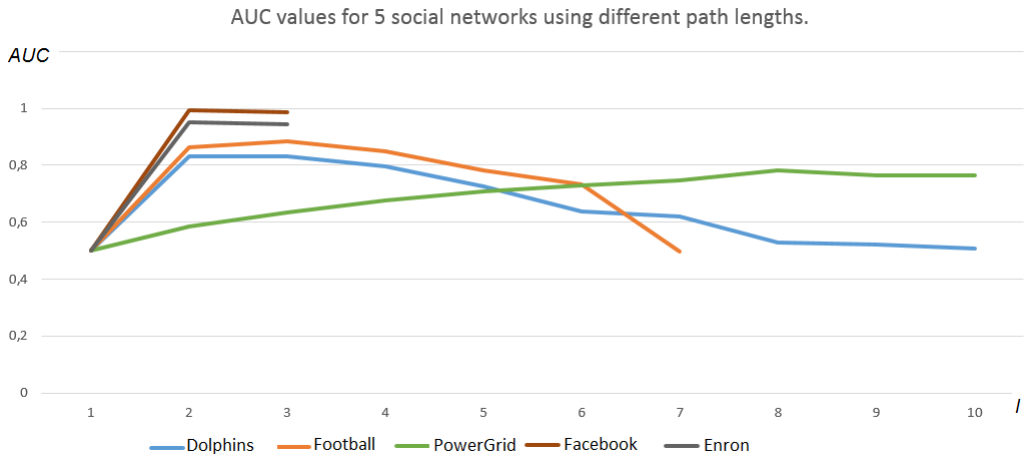


Figure 3.5: The outcomes of our proposed metric using different values of the parameter l in the range 0 to 10 on five datasets.

According to the results, our metric provides better results in terms of AUC using $l = 2$ and $l = 3$ for majority of the graphs except for the power grid, in which the optimal length was $l = 8$ and the value of $AUC = 0.781$.

Paths of length $l = 2$ offer information about the neighbors' neighbors (note, we are using simple paths), whereas paths of length $l = 3$ offer information about the neighbors three hops away. By dividing by the source and destination degrees sum, we reduce the importance of nodes with high degrees (nodes with many friends). In the real world, those vertices contribute little to friendship cycles, and pieces of information are unlikely to crosse them. Using path lengths $l = 2$ and $l = 3$ improves the suggested link prediction algorithm's accuracy. Furthermore,

when we reach the maximum of the AUC values, we could expect the algorithm's performance to remain stable, but experiments reveal that the AUC value decreases. In large networks like Facebook and Enron datasets, the number of links is significant, and due to the performance of the employed system (Intel core i5 2.67 GHz (4 CPUs) and 6 GB of RAM), we were unable to apply the suggested technique to these datasets for $l > 3$.

3.3.3 Comparison with the existing similarity metrics

Table 3.3: Results of the link prediction algorithms in terms of AUC. Bold text represents the maximum AUC for each dataset.

	Football	Dolphins	Grid	Enron	Facebook
CN	0,82	0,833	0,59	0,952	0,992
AP	0,271	0,696	0,446	0,874	0,826
Jaccard	0,833	0,836	0,59	0,947	0,991
RA	0,819	0,829	0,59	**0,953**	**0,994**
AA	0,819	0,829	0,59	**0,953**	0,993
PD 0,02	0,832	**0,84**	0,585	0,952	0,992
Sorensen	0,833	0,836	0,59	0,947	0,991
Hub promoted	0,831	0,816	0,582	0,947	0,986
LHN	0,832	0,82	0,584	0,943	0,958

Continued on next page

Table 3.3: Results of the link prediction algorithms in terms of AUC. Bold text represents the maximum AUC for each dataset. (Continued)

Hub depressed	0,83	0,836	0,59	0,947	0,99
Salton	0,833	0,826	0,584	0,948	0,992
Local path	0,83	**0,84**	0,637	ram prob	0,894
Katz	0,821	**0,84**	0,664	ram prob	0,486
The proposed metric	**0,884**	0,83	**0,78**	0,9476	0,987

The experimental findings in Table 3.3 demonstrated the suggested parameter-free method's efficiency and competitiveness. For the Power grid and Football datasets, it earns the highest AUC values. For the Dolphins network, the suggested method exceeds the AP and CN methods and achieves similar results as the other investigated link prediction methods. The AP and Katz perform worse on the Facebook network than the proposed technique. Except for the AP, all metrics show similar performance for the Enron network. Our proposed metric offers excellent results in all the datasets, unlike other measures where we find some metrics that offer great AUC values in one or two datasets, but their accuracy drop in the rest of the datasets. For example, RA has the best accuracy in the Facebook and Enron datasets, but its performance diminishes in the other three. We measured the correlation between each network feature and the AUC of our metric in Table 3.4; the findings indicate a strong correlation between the proposed metric

and the clustering coefficient, as well as a good correlation between the proposed metric AUC and the average degree. Consequently, as the clustering coefficient or average degree rises, so does the AUC of our metric.

Table 3.4: Correlation between the AUC of the proposed measure and networks features.

Networks	N	M	E	C	r	D	Averge Degree	H	AUC of our proposed metric
Enron	36692	183831	0,221	0,715	-0,11	11	10,0202	13,979	0,9476
Grid	4941	6594	0,063	0,106	0,003	46	2,6691	1,45	0,78
Facebook	4039	88234	0,306	0,617	0,06	8	43,691	2,439	0,987
Football	115	613	0,45	0,403	0,162	4	10,6609	1,006	0,884
Dolphins	62	159	0,379	0,302	-0,043	8	5,129	1,326	0,83
Correlation	0,411	0,721	0,319	**0,948**	0,0307	-0,656	**0,792**	0,469	

3.4 Link prediction using machine learning algorithms

Many studies have approved combining some link prediction criteria to improve the efficacy of the link prediction approach. The majority of them make use of machine learning algorithms. There are various categorization models in the supervised learning context [Suthaharan, 2016], such as random forest, decision trees, k-nearest neighbours, Support-vector machine, artificial neural network, and logistic regression. We investigate link prediction as a supervised learning task in this part. We use accuracy to compare different supervised learning systems' performance predicting ties.

3.4.1 Methodology

The fundamental objective of this section is to investigate the efficiency of the suggested measure when modeling the link prediction problem as a classification task. To this end, we have used accuracy as an evaluation criterion:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made} \tag{3.2}$$

Consider a simple graph $G(V, E)$. To transform the missing link problem into a binary task, we proceed as follows:

1. We generate two subsets: the first contains the edges of E, and the second consists of edges chosen randomly from $\overline{E}$; recall that $|E| = |\overline{E}|$.

2. Then we assign 1 value to edges of E and null value to the ties of $\overline{E}$; these are their true values.

3. After that, we shuffle the edges and split them into $X\ train$ and $X\ test$, $Y\ train$ and $Y\ test$ using the parameter $test\ size = 10\%$ using the train test technique.

4. Finally, we train a classifier like KNN, etc, on the $X\ train$, $Y\ train$ subset and predict the values for $X\ test$ data. The link prediction algorithm was applied using the Sklearn framework [Pedregosa et al., 2011b].

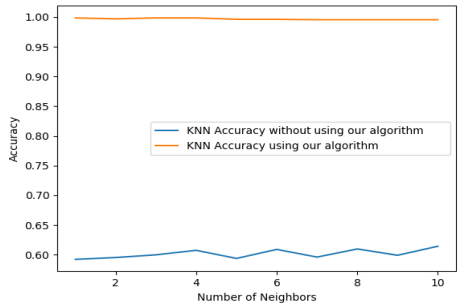
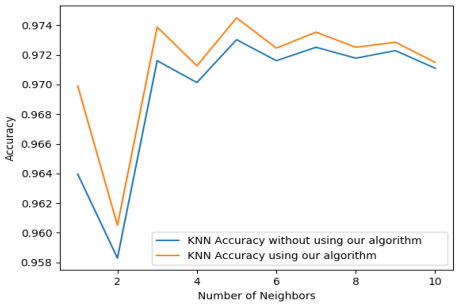
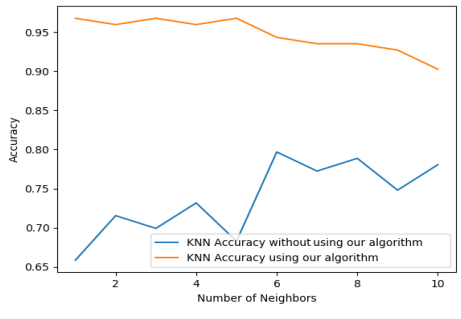
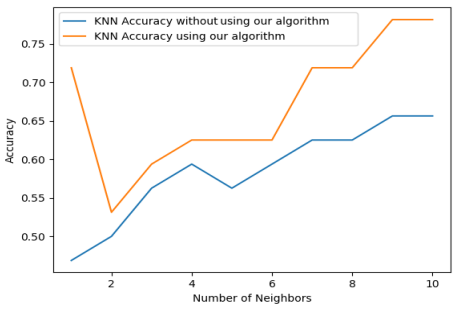
3.4.2 Results

We employ four datasets to analyze the accuracy of the proposed prediction approach: Grid, Facebook, Football, and Dolphins. We first deploy the KNN algorithm with the features [AA, AP, CN, HD, HP, JA, LHN, RA, Salton, Sorensen]. Second, as an extra feature, we employ the proposed parameter-free measure [AA,

AP, CN, HD, HP, JA, LHN, RA, Salton, Sorensen, *Ourmeasure*]. The results are shown in Table 3.5. We use different numbers of neighbors ranging from 1 to 10. The figures show that using our approach as an additional feature enhances link prediction accuracy. For the power grid dataset, the accuracy was increased by 40%.

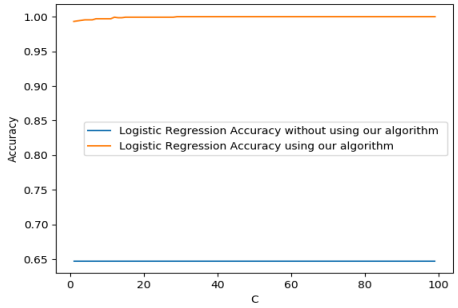
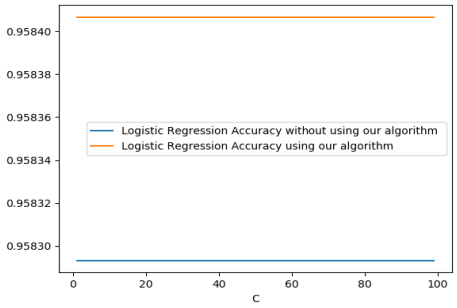
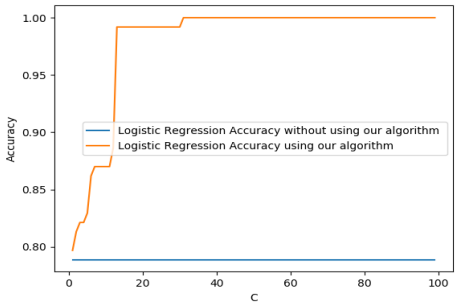
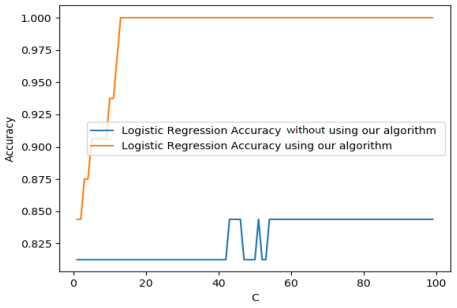
The outcomes of the KNN algorithm in both circumstances in Table 3.5; the blue curve shows the first scenario in which we used only state-of-the-art techniques, while the orange curve indicates the case in which we employed state-of-the-art algorithms and our proposed measure. We can conclude that the orange curve is always the most accurate.

Table 3.5: The outcomes of the KNN algorithm in both circumstances(with and without our algorithm).

Grid	Facebook
	
Football	**Dolphins**
	

We illustrates the findings of the logistic regression in Table 3.6. The blue curve illustrates the first situation in which we only apply state-of-the-art metrics. The orange curve depicts where we applied the proposed technique with state-of-the-art measurements. We employed logistic regression and varied values for the parameter c (Inverse of regularization strength) from 0 to 100, smaller numbers indicated more regularization [Pedregosa et al., 2011b]. As illustrated in Table 3.6, we can sum up that logistic regression algorithm accuracy improves if we use the parameter-free metric as an additional feature for all datasets.

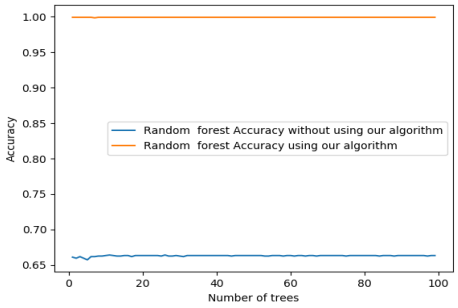
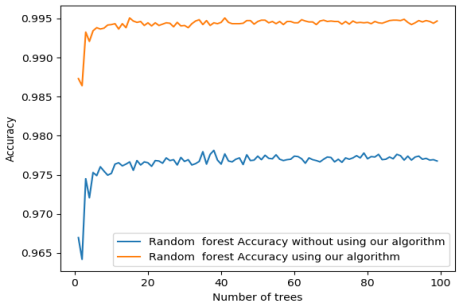
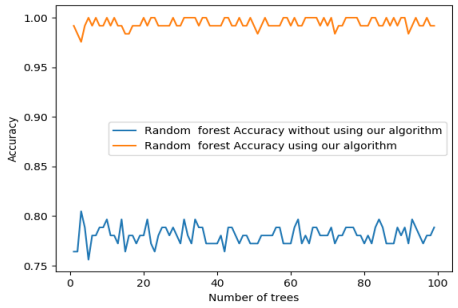
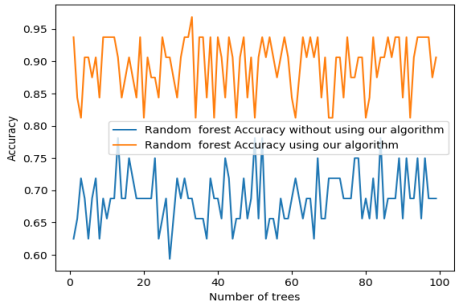
Table 3.6: The results of logistic regression algorithm

Grid	Facebook
	
Football	**Dolphins**
	

We applied random forest to classify an edge e as $e \in E$ or $e \in \bar{E}$. This algorithm reduces the over-fitting [Pedregosa et al., 2011b]. We turn the algorithm using several values of the number of trees in the range 1 to 100. From Table.3.7,

we can sum up that using the parameter-free metric enhances the accuracy.

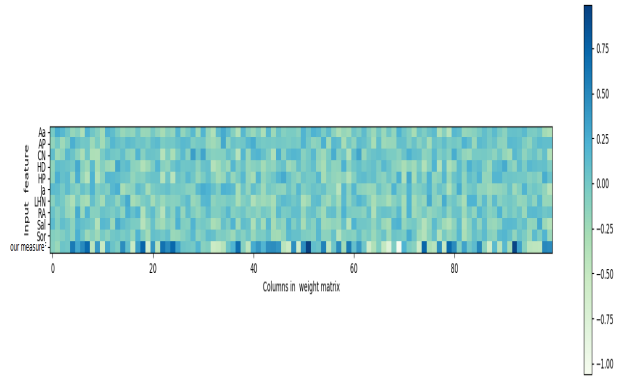
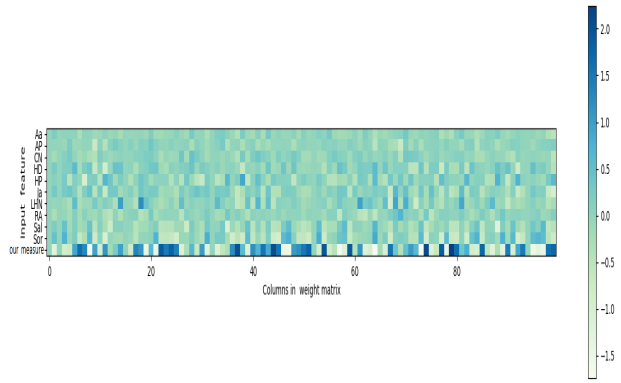
Table 3.7: Random forest algorithm results.

Grid	Facebook
	
Football	**Dolphins**
	

After applying a scaler (Standard scaler from Sklearn preprocessing methods) to the values of the training as well as the testing features, we run the random forest, neural network, and SVM algorithms. Table 3.8 describes how each method performed in the two scenarios examined (with and without the parameter-free metric). We only choose $l = 2$ because this value provides a high AUC. We use Sklearn's built-in function feature importances to investigate the significance of the used parameter for the random forest algorithm. The findings are shown in Table.3.9. We can see that our approach is crucial in the classification task; a similar conclusion can be derived from Table.3.8.

