PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
الجمهورية الجزائرية الديمقراطية الشعبية
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
وزارة التعليم العالي و البحث العلمي
UNIVERSITY OF AMAR TELIDJI LAGHOUAT
جامعة عمار ثليجي بالأغواط

FACULTY OF SCIENCES
كلية العلوم
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES
قسم الرياضيات و الإعلام الآلي

Master Thesis

Domain Mathematics and Computer Science
Field Computer Science
Option Networks, Systems and Distributed Applications

By:

Babaghayou Messaoud

Topic

Machine Learning for Link Prediction in Complex Networks

Defended Publicly in Front of the Jury Composed of

M_s.	H.CHERROUN	PRESIDENT	UATL
M_s.	A.CHORANA	EXAMINER	UATL
M_r.	Y.OUINTEN	EXAMINER	UATL
M_r.	A.LAKHDARI	SUPERVISOR	UATL

Academic Year 2015/2016

Dedication

This thesis is dedicated to:

- The sake of Allah, my Creator and my Master, My great teacher and messenger, Mohammed (May Allah bless and grant him), who taught us the purpose of life.
- My familly, Father, Mother and Sister. You were all the support that i rely on.
- My friends that i know.
- My neerest friends, i know them, and they know themselves without naming.
- All my Prophs who gave me knowledge.
- Laghouat University prophs and students.
- Finally, to every one who contributed from near or far in this thesis.

Babaghayou Messaoud...

Acknowledgements

Firstly, all praise and thanks be to Allah. We have asked Him for His help and we always got it.

I would like then to express my deepest gratitude to my family: Father, Mother and my lonely Sister:

- Dad, thank you for being so strict with your rules that I missed out on some moments that could have led me down the wrong path. For standing your ground when I begged to go somewhere or do something that, in hindsight, could have introduced me to things that would jeopardize my future forever.
- Mom, thank you for making me realize that I'm worth everything in this world. That I must be treated like an important man, and that I should never settle for less than what I deserve.
- Sister, thank you for being every time with me, talking or joking and i wish you will get high marks in your bac exams that you will do soon.

Without each of you, I'd be nowhere near the person I am – and the person I'm still working on becoming. There aren't enough words in the world to express my appreciation, but I think this is a good start. I owe you one.

I thank Mr.Lakhdari for accepting supervising me despite his big busyness and for his encouragements when I thought that I am facing a hard thesis and when I got despairs. I also would like to thank Ms.Bouzouad and Ms.Chorana for their big guidelines and useful help during the realization of this work when I asked them while they have their own busyness and a lot of works and for accepting to be members in the jury. Big thanks go to Mr.Ouinten who gave us a well-knowledge in many science fields during our academic years and for accepting to be one of the jury members and to examine this modest manuscript.

I won't forget those who I consider friends, they are too numerous to name. And I have the pleasure to thank my near friends –without naming- who always have asked for me and for my progress even they have their own problems and busyness, and those who I have asked for them you are all a part of this thesis.

Last but not least, deepest thanks go to all people who have had a significant impact on my graduate career that I will never forget. You have all provided support, encouragement and interest in my thesis work.

I would not be who I am today without you all, thanks again.

Babaghayou Messaoud...

Abstract

Nowdays, networks are omnipresent. The study and understanding of these networks become a greater need. The purpose of this work, is to investigate link prediction task in complex networks using Machine learning techniques. In fact, we propose two approaches to perform link prediction: supervised and unsupervised one. In both techniques a link or a pair of nodes is characterized by several features based on network topology-based metrics. In addition, we investigate many combined features. Concerning the supervised approach, we investigate the KNN and decision tree methods to build the link prediction models. While in the unsupervised approach, we rely on ranking strategy. An experimental study is performed on real networks. The results show that the supervised approach using gathered features reaches good performances with 84% f-measure.

keywords: Machine Learning, Link Prediction, Complex Networks, Supervised Leaning, Unsupervised Learning, Classification, Node-based Metrics.

مُلخّص

في ايّامنا هذه، تُعتبر الشبكات عُنصرًا ظاهرًا في كل مجال. دراستها و فهمها أصبح مُتطلب ضروري. الهدف من هذا العمل هو البحث في موضوع توقّع الرابط في الشبكات المُعقّدة باستعمال تقنيات آلات التعلّم. في الحقيقة، نقترح نهجين لتنفيذ عملية توقع الروابط: نهج مراقب و آخر غير مُراقب. في كلا التقنيتين، الرابط او الثُنائي العُقدي يكون مُمثّل بعدة خصائص مُتمحورة حول مقاييس البنية الخاصة بالعُقدة في الشبكات. بالاضافة الى هذا، نبحث في كثير من الخصائص المُركّبة. بخُصوص النهج المراقب، نبحث في طريقتي ال KNN و Decision Tree لبناء نماذج توقع الرابط. و فيما يخص النهج غير المراقب، نعتمد على استراتيجية الترتيب. دراسة تجريبية تم القيام بها على شبكات حقيقية. النتائج تُظهر ان النهج المُراقب الذي يستعمل خصائص مُجمّعة يحقق آداء جيد بنسبة 84 % من وحدة القياس اف.

الكلمات المفتاحيّة: آلات التعلّم، توقّع الرابط، الشبكات المعقّدة، التعلم المُراقب، التعلم غير المُراقب، التصنيف، المقاييس المُتعلّقة بالعُقد.

Résumé

De nos jours , les réseaux sont omniprésents. L'étude et la compréhension de ces réseaux devient un besoin accru. Le but de ce travail est d'investiguer la tâche de prédiction de liens utilisant les techniques d'apprentissage. En effet, nous avons proposé deux approches de prédiction de lien : l'approache supervisée et l'approache non-supervisée. Dans les deux approches un lien, ou une paire de nœuds, est caractérisé par différentes mesures basées sur la topologie du réseau. Nous avons aussi investigué la combinaison de plusieurs caractéristiques. Concernant l'approche supervisée, nous avons utilisé les méthodes de KNN et les arbres de décision pour construire les modèles de prédiction de liens. Par contre, nous avons utilisé une stratégie de classement dans l'approche non-supervisée. L'étude expérimentale est effectuée sur des réseaux réels. Les résultats prouvent que l'approche supervisée utilisant des caractéristiques composées atteint de bonne performances, 84% f-mesure.

mots-clés: L'apprentissage Automatique, La prédiction des Liens, Les Réseaux Complexes, L'apprentissage Supervisé, L'apprentissage non-Supervisé, La Classification, Les Métrique Basé Sur les Nœud .

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

AA Adamic-Adar Coefficient. 21

CN Commun Neighbors. 21

DT Decision Tree. 36

HG Haggle Network Dataset. 44

HY Hypertex 2009 Network Dataset. 44

JC Jaccard Coefficient. 21

KNN K-Nearest Neighbor. 37

LP Link Prediction. 14

ML Machine Learning. 27

PA Preferential Attachment. 21

SL Supervised Learning. 27

SLLP Supervised Learning Link Prediction. 31

UL UnSupervised Learning. 27

ULLP UnSupervised Learning Link Prediction. 31

Introduction

If we take a look around, we will find that networks are everywhere. Besides, we, as individuals, can consider ourselves as units of different Social Networks based on relationships of different kinds, or in Biological systems : the delicate results of a biochemical network reactions and many other examples [1].

The fast growing of these networks leads to the emergence of many Complex Networks Analysis like studying Complex Networks Centrality, Link Prediction, Community Detection and others [2].

The task of understanding how these networks evolve over time is a fundamental question that still remains not well understood. Techniques, ways, approaches used to predict new links that may appear is an important goal that takes a big interest nowdays. Link prediction applications can be found in several domains such as friends recommendation, terrorist networks detection, collaborations in academic social networks and many other examples [3].

In order to tackle these challenges in this work, we investigate and explore the link prediction problem by proposing approaches that rely on machine learning techniques. Namely supervised and unsupervised learning link prediction appoaches. For each approach, we use simple features relying on the well-known network topology-based metrics to achieve the link prediction task, then, we investigate a non-classic pair of nodes representation. In fact, most works in the litterature deploy features one by one and they did not investigate their combination which

is the weighted sum of these features. Moreover, in the supervised learning, we create a vector representation as gathered features that characterize a pair of nodes. In addition, we compare between supervised and unsupervised approaches from one hand, and between features' discrimination power in the other hand.