The neural network's input features are represented on the y-axis. The x-axis shows the neural network's layers (note that we have used 100 layers) Table 3.8. Each square's color denotes how much every feature contributed to the layer. Our proposed metric contribution has the darkest colors, making it the most significant contributor index compared to the other metrics.

Table 3.8: Neural network algorithm results.

Grid	
Facebook	

Continued on next page

Table 3.8: Neural network algorithm results. (Continued)

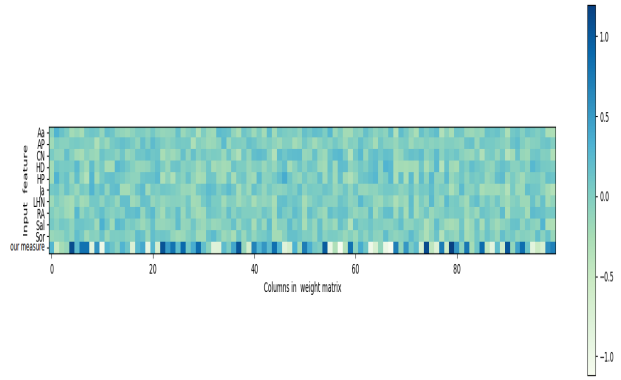
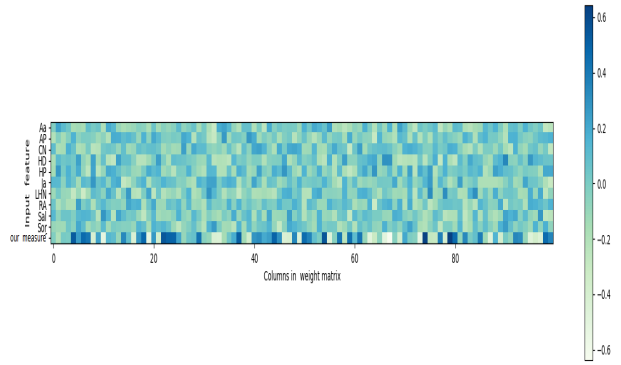
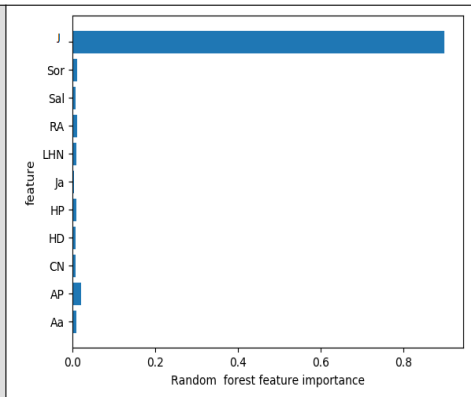
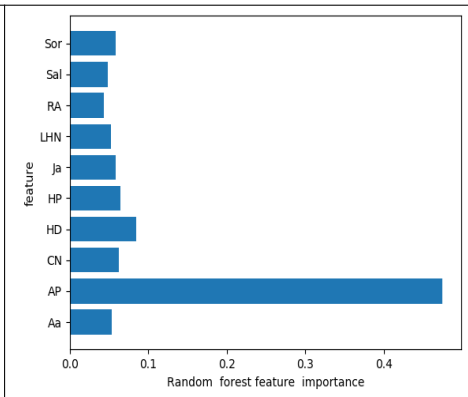
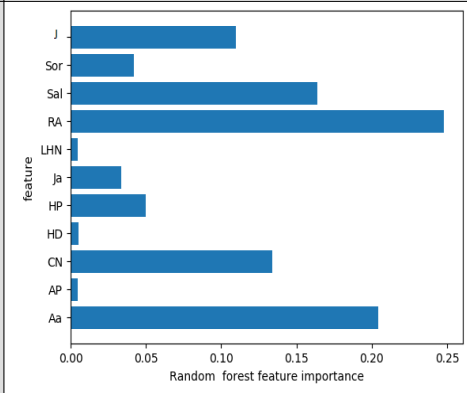
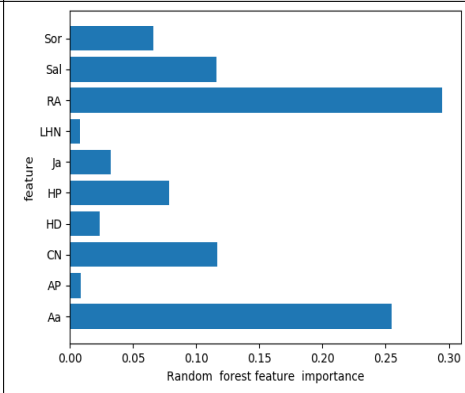
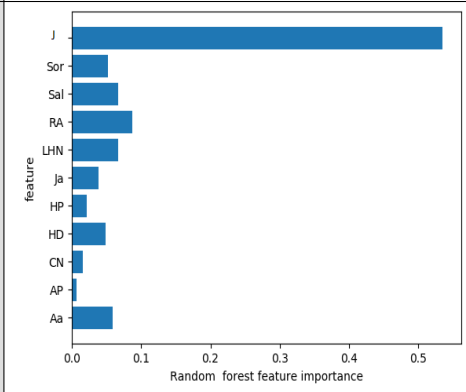
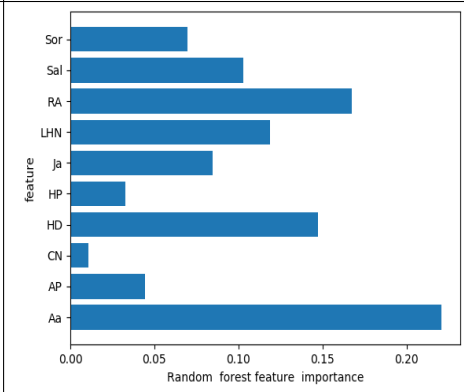
Football	
Dolphins	

Table 3.9: In both situations, we can see that the proposed metric's contribution to the decision has a favorable effect using the Random Forest method.

	The importance of features (our metric included)	The importance of features (our metric excluded)

Continued on next page

Table 3.9: In both situations, we can see that the proposed metric's contribution to the decision has a favorable effect using the Random Forest method. (Continued)

Grid		
Facebook		
Football		

Continued on next page

Table 3.9: In both situations, we can see that the proposed metric's contribution to the decision has a favorable effect using the Random Forest method. (Continued)

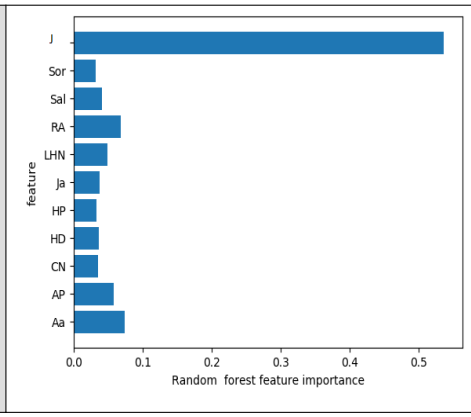
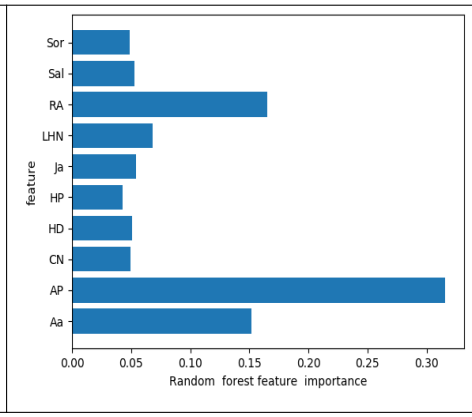
Dolphins		

Table 3.10: Efficiency of SVM, neural network and Decision Tree algorithms with and without using the proposed metric.

	Power grid	Facebook	Football	Dolphins
DT using J	0,999	0,993	0,992	0,938
DT without J	0,662	0,968	0,789	0,688
Neural Network with J	0,999	0,998	0,999	0,999
Neural Network without J	0,663	0,978	0,789	0,812
SVM usign J	0,999	0,983	0,805	0,844
SVM without J	0,625	0,972	0,797	0,812

According to the experimental findings, accuracy tends to rise whenever the parameter-free metric is added to the machine learning algorithm.

3.5 Conclusion

Because graphs or social networks are formed of billions of users/nodes and relationships/edges, link prediction remains a difficult task. We proposed a new parameter-free metric based on path depth and node degree in this chapter. We explored the effect of path length as a parameter and revealed that our method obtained good accuracy for path lengths $l = 2$ and $l = 3$ throughout most datasets. Then, using the AUC, we compared state-of-the-art link prediction methods to the proposed metric. The results demonstrated that the suggested method performs well for various real-world networks. Furthermore, we transformed the link prediction problem into a binary task and used machine learning approaches to address this problem. Using our proposed technique as an extra feature improves accuracy. Referring to the experimental results, the parameter-free proposed method based on path depth and node degree are effective. It has significant potential to solve large social networks' link prediction problems. Next, we will introduce another metric based on the mean received resources. The advantage of this metric relies on its ability to improve the state-of-the-art metrics.

Chapter

4 Mean received ressources meet machine learning algorithms to improve link prediction methods

4.1 Introduction

During the last two decades, social network analysis has received significant attention. These networks are dynamic, with new links appearing and disappearing. The challenge of inferring future links from the current state of the network is known as link prediction. We calculate user similarity using information from nodes and edges in this field. The more similar the users are, the more likely they will connect in the future. Similarity measures are essential in the link prediction area. Many scientists have created numerous measures; refer to chapter 2 for more details. To evaluate the accuracy of those measures, researchers use Area Under Curve (AUC). In the present chapter, we will introduce an innovative parametrized approach for increasing the AUC of link prediction measures from the state-of-the-art by incorporating them with Mean Received Resources (MRR). Results of the experiment indicate that the suggested technique's accuracy surpasses the best available measures. Furthermore, we employed machine learning methods refer to chapters 2 and 3 for more details to identify connections and validate the effectiveness of our proposed combination.

Multiple real-world complex systems are modeled as graphs, with vertices representing entities such as universities and people and edges or links representing the

interactions connecting the nodes. The essential feature of these graphs is their evolution, where links are created and eliminated over time. This later issue is studied in the Link Prediction field and is considered one of social network analysis's most critical challenges (SNA) challenges. It has numerous applications (refer to chapter 2). The authors have presented several solutions to this problem. The authors of [Liben-Nowell and Kleinberg, 2007] compared several similarity metrics using AUC; the results of this paper show that the Adamic Adar metric (refer to chapter 2) [Adamic and Adar, 2003] performs best in four of the five data sets, followed by Jaccard index [Jaccard, 1901];

In [Martinez et al., 2016], the authors have developed the adaptive degree penalization index (ADP). They have introduced a generalized formula for the existing degree penalization local link prediction method (note that this metric has a parameter-free) for the sake of the degree penalization level. Then, using logistic regression, they predicted the parameter's value. The authors of [Gu and Chen, 2016] used a classification algorithm to tackle the link prediction problem (refer to chapter 2 for more detail about machine learning and link prediction), used the accuracy as an objective function, and then transformed the problem into an optimization problem. The feature set contains state-of-the-art measures, with the class label $+1$ for existing connections and -1 for non-existing links. In the paper [Wang et al., 2021], the researchers have used the importance of neighborhood knowledge in link prediction. As a result, they propose to extract structural information from the input samples using a neighborhood neural encoder. The node feature subset and topology feature subset, as well as the social feature subset (following or followed) also the collaborative filtering for the voting feature subset, were the four different types of data that the authors gathered in [Han and Xu, 2016] before training machine learning algorithms like Random forest, Naive Bayes, SVM, and logistic regression classifiers.

In the paper [Matek and Zebec, 2016], the authors combine the Preferential Attachment index and Adamic/Adar metric to solve **LP** problem using weights for each used metric and used AUC to verify the validity of the algorithm. Results gotten shows good accuracy for GitHub dataset using the formula: $S_{comb}(x, y) = 0.7 * s_{PA}(x, y) + 0.3 * s_{AA}(x, y)$. The researchers in [Ahmad et al., 2020] had used the same reasoning when they suggested **C**ommon neighbor and **C**entrality based **P**arametrized **A**lgorithm (CCPA): $S_{xy} = \alpha \times (\Gamma(x) \cap \Gamma(y)) + (1 - \alpha) \times \frac{N}{d_{xy}}$, the parameter α is a free parameter $\alpha \in [0, 1]$. It is used to control the weight of common neighbors and centrality. The fraction $\frac{N}{d_{xy}}$ represents the closeness centrality between the nodes x and y, where N is the number of nodes in the network, d_{xy} is the shortest path between x and y.

As was previously shown, during the past few decades, there has been a massive increase in the number of research that offers accurate predictions of links. Scientists have suggested new measurements and compared their findings with state-of-the-art metrics using the area under the curve (AUC). However, a novel approach that enables LP metrics users to improve their findings' precision is still required. This chapter provides a new approach to the LP problem using a combination of state-of-the-art measurements and Mean Received resources (MRR). This technique is parametrized, giving the user/system complete control over the importance of the LP metric and the MRR. The primary objective is to increase the AUC of existing measurements and any alternative local metric that may be suggested. We showed that the suggested combination had a critical impact on the findings. The links were then classified using machine learning methods. When we include our suggested measure as an additional feature, machine learning findings reveal that models perform very well. Also, we have found that the best-performing algorithm is the decision tree.

To sum up, we have introduced a novel parametrized measure that gives the user/ system complete control over the index. It should be noted that the suggested measurement improves the performance of the state-of-the-art metrics. Then, we investigated the effect of each enhanced version of the link prediction on the utilizing parameter. We investigated the relationship between the parameter and network properties. Finally, we applied machine learning algorithms to validate the suggested method's efficiency.

4.2 The proposed metric

Inspired by the RA [Zhou et al., 2009] index, where the authors had exploited resource flow sent from a source node to a destination vertice via shared neighbors; In a simple graph $G(V, E)$, assume that node A has only one unit to be sent to node B. This task must be accomplished using node A's neighbors only. Node A distributes the resource over all its neighbors, and each neighbor will act the same till reaching node B. By accounting for the Mean Received Resources (MRR) from the source node via the shortest pathways, this measurement is enhanced to a global level. We utilize MRR as the second criterion to boost the accuracy of the link prediction metrics described previously in chapter 2; because neighborhood-based metrics are insufficient for capturing global relationships. The following definitions apply to the mean received resources:

$$MRR_{A,B} = \frac{1}{|Shortest\_Paths(A,B)|} \times \sum_{path \in Shortest\_Paths(A,B)} \left(\prod_{node \in path} \frac{1}{|\Gamma(node)|} \right) \tag{4.1}$$

Figure4.1 explains one shortcoming of neighbors-based metrics (for instance, resource allocation) and demonstrates the benefits of adding the mean received approach in the combination. On the one hand, if $|\Gamma(x) \cap \Gamma(y)| = 0$, then the resource allocation can not capture the interactions between nodes x and y. The MRR, on

the other hand, will use the shortest paths to recognize the interactions between x and y. The problem is illustrated in Figure4.1 with a simple graph of six nodes. The application of RA measure or any of the other neighbor-based metrics described in the preceding chapters provides the following result: $s_{X,Y} = 0$ (see Figure4.1). Nevertheless, the usage of MRR enables us the capturing the interactions between the two nodes x and y; there are three shortest paths:

1. **The circuit passing by the nodes A and B:** $s_1 = \frac{1}{|X|} \times \frac{1}{|A|} \times \frac{1}{|B|} \times \frac{1}{|Y|} = \frac{1}{3} \times \frac{1}{5} \times \frac{1}{4} \times \frac{1}{2}$
2. **The circuit passing by through the nodes C and B:** $s_2 = \frac{1}{|X|} \times \frac{1}{|C|} \times \frac{1}{|B|} \times \frac{1}{|Y|} = \frac{1}{3} \times \frac{1}{8} \times \frac{1}{4} \times \frac{1}{2}$
3. **The circuit passing by through the nodes C and D:** $s_3 = \frac{1}{|X|} \times \frac{1}{|C|} \times \frac{1}{|D|} \times \frac{1}{|Y|} = \frac{1}{3} \times \frac{1}{8} \times \frac{1}{3} \times \frac{1}{2}$

$MRR_{x,y} = \frac{1}{3} \times (s_1 + s_2 + s_3)$

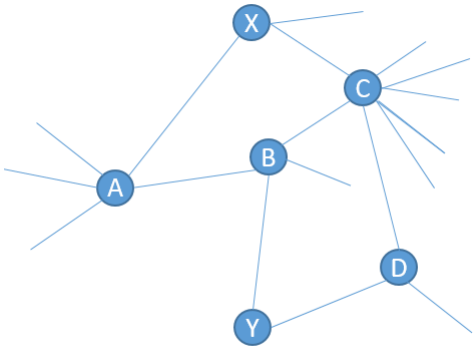


Figure 4.1: A simple undirected graph.