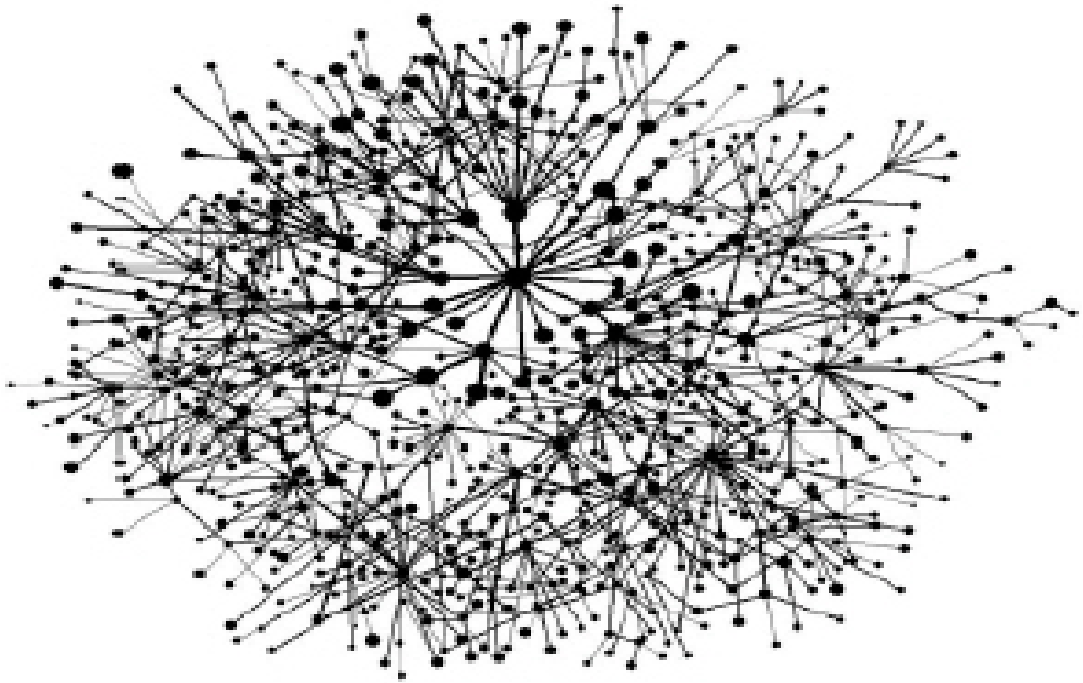
We use real networks with an important size, unlike the most used networks whith less than 100 nodes.

- The first Chapter of this manuscript is dedicated to Complex Networks, their forms, their importance in real life, their common features and their analysis.
- The second Chapter introduces the Link Prediction problem with some applications. We also describe some used metrics and techniques.
- In the third Chapter we describe and explain the principle of our approaches. Followed by the feature choice, then, the used Machine learning techniques.
- Chapter IV is dedicated to describe used tools, metrics and datasets, followed by experiments and obtained results with discussion.

We conclude with some observations and proposing further researches.

Chapter I

Complex Networks: Generalities



In this chapter we describe generalities on Complex Networks with some examples in real life, and we give common features of Complex Networks, also we show Complex Networks Analysis.

1 Complex Networks

1.1 Definition

Complex Networks represent a vast field of many systems whether in nature or society. Many well-known examples can illustrate what is complex networks, such as network of chemicals linked by chemical reactions, Internet, or linked connection between computers and routers and other similar examples. Unlike previously, where these systems were classed in the random graphs, it is increasingly reconized that both topologies and evolutions of our real networks nowadays is controlled and administered by a durable and strong rules [4]. At the same time we can see that networks can be tangible objects in Euclidean space, like electric power grids, Internet, highway/subway systems, and neural networks.

Moreover, they can be entities defined in an abstract space; networks of acquaintances or networks that establish collaborations between individuals. Thus, we will realize that all these Networks are a set of phenomena that take place in our lives [5].

So what we can say is that we have to generate a representation for these phenomena, by considering constituents as nodes or vertices (we can also call them units) of a network, and interactions between them are modeled by links or edges (arcs in case of digraphs) of that network.

Until now we have the suitable representation in one hand and a set of mathematical tools (across centuries of works in applied mathematics and statistical mechanics) in the other hand, by combining both of them, we can understand and extract a lot of data from these observed phenomena and identify some rules, furthermore we can control and manipulate some of them conveniently [1].

Figure I.1: Diseases' Spread in Epidemiology [9]

1.2 Examples

Complex Networks are the core of many problems across the world. Consequently, an interdisciplinary study was growing rapidly during the few last decades [6]. In what follows we show a brief highlighting on such type of Complex Networks:

1.2.1 Epidemiology

Epidemiology is the study of how some diseases may occur in different groups of people and tries to find its causes, it is used to prevent illness. Moreover, it can be considered as a guide to manage patients in whom disease has already developed [7]. In other words, it is the study of health-related states [8], one of the latest serious and critical diseases is that called "zika-virus". Figure I.1 shows two node classes, those who are ill with red circles (vertices), and those who are not with green circles (vertices), so the disease's spread will start as soon as they contact each other (the two categories). Network analysis can reduce or hold the virus's spread by detecting areas where the virus performs its maximum speed of spreading.

1.2.2 Biological Networks

Some biological processes need to be represented often in a network form like protein-protein interaction networks or metabolic networks. Important tasks of modeling, analysis and virtualization have emerged in science nowadays to understand it better and to make biological sense of the big amount of

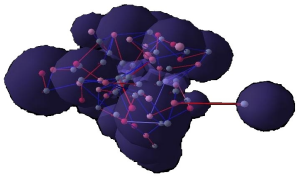


Figure I.2: An Amorphous Computational Network

complex data that is being generated. In fact, it is difficult to interpret due to the complexity of the relationships between the data, that is why biological networks belongs to complex networks type [10]. Figure I.2 illustrates an amorphous computational network model.

1.2.3 Collaboration Networks

This kind of networks are defined as follows : if we have two scientists, and both of them are considered connected only if they have authored one or more papers together [11].

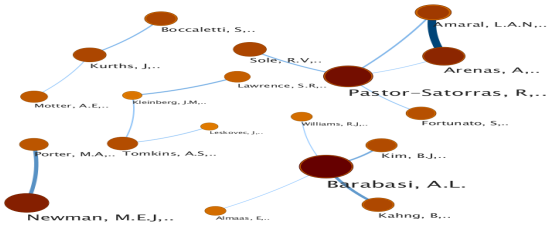


Figure I.3: Collaboration Network in Network Science Field [12]

Figure I.3 illustrates a collection of scientists who are gathered to form a collaboration network.

1.2.4 Citation Networks

Similarly to previous type of networks which concerns scientific research domain, the study of citation networks is also used for both articles and journals, these analysis proceed by considering the similarity of those papers and submitting the set of similarity measures to scaling procedure. There are two common methods in bibliometric coupling, where two citing articles are

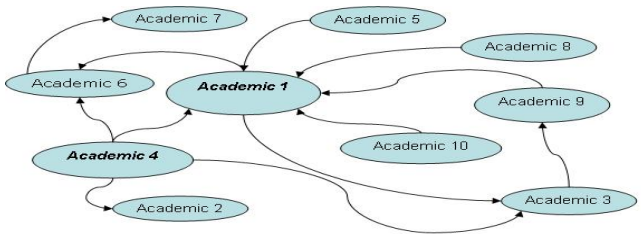


Figure I.4: Academic Citations Network [14]

similar to the extent they cite the same literature, and also co-citation analysis where cited articles are similar to the extent they are cited by the same citing articles [13]. In figure I.4, when any academic paper cite another one, an edge will be created between the first paper and the second one.