Each similarity metric captures a distinct piece of information; the assemblage of those information enables us to combine them into a single equation, which optimizes the classification task. Each local measure (LM) described in the previous

chapter is amplified by combining it with the equation 4.1 using the Weighted sum model (weighted combination [Fishburn, 1967]). The weighted combination is defined as follows:

$$PSI_{x,y} = \alpha \times sigmoid(LM(x,y)) + (1-\alpha) \times sigmoid(MRR(x,y)) \qquad (4.2)$$

The combination parameter α has the values in range $[0,1]$. This parameter regulates how much each component of the equation contributes. For some data sets, the MRR produces excellent results, while the LM results have superior prediction accuracy for others. Consequently, we employ α to modify the contribution of each piece. The MRR returns values between $[0,1]$, but other metrics like CN return values may be greater than 1. The local metric and MRR should be normalized as a result. We employ the sigmoid function to achieve this [Han and Moraga, 1995].

4.2.1 Data sets

In order to evaluate the proposed and state-of-the-art algorithms, real-world datasets are employed. We chose eight well-known real-world data sets (YeastS [Bu et al., 2003, Nakai and Kanehisa, 1991], Powergrid [Kunegis, 2013, Watts and Strogatz, 1998], USAir [Batagelj and Mrvar, 2006], Florida [Ulanowicz and DeAngelis, 2005], Football [Girvan and Newman, 2002], Political network [Kunegis, 2013], Les Miserables [Rossi and Ahmed, 2015], Zakary karate club [Kunegis, 2013]) to evaluate the precision of our approach. Remember that the metric will be better the closer the value of AUC near 1. Table 4.1 presents a summary of each data set:

Table 4.1: Network features.

Network	N	M	C	r	Average degree	D	H	e
YeastS	2284	6646	0,134	-0,099	5,819	4,37	2,84	0,233
Powergrid	4941	6594	0,08	0,003	2,669	18,989	1,45	0,063
USAir	332	2126	0,625	-0,207	12,807	2,738	3,463	0,406
Florida	128	2075	0,334	-0,111	32,421	1,776	1,237	0,622
Football	115	613	0,403	0,162	10,66	2,508	1,006	0,45
Political network	105	440	0,481	-0,132	8,38	3,092	1,41	0,396
Les miserables	77	253	0,559	-0,163	6,571	2,651	1,829	0,434
Zachary	34	77	0,485	-0,478	4,529	2,424	1,668	0,489

Table 4.1 express the characteristics of the graphs (refer to chapter 1 for more detail), the number of nodes N, M is the number of edges, e the efficiency of the network[Latora and Marchiori, 2001], C and r are the clustering coefficient [Watts and Strogatz, 1998] and the assortative coefficient [Newman, 2002] respectively (note that nodes with degree 1 are excluded from the calculation of the clustering coefficient), D is the diameter of the graph and H is the degree of heterogeneity [Snijders, 1981].

4.2.2 Results

90% of the network's size is employed as a training set, and the remaining 10% is utilized as a test set. To demonstrate the influence of the contributions of MRR and LM on the AUC values, we tested the various values of $\alpha \in [0, 1]$ with a step of 0.1.

Table 4.2: The impact of the parameter α on the AUC results

We employed equation 4.2 with different values of α, from $\alpha = 0$ when the MRR generates the scores of links to $\alpha = 1$ where the state-of-the-art measures define the scores of connections. The results are shown in the table 4.2, where the Y-axis represents AUC, and the X-axis represents the value of *alpha*.

We can conclude from Zachary's findings that the ideal pairing is MRR and RA for $\alpha = 0.6$. We can notice from the curve of the results that this pairing is superior to any other set of pairings. According to the Yeasts curves, MRR and AP

combination provides good results compared to (the best $\alpha = 0.2$). We can observe from the USAir measurements that, for $\alpha = 0.1$, the MRR and RA combination curve has little benefit over the MRR and AA combination curve. Results from the Power Grid demonstrate that all combinations deliver excellent results with slight variation, excluding CN, where the accuracy drops for $\alpha > 0$. We get the best accuracy value when $\alpha = 0$; this demonstrates that using MRR alone can produce promising outcomes for some datasets. The findings of Political Networks show that, in terms of AUC, both CN and MRR combination and PD and MRR combination offer the most significant results; the curves of other approaches, except for AP metric, are likewise highly comparable. The Les Miserables Dataset's curve demonstrates how efficiently the PD metric performs compared to other metrics. Football curves clearly show that the majority of combinations perform very similarly. Additionally, the test demonstrates that LHN and MRR is the ideal combination for this dataset. We may conclude from the results of the Florida dataset that RA and MRR together offer the best accuracy.

These tests proved that the only use of MRR or LM is not efficient in classifying the links; The shape of the curves increases to reach a maximum value of AUC and then decreases to reach the LM AUC values. As a result, we can find a value of α that maximizes AUC for each data set and outperforms the prediction accuracy of the state of art algorithms.

Next, we will evaluate our method (see equation 4.2) against the state of art algorithms on 8 datasets. Remember that we utilize the maximum AUC found in table 4.2.

Table 4.3: AUC results of both existing metrics and improved metrics

	USAir	Zachary	Yeasts	Football	Les miserab	Political network	Power Grid	Florida
AA	0,941	0,643	0,715	0,819	0,895	0,902	0,59	0,625
Imp. AA	0,947	0,843	0,754	0,89	0,938	0,925	0,733	0,624
jaccard	0,898	0,464	0,712	0,833	0,83	0,882	0,59	0,536
Imp. jaccard	0,911	0,629	0,751	0,905	0,884	0,923	0,733	0,549
AP	0,87	0,721	0,771	0,271	0,74	0,695	0,446	0,743
Imp. AP	0,756	0,714	0,794	0,864	0,834	0,852	0,7	0,501
Hub D	0,89	0,443	0,712	0,83	0,826	0,869	0,59	0,529
Imp. Hub D	0,9	0,629	0,751	0,908	0,876	0,916	0,733	0,546
Hub P	0,874	0,564	0,713	0,831	0,814	0,886	0,582	0,542
Imp. Hub P	0,876	0,714	0,751	0,905	0,868	0,916	0,733	0,553
LHN	0,778	0,45	0,711	0,832	0,787	0,848	0,584	0,4
Imp. LHN	0,785	0,614	0,751	0,91	0,828	0,909	0,733	0,431
PD	0,929	0,536	0,714	0,832	0,878	0,89	0,585	0,608
Imp. PD	0,942	0,721	0,753	0,902	0,934	0,932	0,733	0,622
RA	0,949	0,657	0,714	0,819	0,9	0,904	0,59	0,63
Imp. RA	0,955	0,857	0,754	0,89	0,934	0,925	0,733	0,631
CN	0,928	0,593	0,714	0,82	0,882	0,893	0,59	0,621
Imp. CN	0,941	0,721	0,753	0,902	0,93	0,932	0,733	0,622
Salton	0,905	0,479	0,712	0,833	0,834	0,886	0,584	0,539

Continued on next page

Table 4.3: AUC results of both existing metrics and improved metrics (Continued)

Imp. Salton	0,92	0,671	0,752	0,905	0,88	0,925	0,733	0,551
Sorensen	0,898	0,464	0,712	0,833	0,83	0,882	0,59	0,536
Imp. Sorensen	0,911	0,629	0,751	0,905	0,884	0,923	0,733	0,549

In the table 4.3, green cells represent the measures with the highest AUC values for each dataset, while red cells indicate the metrics with the lowest AUC values. The informations allows us to make the following deductions: The upgraded Jaccard metric provides a significant improvement in terms of accuracy. Additionally, the improved Jaccard's average AUC value outperforms the simple Jaccard's average value by 6.7 percent in terms of AUC.

The findings of AA demonstrate that, with the exception of the Florida dataset, the modified version has enhanced the algorithm's accuracy. The average AUC value across all datasets improved by 6.5 percent in terms of AUC.

The AUC of the simple AP for the Football dataset is 0.271, which is lower than pure chance; the upgraded version, however, reaches 0.864.

The upgraded version of CN performs better than the originale version for the Florida dataset, with an average improvement of 6.18 percent. In the Power Grid dataset, the efficiency reach 14 percent.

The same observations applies to the remaining metrics. Hub promoted witedness an improvement of 6.38 percent; LHN improvement was 7.13 percent; RA was increased by 6.4 percent; and Salton and Sorensen enhancements with 7.8 percent and 6.7 percent, respectively.

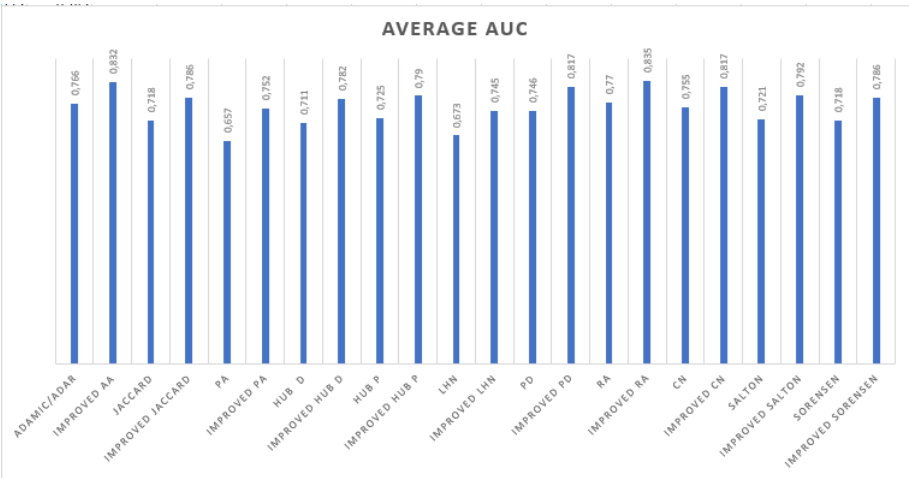


Figure 4.2: Average AUC of algorithms.

To obtain the average AUC throughout all datasets, we compute the mean of each row of the table 4.3 in Figure 4.2. This enables us to evaluate each algorithm's performance across all datasets. The results shown in Figure 4.2 offer robust evidence that the improved version has a higher average AUC than the original metrics. For instance, the improved AP is superior by 6.8%. This conclusion is valid for all other metrics.

The finding demonstrated that the suggested metric performs better than the present local measures. Additionally, it shows that any local metric's precision can be increased. That is to say, the scores given to the links in the E_{test} are higher than those given in $\bar{E}$. Then, the probability that an edge exists in the graph G(V, E) is high compared to an edge from $G(V, \bar{E})$. For instance, in the power grid dataset, LM has $AUC = 0.59$ while the suggested weighted combination achieved $AUC = 0.73$.

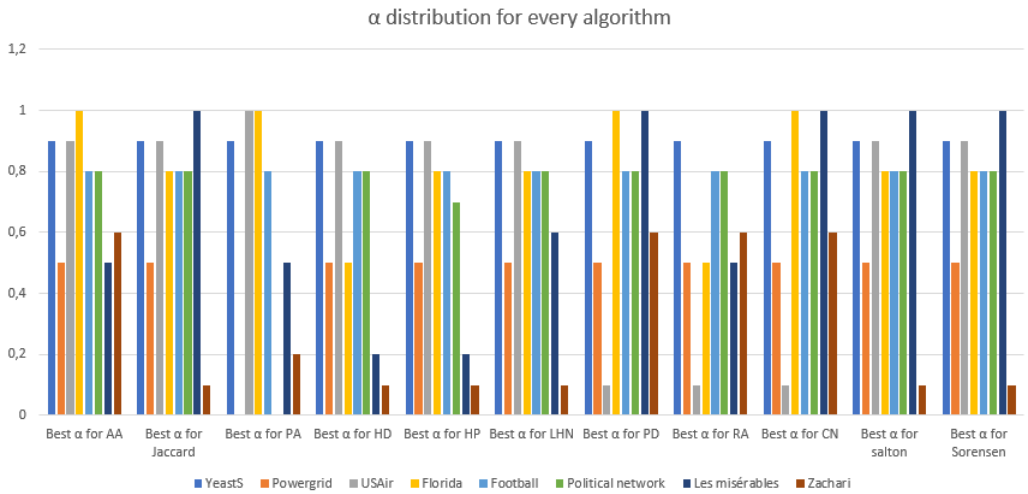


Figure 4.3: The α distribution on each dataset.

The Figure 4.3 indicates that the best $\alpha \in [0,5; 1]$. In addition, we can remark that all the metrics have the same *best* α for the same dataset. For instance, the *best* α is 0.9 for the YeastS network using all algorithms, 0.5 for all algorithms on Power Grid dataset. Next, we will establish a correlation between the *best* α and any network feature mentioned in table 4.1.

Table 4.4: Correlation between the best α of each algorithm and network features.

Correlation	N	M	C	r	Average Degree	D	H	e
Best α for AA	-0,347	0,044	0,034	0,161	0,695	-0,511	0,267	0,42
Best α for Jaccard	-0,175	0,055	0,143	0,521	0,285	-0,277	0,295	0,062
Best α for PA	-0,328	0,055	0,123	0,197	0,594	-0,509	0,431	0,41
Best α for HD	0,097	0,343	-0,15	0,541	0,098	-0,048	0,389	-0,234
Best α for HP	0,052	0,361	-0,209	0,555	0,426	-0,107	0,313	-0,047

Continued on next page

Table 4.4: Correlation between the best α of each algorithm and network features. (Continued)

Best α for LHN	-0,137	0,22	-0,032	0,6	0,394	-0,218	0,325	0,008
Best α for PD	-0,183	-0,12	-0,229	0,219	0,251	-0,277	-0,547	0,289
Best α for RA	0,113	0,1	-0,446	0,291	-0,237	-0,078	-0,439	-0,124
Best α for CN	-0,183	-0,12	-0,229	0,219	0,251	-0,277	-0,547	0,289
Best α for salton	-0,183	-0,12	-0,229	0,219	0,251	-0,277	-0,547	0,289
Best α for Sorensen	-0,499	0,055	0,143	0,521	0,285	-0,277	0,295	0,062

In table 4.4, we calculated the correlation between the best α and the network features. The results show that for AA and PA, *best alpha* is correlated to the average degree. Some other metrics (Jaccard, Hub Depressed, Hub promoted, LHN, and Sorensen) have a moderately strong correlation with r. We found a moderately strong correlation with heterogeneity for the measures CN, PD, and Salton. The RA index is the only metric that exhibits a moderately strong correlation with clustering coefficient C. Therefore, the parameter α can be written as a product of the network feature and a constant. For instance, $\alpha = average\_degree \times Constant$ for AA and PA.

4.3 Link prediction using machine learning algorithms

This section explores the link prediction problem as a classification problem using supervised learning methods. We use artificial neural network [Hkdh, 1999], random forest [Breiman, 2001], k-nearest neighbors [Goldberger et al., 2004], support vector machine (SVM) [Wu et al., 2004], and logistic regression [Kleinbaum et al., 2002]. After that, we evaluate the performance of the supervised learning algorithms by comparing their accuracy.

4.3.1 Methodology

We employ the classification accuracy in order to evaluate the suggested metric's performance when the link prediction is modeled as a classification task.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made} \tag{4.3}$$

Consider a simple graph $G(V, E)$. In order to convert the link prediction issue to a binary task, we construct two sub-sets:

1. The first one includes the links of E. we assign a 1 value to the links from E.
2. We assign a null value to the links from $\bar{E}$ and note that those edges from $\bar{E}$ are randomly selected.

We split the two sets into $test_{set}$ and $train_{set}$ where the $test\ size = 10\%$. Finally, we train our classifier on the train set and then predict the test set. In practice, we use Sklearn framework [Pedregosa et al., 2011a] to apply machine learning algorithms. The best α of RA was used in the learning in each dataset.