1.2.5 Social Networks

A Social Network can be seen as a practice of expanding the number of one's business and social contacts using connections between individuals. Additionally, social networking establishes interconnected online communities, called "social graphs" that allow people making contacts that would be useful for them to know,even if they didn't meet before. Based on the used social media platform, members could be able to make communications or contact any other member that has a relation with them. On the other hand, members can also contact anyone they have connection with another characteristic is that there are some services that require members to have a pre-existing connection to contact or communicate with each other [15]. Further, social networks provide a common forum for representatives of sociology, anthropology, political science, biology, economics, communications science and other disciplines that are created in order to study the empirical structure of social relations and associations that can be represented in network form [16].

Figure I.5: Representation of a Social Network [17]

Graph I.5 shows people (individuals) and some contacts (relationships) between them.

1.2.6 Online Social Networks

An Online Social Network is extended from real Social Networks as well. Thus, it allows people to communicate with friends and acquaintances both old and new ones. Moreover, Online Social Networks include networking for business, pleasure, and all points between them. Also, networks themselves have many and different purposes, and their online counterparts work in many ways [18]. To make communication in this type of network, social media is needed. Here, as a short definition, social media is any support or medium that allows data transfer from individual to another one. Online Social Networks seen in Figure I.6 is a general concept. While the majority of us use at least 2 or 3 online social networks in their life. So, it is important to describe some of the famous ones according to www.ebizmba.com [19] which is a ranking Website that uses updated average of each Website's Alexa Global Traffic Rank, and U.S. Traffic Rank for most social networks:

Figure I.6: Online Social Networks [20]

FaceBook:

Facebook is a very popular free social network Website that allows his registered users to create profiles, upload or download photos and videos, sharing posts and topics, send messages and keep in touch with friends, acquaintances, family. The site is available in 37 different languages, also, it includes public features such as marketplace(post and read classified ads), groups(interaction between members who have common interests), events (publicize an event or invite guests and track those who plan to attend), Pages (create and promote and manage public page built around a specific topic), Presence technology(it allows members to see which contacts from their friends list are online and chat), news updates (updates of friends will appear to you for example when they do some activities like shares or tags...). [21].

Twitter:

Twitter is a free microblogging service for social networking that allows its registered members to broadcast short posts, we call them "tweets". Twitter members can broadcast their own tweets and follow other users' tweets by using multiple platforms and devices. Also tweets can be sent by cell phone as text messages. Unlike Facebook, Twitter's settings are public, thus members don't need to approve social connections, so anyone can follow anyone on public Twitter. Moreover, tweets are limited in characters number (up to 140 chars) this constraint of Twitter's message length (short message service SMS). [22].

LinkedIn:

linkedIn is a networking site designed especially for business community side. Its purpose is to let registered members to make and document networks of other people they know and trust professionally. Any LinkedIn

member's profile page that includes his employment history and education, has professional network news feeds and a limited number of customizable modules. To be a member of LinkedIn is free (basic membership). Also network members are called -connections-. Unlike the previous two free social networking sites Facebook and Twitter, LinkedIn requires connections to have pre-existing relationship between its members. [23].

Pinterest:

Pinterest is another social Website made for sharing and categorizing online images, it is a concatenation of the two words "pin" and "interest". In Pinterest categories you can find science, fashion, food and drink, home décor, sport, art, and it include an almost endless list of other possibilities. A Pinterest user can add a "Pin it" button to his browser in order to select and pin online images to "virtual pinboards", which are used to organize categories. Other thing is when you click on an image, it will take you to the original source. [24].

Google Plus+:

Google+ is the 5th social network according to ebizmba's ranking [19]. Google+ is trying to replicate the way people across the world interact offline more closely than is the case in the previous social networking sites (Facebook, Twitter...). This project's slogan is "Real-life sharing rethought for the Web". One of its important features are (circles, hangouts, huddle, instant upload, streams, sparks...). Google+ is also one of the Google's applications set, such as Gmail, Google Drive, Google Maps and other Google apps. This service has been launched in beta on 28^{th} june 2011 [25].

2 Common Features of Complex Networks

Complex Networks share similar features, in a way that they relatively have a low degree compared to their maximal degree which is $n-1$ in a graph that contains n nodes, or we can say that those networks are "sparse". Moreover, many real networks are scale-free, small worlds, highly clustered, and have

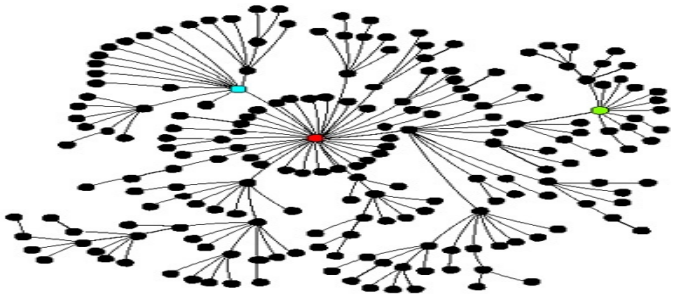


Figure I.7: Example of a Scale-Free Network Graph [28]

community structures. All these are empirical notions. Thus, inherently not very mathematically precise. So, we will try to describe briefly these properties [26].

2.1 Scale-Free (Power Law)

A scale-free, or scale-free network is a connected graph or network has a property that the number of links k originating from a given node exhibits a power law distribution:

$$P(k) \sim k^{-\gamma} \tag{I.1}$$

A scale-free network can be built by progressively adding nodes to an existing network and introducing links to existing nodes with preferential attachement so that the probability of links to a given node i is proportional to the total number of existing links that we have in our graph which is illustrated in Figure I.7. The formula is defined as follows:

$$P(linking\,to\,node\,i) \sim \frac{k_i}{\sum_j k_j} \tag{I.2}$$

Scale-free phenomena occur in many areas in life such as science, engineering or in topology of Web pages, where nodes are individual Web pages and in the other side hyper-links as a links, or the power grid example where nodes refers to generators or transformers, and links are power transmission lines. [27].

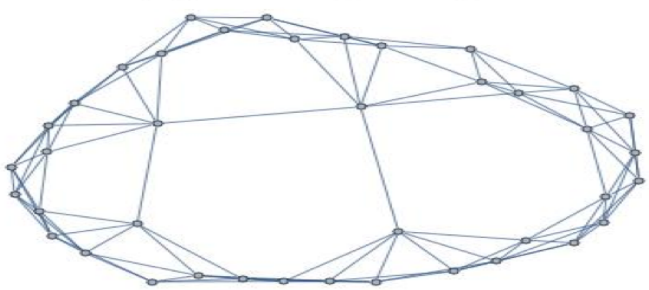


Figure I.8: Small World Graph(Six Degrees of Separation) [30]

2.2 Small World (Six Degrees of Separation)

This property shows that distance between any pair of nodes is small. This property is shared by many real world networks. This notion has been popularized by terms like "six degrees of separation" between any two people in social networks for example, meaning they are typically connected by a chain of no more than six edges where people are considered as nodes and friendship or acquaintance are linked by an edge in the graph [29]. As seen in Figure I.8, we can reach any node in the graph with six or less edges. If you are a Facebook user, then your friendship graph will look something like the previous example.

2.3 Highly Clustered Coefficient

The "highly clustered" property in network science describes how entities in complex systems interact, and argues that the structure of the network influences processing. Clustering coefficient C is one measure of network structure refers to the extent to which neighbors of a node are also neighbors of each other [31].

2.4 Community Structure

Community structure is also one of the main features that characterizes complex networks, in the sense that network nodes are joined together in tightly knit groups, between which there are only looser connections [32]. An

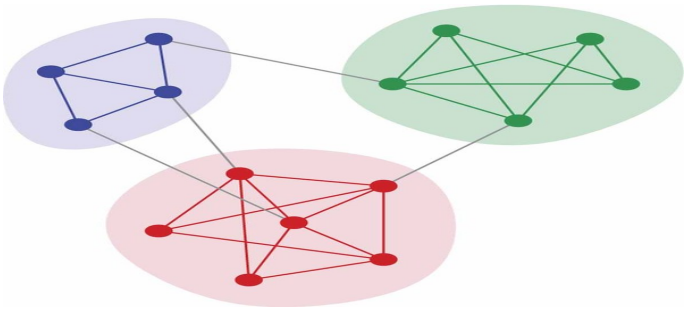


Figure I.9: Example of Community Structure in a Network [33]

illustration of community structure is shown in I.9 .