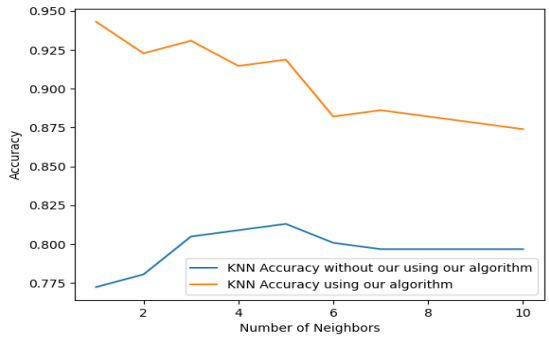
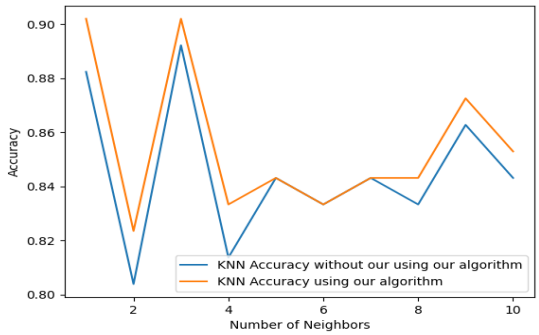
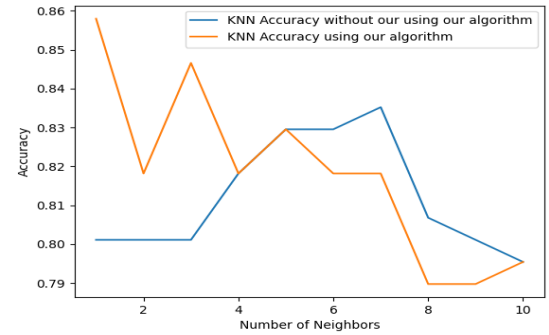
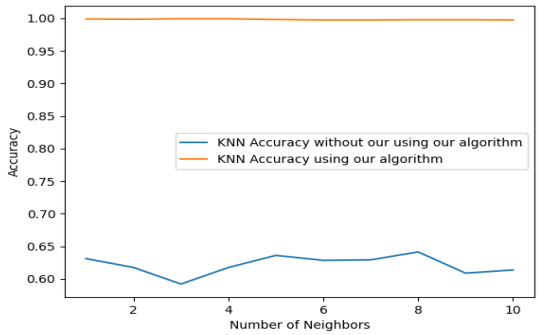
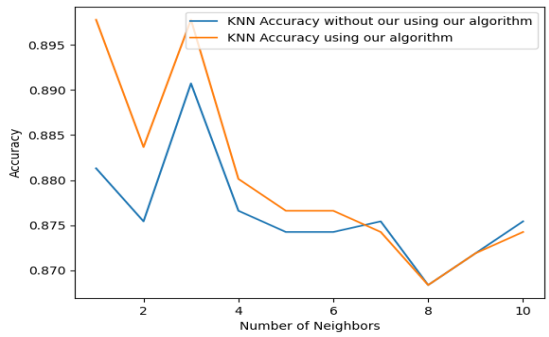
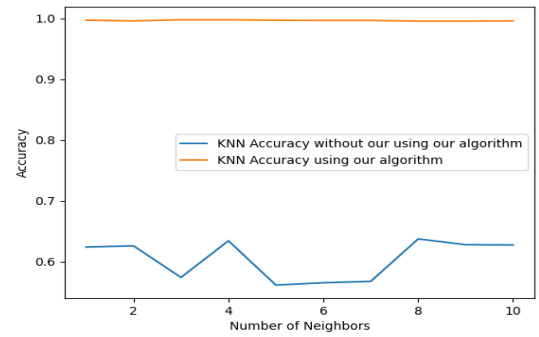
4.3.2 Results

Table 4.5: The results of KNN algorithm using MRR and without MRR

Football dataset	Les miserables dataset

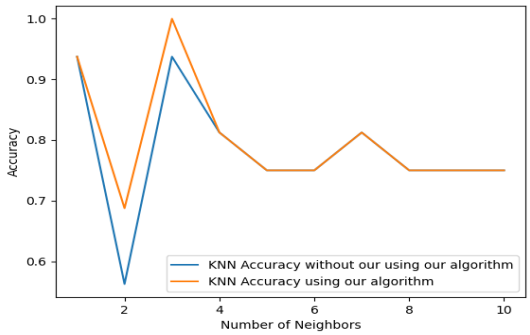
Continued on next page

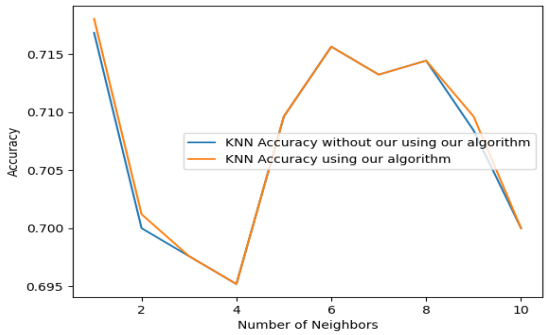
Table 4.5: The results of KNN algorithm using MRR and without MRR (Continued)

	
Political network dataset	Power Grid dataset
	
USAir dataset	YeastS dataset
	
Zachari dataset	Forida dataset

Continued on next page

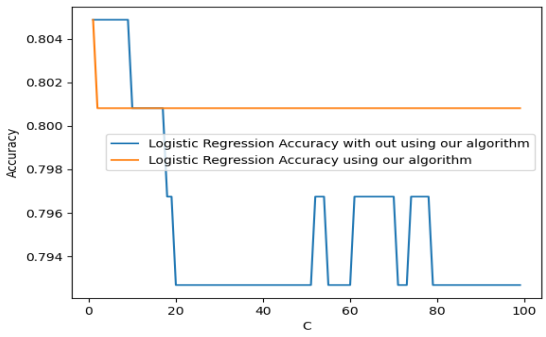
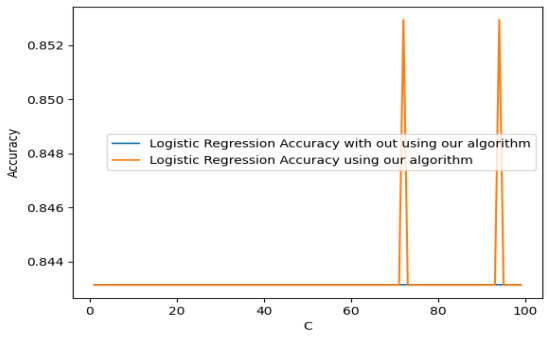
Table 4.5: The results of KNN algorithm using MRR and without MRR (Continued)





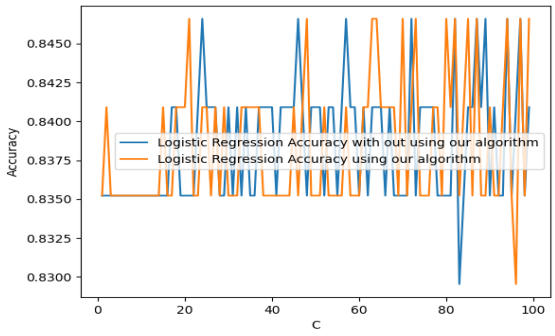
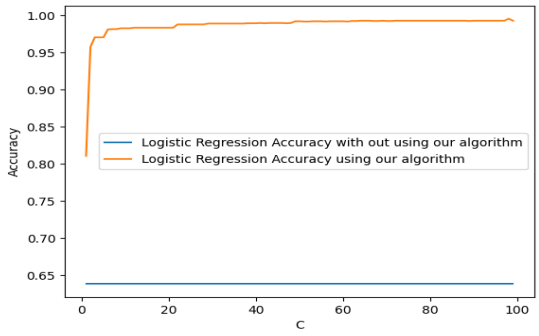
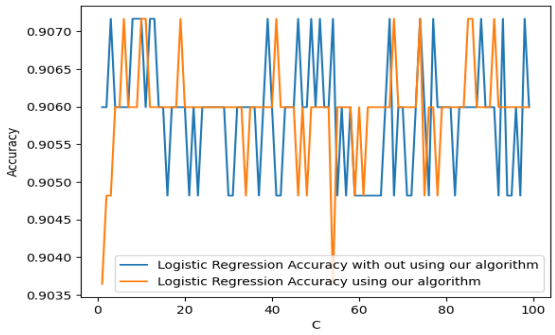
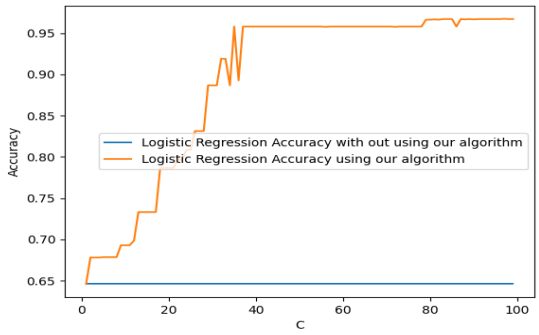
The outcome of the KNN algorithm in two situations is shown in the table 4.5, the accuracy is depicted on the y-axis, while the x-axis indicates the number of neighbors. The blue curve illustrates the first instance where we employ only the state of art algorithms. The orange curve describes the scenario when we combine the proposed metric with state-of-the-art algorithms. Because the orange curve consistently has the highest accuracy, we may infer that adding the PSI (equation 4.2) as an additional feature makes the classification process more accurate.

Table 4.6: The inverse of regularization strength is shown on the x-axis. The accuracy of the logistic regression model is displayed on the y-axis.

Football datast	Les miserables dataset
	

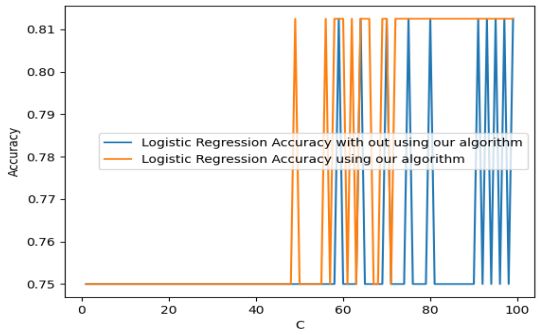
Continued on next page

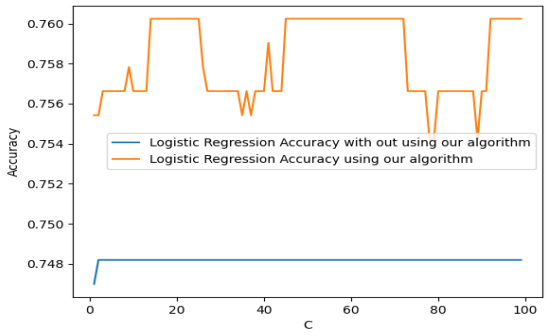
Table 4.6: The inverse of regularization strength is shown on the x-axis. The accuracy of the logistic regression model is displayed on the y-axis. (Continued)

Political network dataset	Power Grid dataset
	
USAir dataset	**YeastS dataset**
	
Zachari dataset	**Forida dataset**

Continued on next page

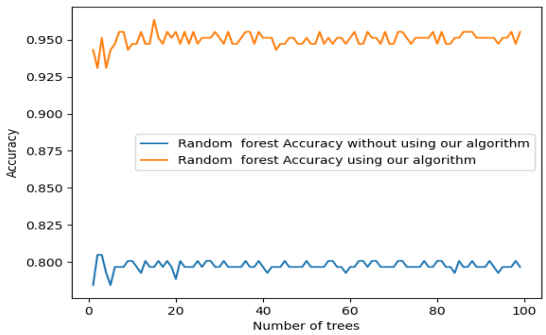
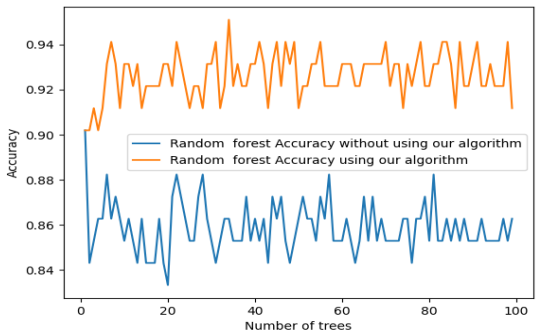
Table 4.6: The inverse of regularization strength is shown on the x-axis. The accuracy of the logistic regression model is displayed on the y-axis. (Continued)





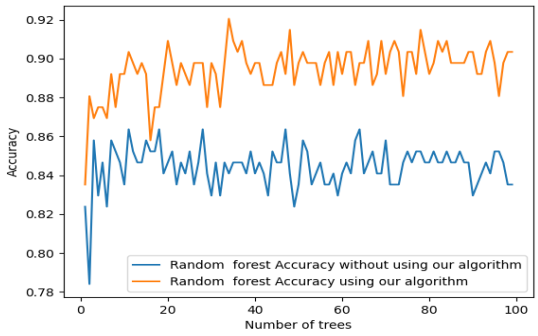
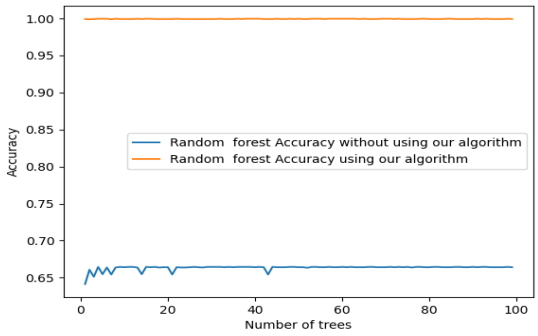
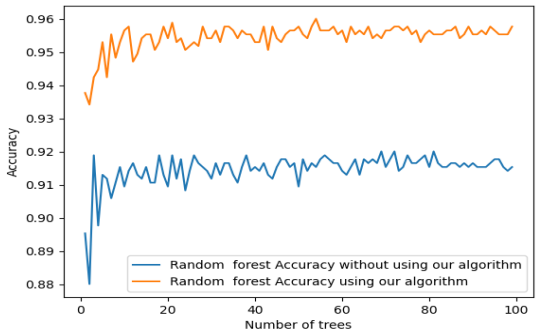
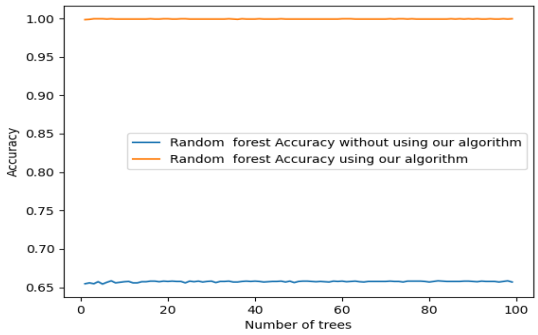
The outcomes of logistic regression are displayed in the table 4.6. When we combine the state-of-the-art techniques for logistic regression with the PSI metric (equation 4.2), we get higher accuracy. The blue curve represents the scenario where we only use link prediction algorithms. We could not observe a clear advantage because the logistic regression did not converge for the political, USAir, and Zachary datasets.

Table 4.7: The x-axis represents the number of trees. The y-axis represents the accuracy of random forest model.

Football datast	Les miserables dataset
	

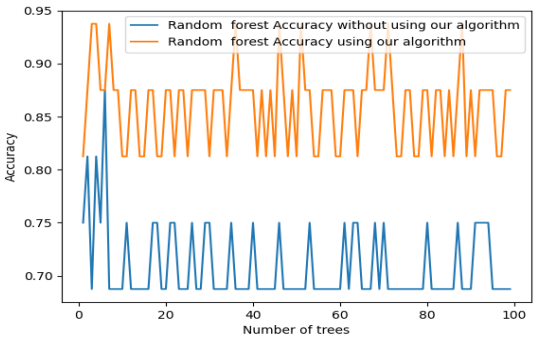
Continued on next page

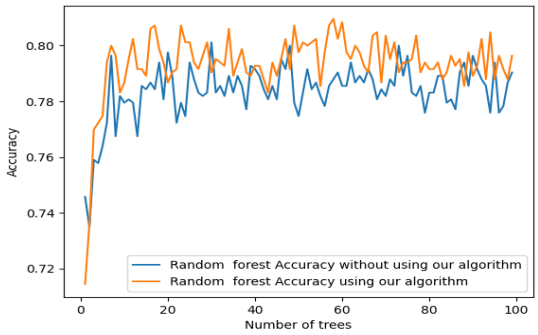
Table 4.7: The x-axis represents the number of trees. The y-axis represents the accuracy of random forest model. (Continued)

Political network dataset	Power Grid dataset
	
USAir dataset	**YeastS dataset**
	
Zachari dataset	**Forida dataset**

Continued on next page

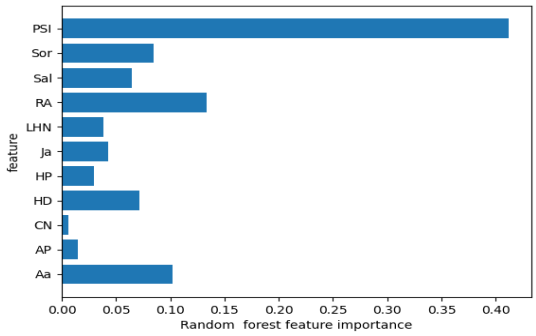
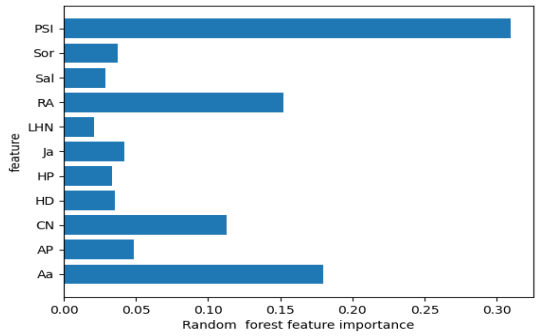
Table 4.7: The x-axis represents the number of trees. The y-axis represents the accuracy of random forest model. (Continued)





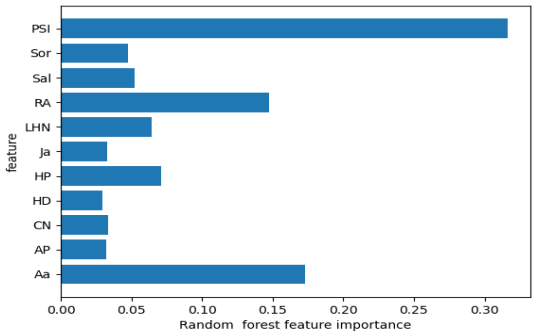
The outcomes of the random forest algorithm in two examples are displayed in Table 4.7. When we exclusively apply state-of-the-art algorithms, the curve is represented in blue. The curve in orange shows the second scenario when we combine the proposed metric with state-of-the-art algorithms. The orange curve and blue indicates that our metric makes the decision more accurate.

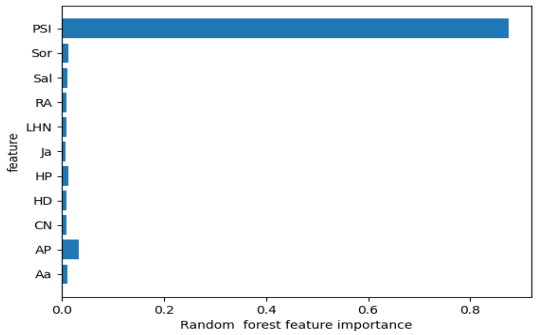
Table 4.8: Random forest features importances.

Footbal datast	Les miserables dataset
	
Political network dataset	Power Grid dataset

Continued on next page

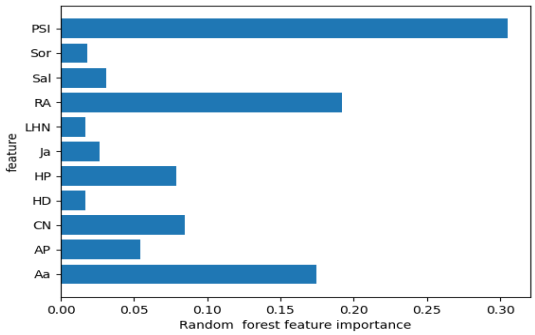
Table 4.8: Random forest features importances. (Continued)

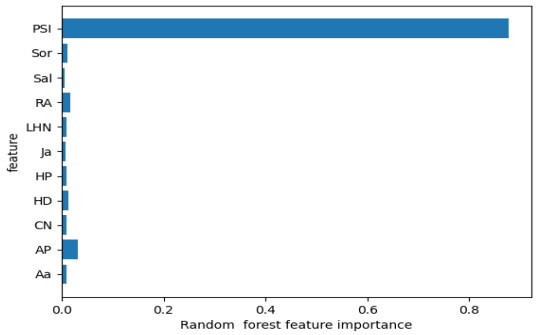




USAir dataset

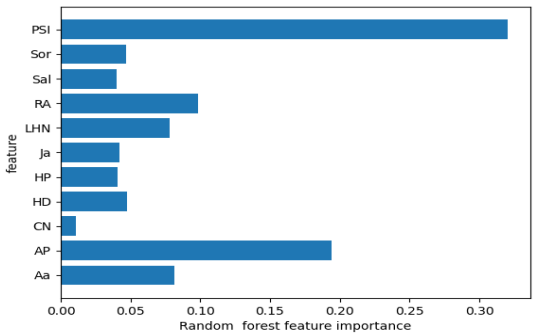
YeastS dataset

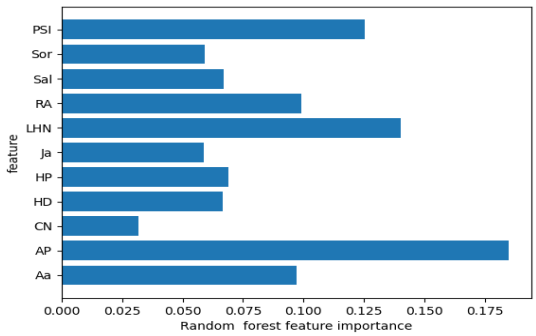




Zachary dataset

Forida dataset

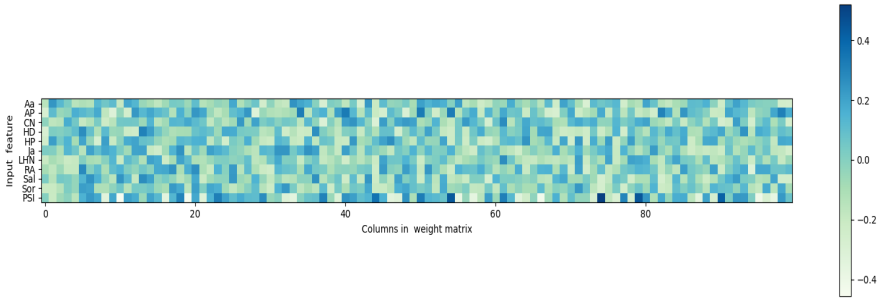
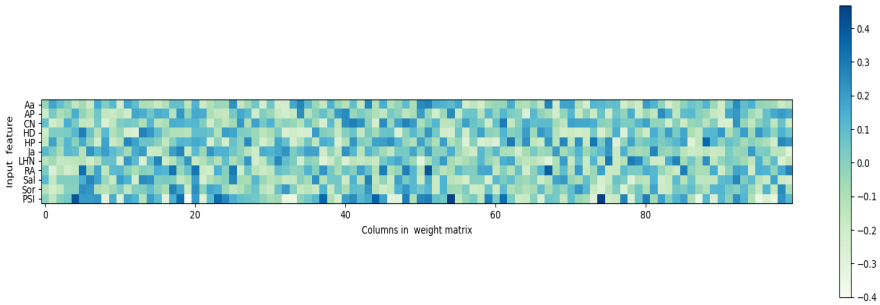
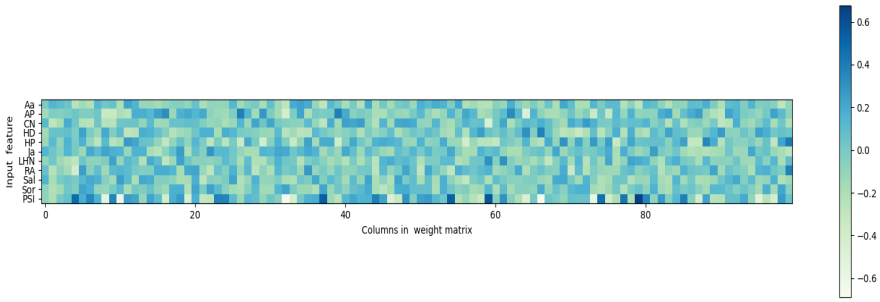




The significance of every feature used in the random forest technique (see table 4.7) is displayed in table 4.8. We can confirm that the suggested metric (equation

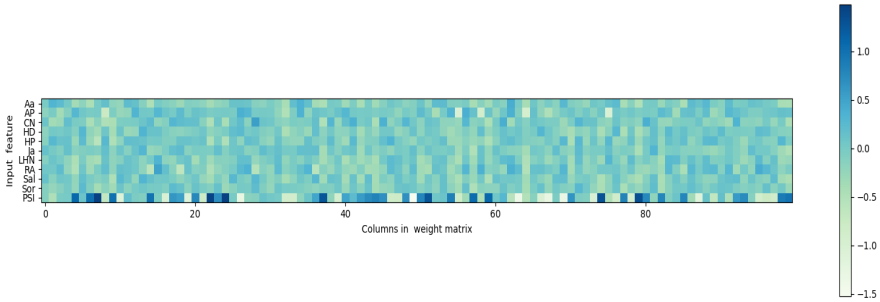
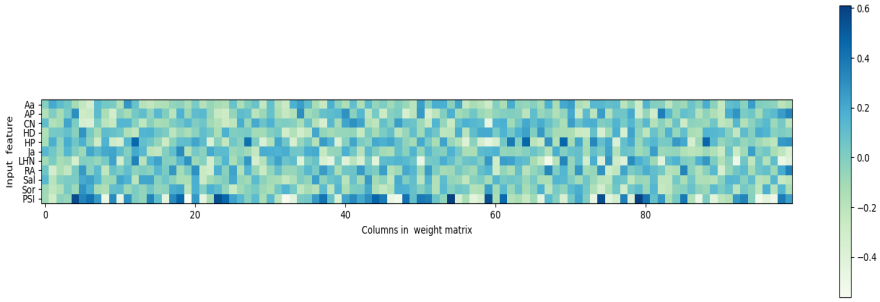
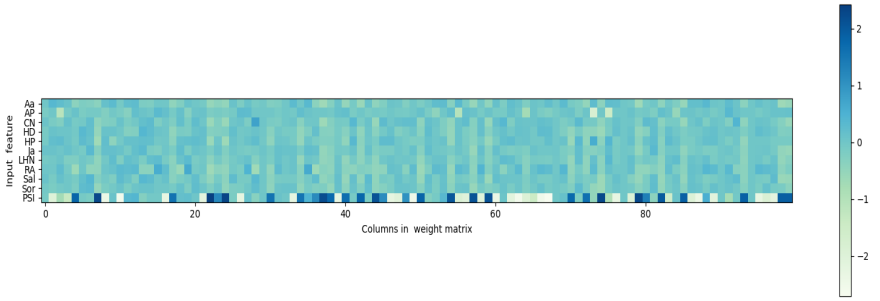
4.2) adds more to the decision-making process than the other metrics.

Table 4.9: Results of neural network.

	Football dataset
	Les miserables dataset
	Political network dataset

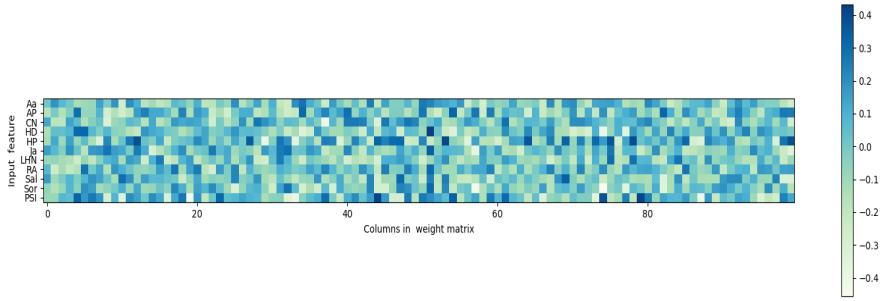
Continued on next page

Table 4.9: Results of neural network. (Continued)

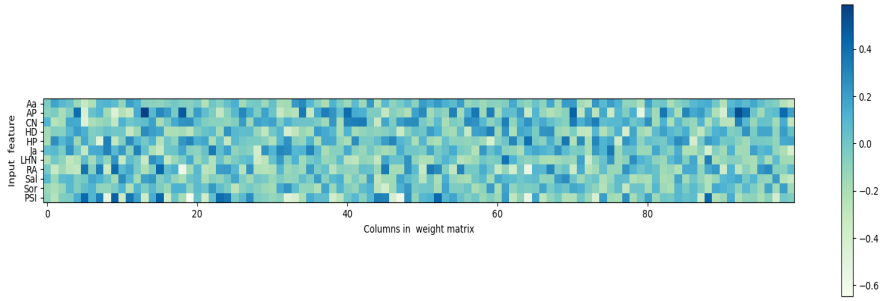
	
Power Grid dataset	
	
USAir dataset	
	
YeastS dataset	

Continued on next page

Table 4.9: Results of neural network. (Continued)



Zachari dataset



Forida dataset

The weights of each hidden unit are represented in the table 4.9. Large positive values are represented by blue areas, whereas white areas represent negative values. If the color of the cells is darker for any dataset, it indicates that the process values the utilized parameter highly. We can see that our suggested metric has the darkest row for most datasets. As a result, it influences how decisions are made.

Table 4.10: Results of support vector machine (SVM), neural network and decision tree.

	SVM with PSI	SVM without PSI	Neural network with PSI	Neural network without PSI	Decision tree with PSI	Decision tree without PSI
Les miserables	0.902	0.892	0.922	0.912	0.971	0.873
Political network	0.858	0.852	0.862	0.852	0.875	0.841
Football	0.756	0.756	0.768	0.760	0.959	0.760
USAir	0.942	0.892	0.968	0.913	0.969	0.885
Power Grid	0.995	0.638	0.999	0.643	0.998	0.641
Zachary	0.677	0.677	0.774	0.710	0.903	0.742
Florida	0.763	0.760	0.778	0.761	0.740	0.761
Yeasts	0.808	0.801	0.834	0.819	0.985	0.806

According to the tables above, PSI (see equation 4.2) improves KNN accuracy, particularly for the football, power grid, and YeastS datasets. The power grid, football, yeast, and Florida datasets showed the best results using the logistic regression approach. We notice that PSI improved performance across all datasets for the random forest model. The same result applies to neural networks. On all of the datasets, decision trees had the best accuracy overall. The outcomes demonstrate that adding the PSI metric as an additional feature has improved performance for all datasets.

4.4 Conclusion

A novel parametrized measure for link prediction in social networks is presented in this chapter. We build the suggested metric by utilizing local similarity measures (LM) from the state-of-the-art and mean received resources (MRR). Because this measure is parametrized, the system or user can change the weighting of the component being taken into account. Using the AUC value on eight datasets from various fields, we evaluate the effectiveness of the suggested metric and compaire the results with 11 other metrics from the state of the art.

The results of this chapter demonstrate that the suggested measure outperforms local similarity metrics on all datasets. Even if two nodes do not share any common neighbor, our proposed index can capture the potential interaction. Additionally, we have discovered correlations between various network properties and the parameter α. Furthermore, we have classified edges using machine learning methods. The results demonstrate that the accuracy of any algorithm increases anytime the proposed index is used as an additional feature. We also concluded that the decision tree method performs best in accuracy.

This research can be applied to other graphs, including directed and weighted networks. Additionally, dealing with imbalanced datasets in link prediction could be a massive problem because most real-world networks are highly sparse.

.

Chapter

5 Link prediction using betweenness centrality and graph neural networks

5.1 Introduction

Connecting people and sharing information are the major roles of social networks. Researchers have studied several problems in complex networks, like recognizing communities, identifying the role of users, or trying to infer the upcoming relations. The latter problem is known as link prediction in social networks. The main objective of the link prediction (LP) is to identify the upcoming link between pairs of nodes from the current state of the network. LP is used in a variety of fields, including network reconstruction [Wu et al., 2019], recommender systems [Chen et al., 2005], network completion [Hanneke and Xing, 2009], and spam mail detection [Karim et al., 2019].

In this chapter, we solve the link prediction problem using graph theory and machine learning. $G(V, E)$ is a graph that illustrates the network N, V is a finite set of nodes or vertices that represents the network users, and E is a finite set of edges that models the connections between users. We aim to identify the ties of time $t + 1$ from the edges of time t.

Previous researchers have proposed new metrics that provide good accuracy for a particular graph [Hasan and Zaki, 2011, Kumar et al., 2020, Liben-Nowell and Kleinberg, 2007]. Other works have studied the link prediction problem as

a machine learning problem [Al Hasan et al., 2006] where features are metrics or nodes attributes and the label of edges is binary. In their work [Wu et al., 2019], the authors have proposed a model based on the OWA operator (Ordered weighted averaging aggregation operator) where the network is rebuilt using two methods: a matrix completion method called "Low Rank" and an OWA operator-based ensemble approach. The authors in [Aziz et al., 2020] have suggested a new prediction approach that incorporates knowledge about nodes along local paths to improve the accuracy of the existing path-based methods. In [Backstrom and Leskovec, 2011], the authors have created a supervised Random Walks-based algorithm that naturally combines network structure information with node and edge level attributes. They accomplished this by guiding a random walk on the graph with these attributes. They designed a supervised learning task intending to learn a feature that assigns strengths to network edges so that a random walker is likelier to visit nodes that enable new connections creation. The authors in [Gu and Chen, 2016] have proposed a precision optimization-based algorithm. It has treated link prediction as an objective function in the process. As a result, LP is considered an optimization problem. A set of topological features is defined for each ordered pair of nodes. Link prediction can be treated as a binary classification using those features as node pair attributes, with the class label of each node pair determined by whether the node pair exist or not. Then, precision optimization can be used to solve the binary classification problem. In [Aghabozorgi and Khayyambashi, 2018], the authors have proposed a new similarity measure in which structural units are used as the source of similarity estimation. This similarity measure is evaluated in a supervised learning experiment framework, in which it is compared to the existing ones.

Although these sophisticated methods give more accurate results, they do not produce the same accuracy for every type of network and may require a high

computation time and memory. Therefore, this work focuses on local similarity-based methods that can deal with large networks in a reasonable computational time. In addition to the ease of using the local similarity metrics, they are also suitable for large-scale networks. However, the existing methods still suffer from a lack of accuracy. Improving this latter is still a challenging task. In this chapter, we propose a novel approach based on combining local similarity-based methods and node attributes. We use the betweenness centrality, which captures the number of shortest paths passing through a node and its role in sharing information in a network. Then, if two unliked nodes have received a huge amount of data, they will more likely form a link in the future. We also use a damping factor to control the impact of a similarity index. We call our method local similarity and betweenness centrality (LSBC). Our theoretical and experimental parts show that the proposed combination produces more accurate results and improves the local metrics' performance. In addition, this method was designed for any small, mid-size and large-scale networks.

Moreover, we consider link prediction as a classification task. As it is known, the graph neural network (GNN) becomes an important tool to learn over a graph [Niepert et al., 2016], [Bruna et al., 2013]. In this context, link prediction uses the GNN and node features to classify links [Li et al., 2020], [Zhang et al., 2021]. Both of these tools give high performances in link prediction. We use the artificial neural network to classify links in supervised learning. The experiments confirm that the use of the proposed metric as an additional feature improves the classification accuracy, and the cross-entropy loss is remarkably minor.

5.2 The proposed LSBC method

Local similarity-based metrics are very easy to compute and can be used for large-scale networks. Nevertheless, they do not produce a similar accuracy for each

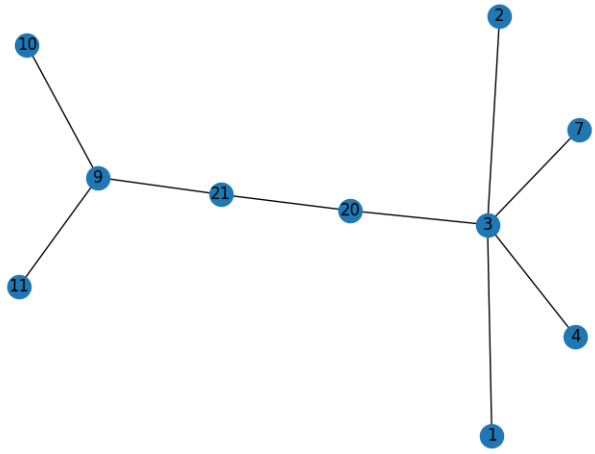


Figure 5.1: Example of social network

network. Furthermore, the use of some additional properties enhances their performance. So, we can obtain a metric dealing with large-scale networks and give more accurate results in a reasonable computational time. For those reasons, we base the proposed metric on two fundamental properties: a local similarity index and the betweenness centrality. This latter calculates the node's power over the entire network [Newman, 2005]. The idea behind the betweenness centrality is as follows: if several shortest paths cross a node x, the amount of information it receives increases; as a result of that, if both nodes have several pieces of information, they are more likely to be linked to boosting their amount of data. The combined local similarity and betweenness centrality method (LSBC) is defined as:

$$S(x,y) = \alpha \times SI(x,y)^{1-\alpha} + sigmoid(b(x) + b(y)) \tag{5.1}$$

Given a pair of nodes x and y, $SI(x,y)$ is a local similarity index. $b(x)$ is the betweenness centrality of the node x. α is a dumping parameter between $[0;1]$. It is used to control the contribution of the local similarity.

To illustrate the theoretical contribution of the proposed metric, we consider the network illustrated in Figure 5.1. We consider the two unliked nodes 3 and

9, and the scores of all common neighbors based metrics between these two nodes is $SI(3,9) = 0$. However, using the proposed metric, the obtained score is about 0.75 with a dumping parameter $\alpha = 0.5$. For example, considering the CN and AA metrics, the similarity scores are given by $CN(3,9) = AA(3,9) = 0$ while when the proposed combination is used, we obtain $LSBC_{CN}(3,9) = 0.7574$ and $LSBC_{AA}(3,9) = 0.7574$. The nodes 3 and 9 do not have common neighbors, but they receive a huge amount of information through the shortest paths and can be good spreaders. As a result, both nodes can be linked in the future. We can presume that the proposed combination can further produce more accurate results by taking into account the local similarity metric and the node's influence. The parameter α is used to control the importance of the local similarity-based metric compared to the betweenness centrality.

5.2.1 Data sets

To evaluate the proposed LSBC method, we use various networks from different fields. Note that we are using only simple graphs. The used datasets are described as follows:

1. Jazz [Rossi and Ahmed, 2015]: Every hub is a Jazz artist, and an edge signifies that two artists have played together in a band. The information was gathered in 2003.

2. Dolphins [Rossi and Ahmed, 2015]: An informal group of bottlenose dolphins. The collection includes a list of all connections, where a link addresses a regular association between dolphins.

3. C-Elegans [Rossi and Ahmed, 2015]: The neural network of the nematode worm C.elegans, in which an edge connects two neurons linked by a synapse or a gap junction.

4. Karate [Rossi and Ahmed, 2015]: Wayne Zachary collected the data on social ties among members of a university karate club in 1977.

5. Macaque [Négyessy et al., 2006]: Cortical connectomes are modeled as nodes and links.

6. USAir97 [Rossi and Ahmed, 2015]: The network of the US air transportation system in 1997. Nodes represent the American airports, and the airlines between airports illustrate the edges.

7. NetScience [Rossi and Ahmed, 2015]: A network of coauthorships between scientists who are publishing on the topic of network science.

8. Les Miserables [Rossi and Ahmed, 2015]: The network of appearances of characters in the novel "Les Miserables."

Table 5.1: The topological features of the benchmark networks.

	Jazz	Dolphins	C-elegans	Karate	Macaque	USAir97	NetScienc	Les miserables
e	0,513	0,379	0,449	0,492	0,5	0,406	0,016	0,434
$\|V\|$	198	62	131	34	242	332	1461	77
$\|E\|$	2742	159	687	78	3054	2126	2742	253
C	0,617	0,258	0,2451	0,5706	0,45	0,6252	0,693	0,559
r	0,02	-0,0435	0,0218	-0,475	-0,0547	-0,2078	0,4616	-0,163
k	27,69	5,129	10,488	4,588	25,239	12,807	4,823	6,571
d	2,235	3,356	2,5234	2,408	2,2174	2,7381	6,04	2,651
H	1,39	1,326	1,296	1,693	1,531	3,463	1,84	1,829

Table 5.1 shows the basic topological features of the used real world benchmark networks, where e is the eccentricity. $|V|$ and $|E|$ denote the number of nodes and edges, respectively. The average degree is k, the shortest average distance is d, the clustering coefficient is C, the assortativity coefficient is r, and the degree heterogeneity is H, which is defined as $H = \frac{<k>^2}{<k^2>}$.

5.2.2 Results

In order to show the effectiveness of the proposed LSBC combination method, we conduct experimentation of traditional local similarity based methods and their improved version on the dataset previously introduced. We use the area under curve as an evaluation criterion.

In table 5.2, we have compared the original similarity index with the improved LSBC version introduced in section 3.2. Red cells refer to the lowest AUC values obtained in both original and improved algorithms. In contrast, the green cells refer to the highest AUC values obtained in both original and improved algorithms. The result shows that the enhanced LSBC metric gets the highest AUC values in all datasets because it considers the traditional local similarity and the influence of the nodes over the network. Besides that, we can notice that the worst performing algorithms are the original version of PA and LHN, where PA got the worst AUC value in six datasets and LHN in two datasets. The best performing algorithms were the improved version of RA, which gets the highest AUC value 6 times, and the improved version of AA, which has the best AUC value four times. For Jazz, Dolphins, C-Elegans, Macaque, NetScience, and Les Miserables datasets, the worst performing algorithm is the original version of PA with the value of AUC between 0.6 and 0.77, which is very close to random. For the datasets Karate and USAir97, the worst performing algorithm was LHN with $AUC = 0.47$, which is lower than random. Results confirm that the AUC values of original metrics are improved; in

	Jazz		Dolphins		C-elegans		Karate	
	Old	Improved	Old	Improved	Old	Improved	Old	Improved
AA	0,962	0,971	0,78	0,827	0,782	0,826	0,643	0,886
JC	0,956	0,977	0,79	0,833	0,775	0,801	0,5	0,871
PA	0,77	0,79	0,613	0,68	0,615	0,696	0,729	0,871
HD	0,947	0,97	0,79	0,827	0,779	0,815	0,5	0,871
HP	0,948	0,958	0,79	0,8	0,758	0,791	0,621	0,879
LHN	0,904	0,928	0,8	0,8	0,76	0,785	0,471	0,871
PD	0,956	0,965	0,787	0,827	0,78	0,813	0,557	0,871
RA	0,97	0,976	0,787	0,827	0,779	0,826	0,657	0,886
CN	0,954	0,965	0,783	0,827	0,777	0,813	0,629	0,871
SA	0,963	0,978	0,783	0,833	0,77	0,797	0,514	0,871
SO	0,956	0,975	0,79	0,833	0,775	0,806	0,5	0,871
	Macaque		USAir97		Netscience		Les miserables	
	Old	Improved	Old	Improved	Old	Improved	Old	Improved
AA	0,897	0,913	0,931	0,965	0,933	0,933	0,928	0,964
JC	0,883	0,9	0,892	0,949	0,932	0,933	0,858	0,936
PA	0,772	0,818	0,862	0,903	0,618	0,657	0,774	0,84
HD	0,869	0,894	0,882	0,941	0,932	0,933	0,866	0,94
HP	0,855	0,856	0,867	0,913	0,933	0,933	0,832	0,918
LHN	0,797	0,832	0,776	0,892	0,932	0,932	0,8	0,892
PD	0,893	0,913	0,923	0,964	0,933	0,933	0,898	0,966
RA	0,899	0,918	0,939	0,97	0,933	0,933	0,928	0,964
CN	0,891	0,91	0,922	0,956	0,933	0,933	0,904	0,948
SA	0,887	0,901	0,9	0,948	0,932	0,933	0,87	0,938
SO	0,883	0,902	0,892	0,953	0,932	0,933	0,858	0,94

Table 5.2: AUC values of local similarity metrics and their improved version

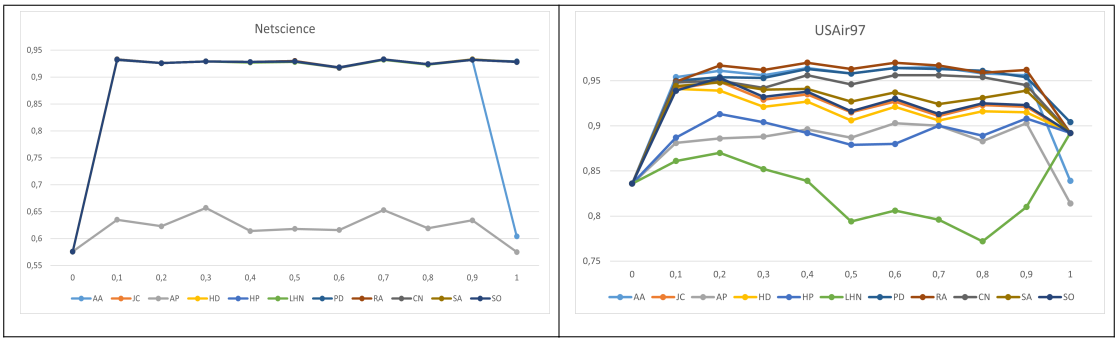
fact, the best AUC obtained in original metrics got enhanced to reach the overall best value. For instance, the best AUC of original metrics in Jazz dataset was given by Salton metric with $AUC = 0.963$ while the improved version reaches $AUC = 0.978$.

Next, we examine the quality of the proposed LSBC combination using different values of α. We study the effect of the dumping factor α on the accuracy of the results.

Table 5.3: Average AUC values using different values of α .

Continued on next page

Table 5.3: Average AUC values using different values of α . (Continued)



In table 5.3, The x-axis refers to the values of α tested on each network. Note that the tested range is $[0; 1]$. For each value of α, the simulation is run 10 times to obtain an average AUC value. The AUC results are shown on the y-axis. We can see that for the dataset C-elegans, the highest AUC value is obtained with $\alpha = 0.4$ for all metrics, but the best performing algorithm is RA. For Dolphins dataset, JC and SO metrics achieve the best AUC value when $\alpha = 0.5$ as well as SA metric with $\alpha = 0.1$. This latter was the highest performing metric on Jazz dataset. When $\alpha = 1$ and $\alpha = 0.8$, AA and RA perform best for Karate dataset. The best performing algorithm for the Macaque dataset was RA with $\alpha = 0.5$ as well as we can draw the same conclusion for USAir97. Finally, PD is the highest performing metric for Les Miserables network with $\alpha = 0.3$. It is clear that the identification of the α value for which the LSBC algorithm will give the more accurate results for different networks is very difficult. In fact, The best value of α changes from network to network.

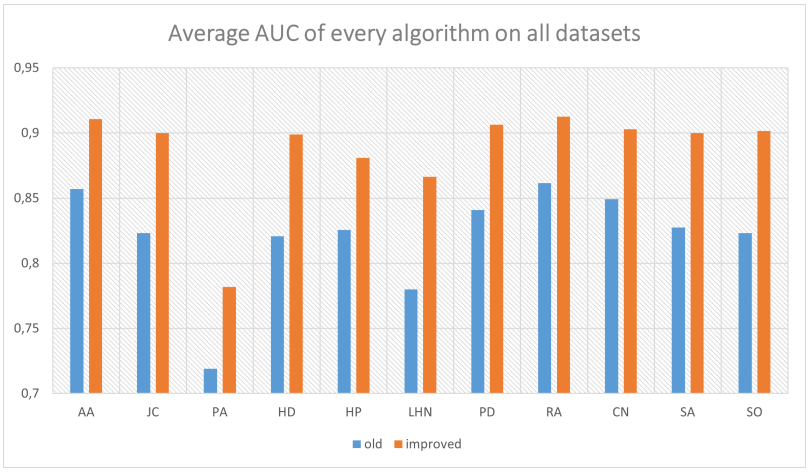


Figure 5.2: Average AUC of all the local metrics

In this part, we analyze the performance of the state of art and the proposed LSBC metrics for all the datasets. To this end, we calculate the average AUC of both original (LS) and improved versions (LSBC) obtained for all the networks. Figure 5.2 shows that the improved version reaches higher AUC values compared to the original versions of local similarity measures. We can notice that the improved AA and RA provide the most accurate results with $AUC = 0.91$.

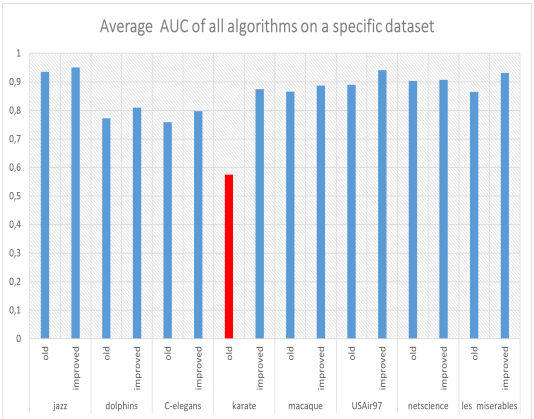


Figure 5.3: Average AUC of all the algorithms for each dataset

Likewise, still in the context of evaluation, we calculate the average AUC of all the algorithms (original LS and improved version LSBC) for each dataset. Figure 5.3 shows that the average AUC is around 0.58 for karate network when we apply the traditional local similarity metrics. While the improved version LSBC reaches around 0.88 which is a big enhancement. Moreover, we can see that for all the networks, the LSBC have achieved a higher value of AUC compared to the local similarity measures.

From the above results, we can conclude that the proposed combination of local similarity metrics and the betweenness centrality LSBC surpasses all the state of arts metrics. It provides more accurate results in various types of network. Note that the improved version of resource allocation RA was the highest performing algorithm in all the datasets with a value of $AUC = 0.91$.

5.3 Link prediction using neural networks

Many works have considered the topological and node features in order to enhance the quality of the link prediction method. The majority of them examine the link prediction problem as a classification task and make use of machine learning algorithms. In the context of supervised learning, one of the most commonly used algorithms is the artificial neural network. In this section, we use neural network model to classify links. In order to evaluate the performance of this latter, we use the accuracy metric and the cross-entropy loss.

5.3.1 Methodology

The primary goal of this section is to quantify the performance of the proposed metric LSBC when modeling the link prediction as a classification task. The classification accuracy is known as an essential metric to evaluate the algorithms

classification. It is defined as follows:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made} \tag{5.2}$$

The loss function is used in machine learning to detect errors or deviations in the learning process, that is to say, the smaller the loss the better the model. A perfect model has a cross-entropy loss of 0. During the model compilation phase, we have used Keras to implement the neural networks. It requires a loss function. In this work, we used *sparse_categorical_crossentropy* [Murphy, 2012]. It is defined by:

$$L_{CE} = -\sum_{i=1}^{n} t_i \times log(p_i) \tag{5.3}$$

Where t_i is the ground-truth label and p_i is the softmax probability for the i^{th} class. Note that in our case, we have only two classes $n = 2$.

Let $G(V, E)$ be an undirected and unweighted graph. In order to convert the link prediction problem to a binary task, we proceed as follows:

1. We read the graph.
2. We generate set1 that contains $|k|$ edges from $\bar{G}$ and $|k|$ edges from G.
3. For every edge in set1, we generate two lists: $FeatureList1_{edge}$ that contains $[AA, AP, CN,$
   $HD, HP, JA, LHN, PD, RA, SA, SO, LSBC]$ and $FeatureList2_{edge}$ that contains $[AA, AP, CN, HD, HP, JA, LHN, PD, RA, SA, SO]$
4. We split each Feature list into train and test sets and then fit them into the neural network.

5.3.2 Results

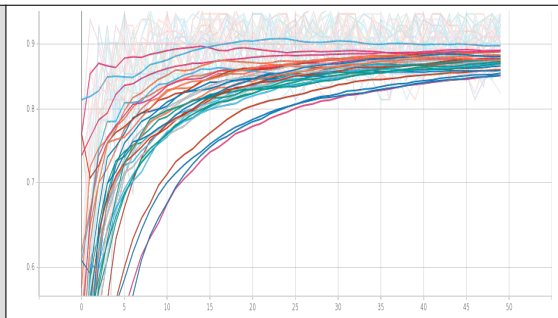
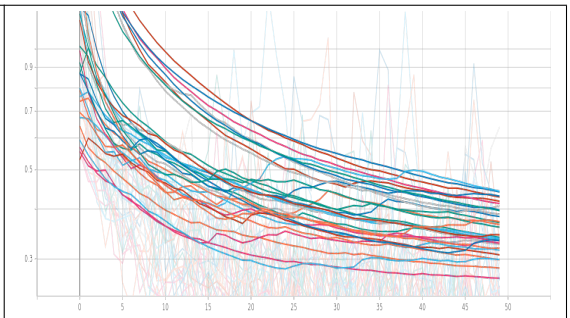
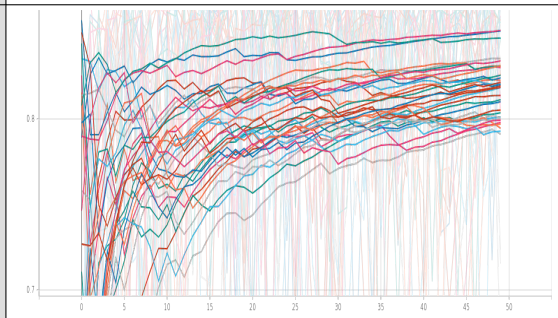
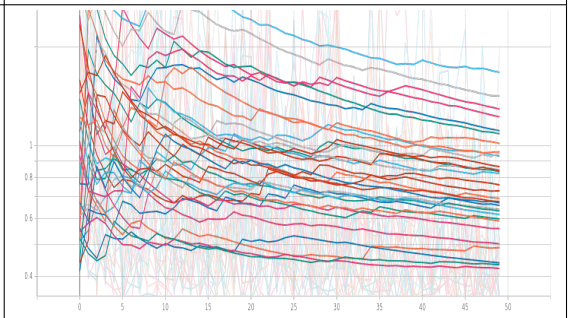
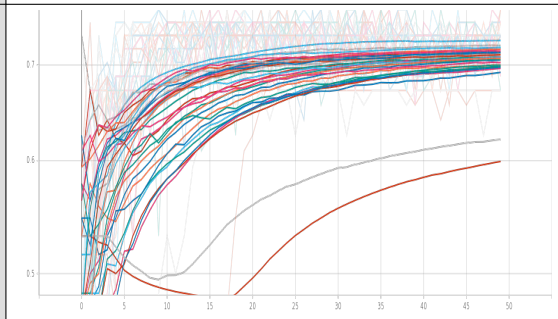
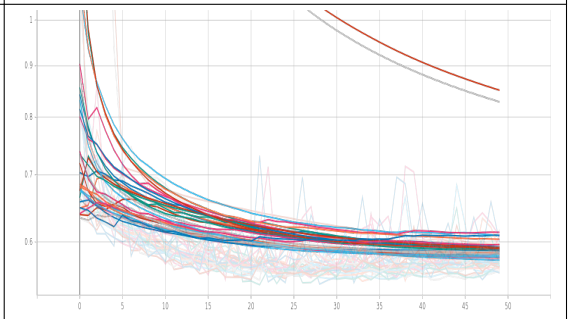
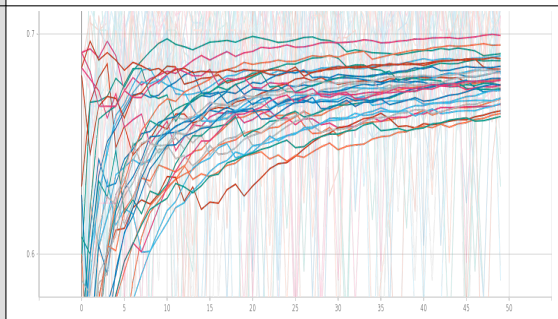
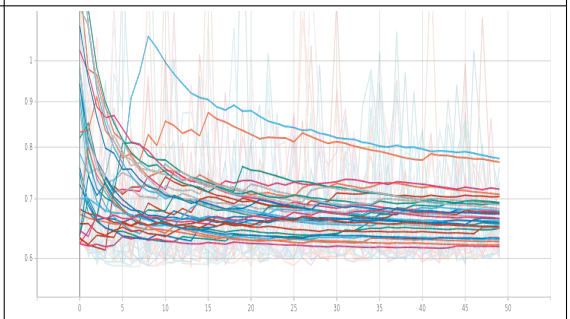
We have trained multiple models where the first and second layers have units in the range $[16, 32, 64]$. Table 5.4 represents the best three results in terms of accuracy and loss. For Les miserables network, the best model in terms of accuracy ($Accuracy = 0.84$) and loss ($L_{CE} = 0.36$) is the $ModelLSBC - 32 \times 64$ where we add the LSBC as an additional feature. The first layer and second layer have 64 units. The same conclusion could be drawn for jazz dataset where the best model is also $ModelLSBC - 64 \times 64$. Its accuracy reaches the value $Accuracy = 0.86$ and its loss score is only $L_{CE} = 0.33$. For Dolphins dataset, the $ModelLSBC - 64 \times 32$ provides the best accuracy value $Accuracy = 0.68$. The $modelLSBC - 64 \times 64$ surpasses the other models in both accuracy and loss for C-elegans network. For karate dataset, the best model is $model - 64 \times 16$ in term of accuracy and $modelLSBC - 32 \times 16$ in term of loss. For Macaque, Net-science and USAir97 datasets, the best model is $modelLSBC - 64 \times 64$ in both accuracy and loss. The most remarkable result to emerge from the data is that the add of LSBC as additional feature increases the accuracy and the loss becomes remarkably minor. Overall, we can conclude that using the $modelLSBC - 64 \times 64$ provides the best results in terms of accuracy and loss for the majority of datasets.

Table 5.4: Results of neural networks.

	Accuracy	Loss

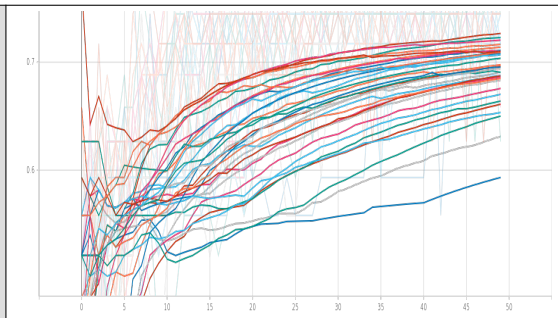
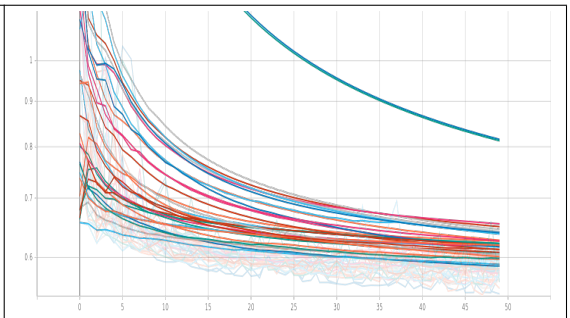
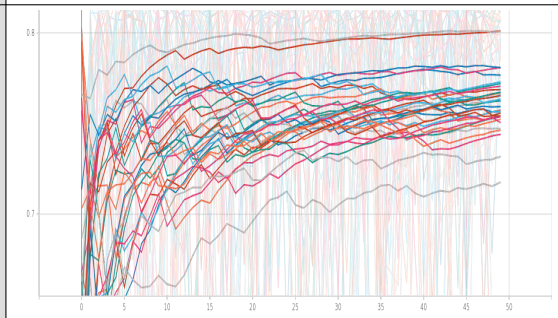
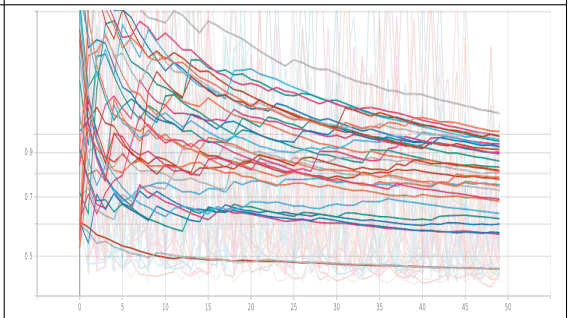
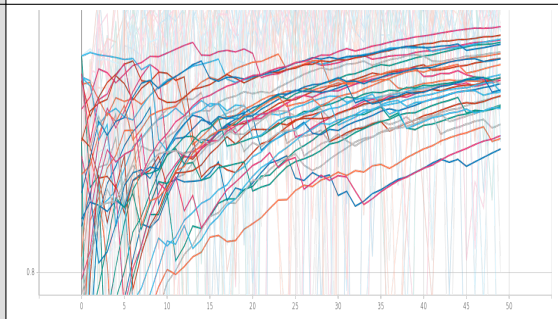
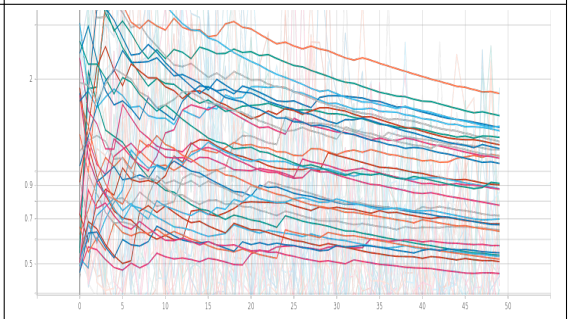
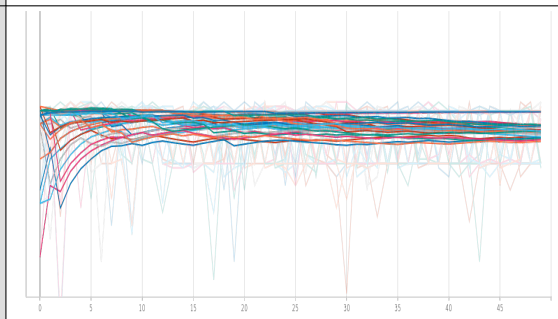
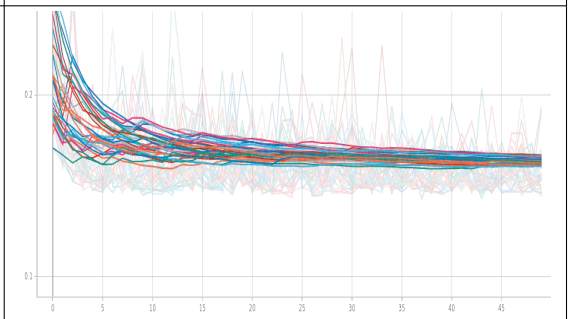
Continued on next page

Table 5.4: Results of neural networks. (Continued)

les miserables		
Jazz		
Dolphins		
C-elegans		

Continued on next page

Table 5.4: Results of neural networks. (Continued)

Karate		
Macaque		
USAir		
Net Science		

5.4 Conclusion

In this chapter, we have proposed a novel method to improve the state-of-the-art local similarity metrics using the betweenness centrality of unlinked nodes. The proposed LSBC method is parameterized in order to control the contribution of the local similarity measure. We have tested the LSBC method on 8 datasets from real world networks. The results demonstrated that the proposed metric outperforms the state-of-the-art metrics in terms of AUC. Moreover, we studied the impact of the used dumping parameter on the AUC results and showed the best one for each dataset. We also show that all the improved versions of local similarity metrics reach a higher value of AUC. Finally, We have investigated the use of machine learning to classify links. We have trained a neural network using metrics as features and shown that when LSBC is used as an additional feature, the classification accuracy increases and the cross-entropy loss approaches zero. In addition, we can improve the accuracy by adding more hidden layers, changing the activation (we used only RELU), adding dropout layers and more epochs.

General Conclusion and Perspectives

In this general conclusion, we outline the significant results of our research on link prediction in complex networks. Furthermore, we introduce and discuss horizons for future research.

In **chapter 1**, we covered the fundamental network structures used in social network analysis. The most common graph representations were then represented. We also used examples to explain node influence and network description. In **chapter 2**, we presents the applications of link prediction, the different methods, and techniques to solve link prediction using local similarity metrics, global similarity metrics, and quasi-local similarity metrics. Finally, solving link prediction using machine learning and deep learning. We have examined their main advantages and drawbacks and proposed some examples to illustrate them. After that, we introduced the evaluation metric (AUC). In **chapter 3**, we have examined some of the most used metrics in this field, compared their performance using AUC, then introduced a new similarity metric. This later relay on the path length between source and destination nodes and the degree of those nodes. Results show that our metric has a higher AUC; it is more accurate in identifying links that will be added in the future. Finally, we have used these metrics to solve link prediction using classification algorithms. Results show that the addition of our metric has a significant impact on the decision process; it improves accuracy. The **chapter 4**

presented a new parametrized metric for link prediction in social networks. This metric was based on both mean received resources (MRR) and a local similarity measure introduced in chapter 2 from the state-of-art. The particularity of our metric relies on its ability to meet the requirements of the system/user by adjusting the parameter. We test the performance of the proposed metric using the AUC value on 8 datasets from different fields, and we compare its results with 11 existing metrics. In **chapter 5**, we presented an ongoing work in which we have introduced new metrics that enable the users to enhance the state-of-the-art metrics using betweenness centrality; the results show a considerable improvement in terms of AUC up to 30%. We also considered graph neural networks as classifiers to distinguish the links. Results show that the model using the LSBC method 64×64 provides the best performance.

Regarding future works, in the **third chapter**, we utilized simple paths; the other option is to use the shortest paths. Instead of employing only the degrees of the source and destination nodes, we may also use the degrees of intermediate nodes. For the **fourth chapter**, we have proposed an approach that boosts the accuracy of state-of-the-art metrics. We may apply this strategy to the measure presented in the first chapter, and we can also utilize a percentage of the shortest paths, not all the shortest routes, to determine the value of MRR. The metric in **chapter five** uses the betweenness, which is time-consuming; we might employ other metrics of node importance, such as degree or eigenvectors. Finally, we want to convert our suggested metrics into directed networks, weighted networks, labeled networks, and bipartite graphs. Also, we want to study link prediction solutions utilizing graph embedding and dimensionality reduction. We hope to apply our measurements to larger datasets.

Bibliography

Eigenvector centrality. URL `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.centrality.eigenvector_centrality.html`.

Sisay Fissaha Adafre and Maarten de Rijke. Discovering missing links in wikipedia. In *Proceedings of the 3rd international workshop on Link discovery*, pages 90–97, 2005.

Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

Farshad Aghabozorgi and Mohammad Reza Khayyambashi. A new similarity measure for link prediction based on local structures in social networks. *Physica A: Statistical Mechanics and its Applications*, 501:12–23, 2018.

Iftikhar Ahmad, Muhammad Usman Akhtar, Salma Noor, and Ambreen Shahnaz. Missing link prediction using common neighbor and centrality based parameterized algorithm. *Scientific reports*, 10(1):1–9, 2020.

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, Eric P Xing, and Tommi Jaakkola. Mixed membership stochastic block models for relational data with

application to protein-protein interactions. In *Proceedings of the international biometrics society annual meeting*, volume 15, 2006.

Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, volume 30, pages 798–805, 2006.

Gerald Alexanderson. About the cover: Euler and königsberg's bridges: A historical view. *Bulletin of the american mathematical society*, 43(4):567–573, 2006.

Akash Anil, Durgesh Kumar, Shubhanshu Sharma, Rakesh Singha, Ranjan Sarmah, Nitesh Bhattacharya, and Sanasam Ranbir Singh. Link prediction using social network analysis over heterogeneous terrorist network. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 267–272. IEEE, 2015.

Jibouni Ayoub, Dounia Lotfi, Mohamed El Marraki, and Ahmed Hammouch. Accurate link prediction method based on path length between a pair of unlinked nodes and their degree. *Social Network Analysis and Mining*, 10(1):1–13, 2020.

Jibouni Ayoub, Dounia Lotfi, and Ahmed Hammouch. Mean received resources meet machine learning algorithms to improve link prediction methods. *Information*, 13(1):35, 2022.

Furqan Aziz, Haji Gul, Ishtiaq Muhammad, and Irfan Uddin. Link prediction using node information on local paths. *Physica A: Statistical Mechanics and its Applications*, 557:124980, 2020.

Furqan Aziz, Victor Roth Cardoso, Laura Bravo-Merodio, Dominic Russ, Samantha C Pendleton, John A Williams, Animesh Acharjee, and Georgios V Gkoutos. Multimorbidity prediction using link prediction. *Scientific Reports*, 11(1):1–11, 2021.

Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644, 2011.

Albert-Laszlo Barabasi. *Linked: How everything is connected to everything else and what it means*. Plume, 2003.

Albert-László Barabási. Scale-free networks: a decade and beyond. *science*, 325 (5939):412–413, 2009.

Albert-Laszlo Barabâsi, Hawoong Jeong, Zoltan Néda, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3-4):590–614, 2002.

John A Barnes. Graph theory and social networks: A technical comment on connectedness and connectivity. *Sociology*, 3(2):215–232, 1969.

V Batagelj and A Mrvar. Pajek datasets. *USAir97. net*, 2006.

Alex Bavelas. Communication patterns in task-oriented groups. *The journal of the acoustical society of America*, 22(6):725–730, 1950.

Kamal Berahmand, Asgarali Bouyer, and Negin Samadi. A new centrality measure based on the negative and positive effects of clustering coefficient for identifying influential spreaders in complex networks. *Chaos, Solitons & Fractals*, 110:41–54, 2018.

Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

Phillip Bonacich. Power and centrality: A family of measures. *American journal of sociology*, 92(5):1170–1182, 1987.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Dongbo Bu, Yi Zhao, Lun Cai, Hong Xue, Xiaopeng Zhu, Hongchao Lu, Jingfen Zhang, Shiwei Sun, Lunjiang Ling, Nan Zhang, et al. Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic acids research*, 31(9):2443–2450, 2003.

Andrew G Bunn, Dean L Urban, and Timothy H Keitt. Landscape connectivity: a conservation application of graph theory. *Journal of environmental management*, 59(4):265–278, 2000.

Cambridge. social network. `https://dictionary.cambridge.org/fr/dictionnaire/anglais/social-network`, December 2021.

Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Scientific reports*, 3(1):1–14, 2013.

Sheryl L Chang, Mahendra Piraveenan, and Mikhail Prokopenko. Impact of network assortativity on epidemic and vaccination behaviour. *Chaos, solitons & fractals*, 140:110143, 2020.

Hsinchun Chen, Xin Li, and Zan Huang. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, pages 141–142. IEEE, 2005.

Giulio Cimini, Rossana Mastrandrea, and Tiziano Squartini. *Reconstructing networks*. Cambridge University Press, 2021.

Reuven Cohen and Shlomo Havlin. *Complex networks: structure, robustness and function*. Cambridge university press, 2010.

Reinhard Diestel. *Graph theory*. Graduate Texts in Mathematics. Springer, 3rd edition, 2006. ISBN 9783540261834,3540261834. URL `http://gen.lib.rus.ec/book/index.php?md5=a330de31ca23da91e5fe49a1346cdd00`.

Sergei N Dorogovtsev, Sergei N Dorogovtsev, and Jose FF Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. Oxford university press, 2003.

Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

Ilham Esslimani, Armelle Brun, and Anne Boyer. Densifying a behavioral recommender system by social networks link prediction methods. *Social Network Analysis and Mining*, 1(3):159–172, 2011.

Leonhard Euler. The seven bridges of königsberg. *The world of mathematics*, 1: 573–580, 1956.

Peter C Fishburn. Letter to the editor additive utilities with incomplete product sets: application to priorities and assignments. *Operations Research*, 15(3):537–542, 1967.

Francesco Folino and Clara Pizzuti. Link prediction approaches for disease networks. In *International Conference on Information Technology in Bio-and Medical Informatics*, pages 99–108. Springer, 2012.

Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, 19(3):355–369, 2007.

Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Jennifer Golbeck. *Analyzing the social web*. Newnes, 2013.

Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *Advances in neural information processing systems*, 17, 2004.

Shensheng Gu and Ling Chen. Link prediction based on precision optimization. In *International Conference on Geo-Informatics in Resource Management and Sustainable Ecosystem*, pages 131–141. Springer, 2016.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer, 1995.

Siyao Han and Yan Xu. Link prediction in microblog network using supervised learning with multiple features. *J. Comput.*, 11(1):72–82, 2016.

James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

Steve Hanneke and Eric P Xing. Network completion and survey sampling. In *Artificial Intelligence and Statistics*, pages 209–215. PMLR, 2009.

Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

Bhadeshia Hkdh. Neural networks in materials science. *ISIJ international*, 39(10): 966–979, 1999.

Hai-Bo Hu and Xiao-Fan Wang. Unified index to quantifying heterogeneity of complex networks. *Physica A: Statistical Mechanics and its Applications*, 387 (14):3769–3780, 2008.

Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 108:87–111, 2018.

Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, 2002.

Ayoub Jibouni, Dounia Lotfi, Mohamed El Marraki, and Ahmed Hammouch. A novel parameter free approach for link prediction. In *2018 6th International Con-*

ference on Wireless Networks and Mobile Communications (WINCOM), pages 1–6. IEEE, 2018.

Asif Karim, Sami Azam, Bharanidharan Shanmugam, Krishnan Kannoorpatti, and Mamoun Alazab. A comprehensive survey for intelligent spam email detection. *IEEE Access*, 7:168261–168295, 2019.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *CEAS*, 2004.

Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.

Jérôme Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22nd international conference on world wide web*, pages 1343–1350, 2013.

Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical review letters*, 87(19):198701, 2001.

Elizabeth A Leicht, Petter Holme, and Mark EJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.

Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. *Advances in neural information processing systems*, 25, 2012.

Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Cong Li. *Characterization and Design of Complex Networks*. PhD thesis, 2014.

LanXi Li, XuZhen Zhu, and Hui Tian. Link prediction based on nonequilibrium cooperation effect. *International Journal of Modern Physics B*, 32(11):1850128, 2018.

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems*, 33:4465–4478, 2020.

Xin Li and Hsinchun Chen. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, 54 (2):880–890, 2013.

David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

Ryan Lichtnwalter and Nitesh V Chawla. Link prediction: fair and effective evaluation. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 376–383. IEEE, 2012.

Weiping Liu and Linyuan Lü. Link prediction based on local random walk. *EPL (Europhysics Letters)*, 89(5):58007, 2010.

Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

Linyuan Lü, Ci-Hang Jin, and Tao Zhou. Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, 80(4):046122, 2009.

R Duncan Luce and Albert D Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949.

David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

mark zuckerberg. Facebook. URL `http://www.facebook.com`.

Victor Martinez, Fernando Berzal, and Juan-Carlos Cubero. Adaptive degree penalization for link prediction. *Journal of Computational Science*, 13:1–9, 2016.

Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, 49(4):1–33, 2016.

Tadej Matek and Svit Timej Zebec. Github open source project recommendation system. *arXiv preprint arXiv:1602.02594*, 2016.

McGraw-Hill. Dictionary of scientific and technical terms, 6e. information network. `https://encyclopedia2.thefreedictionary.com/information+network`, Retrieved December 3 2021 2003.

Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Kenta Nakai and Minoru Kanehisa. Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins: Structure, Function, and Bioinformatics*, 11(2):95–110, 1991.

Ramesh M Nallapati, Amr Ahmed, Eric P Xing, and William W Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 542–550, 2008.

Elahe Nasiri, Kamal Berahmand, Mehrdad Rostami, and Mohammad Dabiri. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Computers in Biology and Medicine*, 137:104772, 2021.

László Négyessy, Tamás Nepusz, László Kocsis, and Fülöp Bazsó. Prediction of the main cortical areas and connections involved in the tactile function of the visual cortex by network analysis. *European Journal of Neuroscience*, 23(7): 1919–1930, 2006.

Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

Mark EJ Newman. Assortative mixing in networks. *Physical review letters*, 89 (20):208701, 2002.

Mark EJ Newman. Mixing patterns in networks. *Physical review E*, 67(2):026126, 2003.

Mark EJ Newman. A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54, 2005.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

Rogier Noldus and Piet Van Mieghem. Assortativity in complex networks. *Journal of Complex Networks*, 3(4):507–542, 2015.

Antonio Pecli, Bruno Giovanini, Carla C Pacheco, Carlos Moreira, Fernando Ferreira, Frederico Tosta, Júlio Tesolin, Marcio Vinicius Dias, Silas P Lima Filho, Maria Cláudia Cavalcanti, et al. Dimensionality reduction for supervised learning in link prediction problems. In *ICEIS (1)*, pages 295–302, 2015.

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. *Journal of Machine Learning Research*, 12:2825–2830, 2011a.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011b.

Manisha Pujari. *Link Prediction in Large-Scale Complex Networks (Application to Bibliographical Networks)*. PhD thesis, Université Sorbonne Paris Cité, 2015.

Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition*. Packt Publishing. ISBN 978-1789955750.

Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L

Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555, 2002.

Ryan Rossi and Nesreen Ahmed. The network data repository with interactive graph analytics and visualization. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

Gerard Salton and Michael J McGill. *Introduction to modern information retrieval*. mcgraw-hill, 1983.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

Robert Sedgewick. *Algorithms in C, part 5: graph algorithms*. Pearson Education, 2001.

Chenxi Shao and Yang Xu. Data exchange similarity based on flow field for link prediction problem. In *2016 Sixth International Conference on Information Science and Technology (ICIST)*, pages 84–89. IEEE, 2016.

Vishnu Dutt Sharma, Santosh Kumar Yadav, Sumit Kumar Yadav, Kamakhya Narain Singh, and Suraj Sharma. An effective approach to protect social media account from spam mail–a machine learning approach. *Materials Today: Proceedings*, 2021.

Tom AB Snijders. The degree variance: an index of graph heterogeneity. *Social networks*, 3(3):163–174, 1981.

Th A Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34, 1948.

Sucheta Soundarajan and John Hopcroft. Using community information to improve the precision of link prediction methods. In *Proceedings of the 21st international conference on World Wide Web*, pages 607–608, 2012.

Shan Suthaharan. Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst*, 36:1–12, 2016.

Robert E Ulanowicz and Donald L DeAngelis. Network analysis of trophic dynamics in south florida ecosystems. *US Geological Survey Program on the South Florida Ecosystem*, 114:45, 2005.

Jorge Carlos Valverde-Rebaza and Alneu de Andrade Lopes. Link prediction in complex networks based on cluster information. In *Brazilian Symposium on Artificial Intelligence*, pages 92–101. Springer, 2012.

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

Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.

Zhitao Wang, Yong Zhou, Litao Hong, Yuanhang Zou, and Hanjing Su. Pairwise learning for neural link prediction. *arXiv preprint arXiv:2112.02936*, 2021.

Stanley Wasserman, Katherine Faust, et al. Social network analysis: Methods and applications. 1994.

Duncan J Watts and Steven H Strogatz. Collective dynamics of "small-world"networks. *nature*, 393(6684):440–442, 1998.

Mei Wu, Shunyao Wu, Qi Zhang, Chuanyu Xue, Hongsheng Kan, and Fengjing Shao. Enhancing link prediction via network reconstruction. *Physica A: Statistical Mechanics and its Applications*, 534:122346, 2019.

Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5(Aug):975–1005, 2004.

Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009.

Yu-Xiao Zhu, Linyuan Lü, Qian-Ming Zhang, and Tao Zhou. Uncovering missing links with cold ends. *Physica A: Statistical Mechanics and its Applications*, 391 (22):5769–5778, 2012.

Résumé

Le terme « réseaux complexes » est utilisé pour décrire l'ensemble des relations d'amitié, connaissances, ou autres types de relations entre des entités. Ces réseaux se caractérisent par leurs évolutions en termes d'utilisateurs et en termes de relations crées entre ces derniers. La théorie de graphe donne des outils efficaces pour étudier et analyser ces réseaux, beaucoup de notions comme la notion d'amis, chemins, le plus court chemin, degré d'un nœud, et autres sont d'origine de la théorie de graphe. Des exemples de méthodes et de mesures comme Jaccard, Adamic/Adar, Katz sont basés sur ces notions. L'un des problèmes les plus importants dans l'analyse des réseaux sociaux est la prédiction des liens. Dans ce problème on cherche à trouver les liens qui se produisant dans le temps t+1 en se basant sur les liens disponibles au temps t. Pour résoudre ce problème plusieurs métriques ont été développées comme le nombre d'amis en communs, le plus court chemin, coefficient de Jaccard et autres. Le but de cette thèse est de proposer de nouvelles mesures qui dépassent la précision des méthodes de l'état de l'art dans un temps d'exécution raisonnable. Nous avons également utilisé des méthodes d'apprentissage automatique pour résoudre le problème de prédiction de liens. Finalement, une variété d'expérimentations a montré l'efficacité et la précision de toutes les méthodes proposées sur une série de réseaux réels.

Mots-clefs : Réseaux complexes, théorie des graphes, analyse des réseaux sociaux, recommandation sociale, mesures de similarité, aire sous la courbe, apprentissage automatique, modèles de régression, comportements des réseaux, prédiction des liens.

Abstract

The term « complex network » has entered modern usage, and it now refers to circles of friends, acquaintances, and any other type of relationship between entities. Due to their enormous structure and dynamic behavior, many real-world systems are modeled as complex networks. Graph theory provides practical tools for comprehending and analyzing their process and evolution. Many structural properties are used, such as a neighbor, paths, shortest paths, and node degree. Based on those features, we have proposed metrics to predict the upcoming links in a network. This problem is known as link prediction, where we predict the upcoming links at time t+1 from a known graph at time t. Metrics such as Jaccard, Common neighbors, and Local paths, have been proposed to solve the link prediction problem. However, it is still an active zone where the existing metrics lack accuracy or cannot deal with large-scale networks. In this thesis, our primary motivation is to propose new metrics and models that provide accurate predictions in reasonable execution time. Then, we evaluate the proposed methods against state-of-the-art metrics using Area Under Curve. In addition, we use machine learning models to solve the link prediction problem as a classification task. All the proposed methods show good performance in different types of real-world datasets.

Key Words: Complex Networks, Graph theory, Social Network Analysis, Social Recommendation, Similarity Measures, Area Under Curve, Machine Learning, Regression Models, Network Behaviors, Link Prediction.