3 Complex Networks Analysis

Complex Networks are everywhere, as a result, they have attracted considerable interest, also because of the wide range of their applications, sometimes they are studied using classic approaches. An example is the study of a specific system, then, finding its complex network (it can be represented as a graph), next, analyzing the corresponding properties. Many Complex Networks analysis techniques are used nowadays, depending on the purpose of this analysis, we briefly describe some of them starting by Community Detection, then link prediction, and we will finish by centrality and its variants.

3.1 Community Detection

Previously, we saw what does mean Community Structure. Many methods where proposed to solve this problem, one of the best techniques used by community detection algorithms is that of calculating "modularity", used to quantify the quality of a discovered partition in the studied complex network [34].

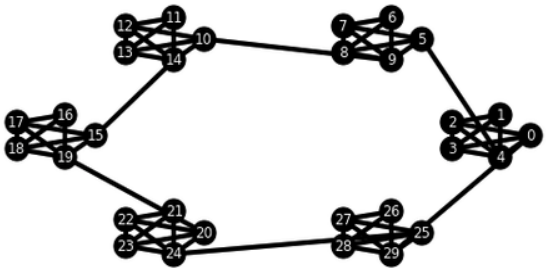


Figure I.10: A Set of Detected Communities

3.2 Link Prediction

In network science, Link Prediction (LP) may be seen as the task of determinating a future link (relationship in Social Networks). Let us take a Social Network as an example, supposing that it is highly dynamic object, as a consequence, it will rapidly change over time through the formation of new edges (relationships), this is due to the appearance of new interactions in the underlying social structure [35].

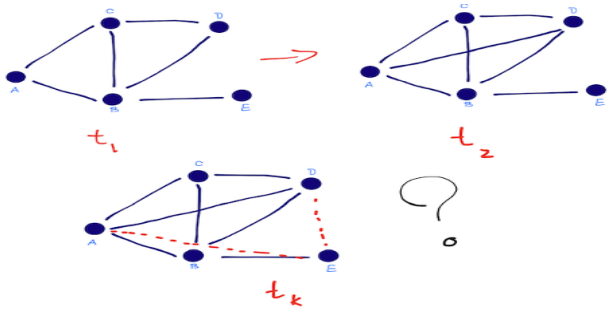


Figure I.11: Social Network's Evolution Between instant $t1$ and $t2$ [36]

Figure I.11 represents a snapshot of a social network in an instant $t1$ and a probable prediction for links that may appear in the next time $t2$ (in reality, it is during the interval of time between $t1$ and $t2$).

3.3 Centrality

Our third important metric for complex network analysis (CNA) is to find (or not) any vertices that are more important than others. When we talk about

"importance" of a vertex, it depends on graph's nature and what is modeling. Let us take a social network as an example, in relationships between people, a vertex (individual) with high degree may mean that he is an influential person in this network [37]. In what follows, we will give a brief explanation of three important centrality metrics: Degree Centrality, Closeness Centrality and Betweenness Centrality.

Degree Centrality "DC"

Degree centrality is based on how many relationships has an individual (or in/out connections in case of oriented network graph). An individual having more ties to others involves that the latter is important. Individuals having many ties, means that they may have access to, and they can call other resources of the network as a whole. It is very simple and very effective measure of a node's (individual/actor) centrality.

Closeness Centrality "CC"

Degree Centrality measure might be criticized because they always take into consideration the immediate ties that an individual has, or it's neighbors' ties, rather than indirect ties to all others. One scenario is if an individual is tied to a large number of others, but those others are actually disconnected from the network as a whole. In this case, the individual will be central but only in its local neighborhood. Here, the Closeness Centrality approaches calculate distances from an individual to all others in the network by focusing on the distance from each one to all others.

Betweenness Centrality "BC"

Betweenness Centrality focuses on individuals when they act as a bridge during a communication (relationship) between two other individuals. Let's imagine that I want a specific stuff from my university's chancellor, here, I must forward my demand passing by my department chair, then, a dean, finally, a vice

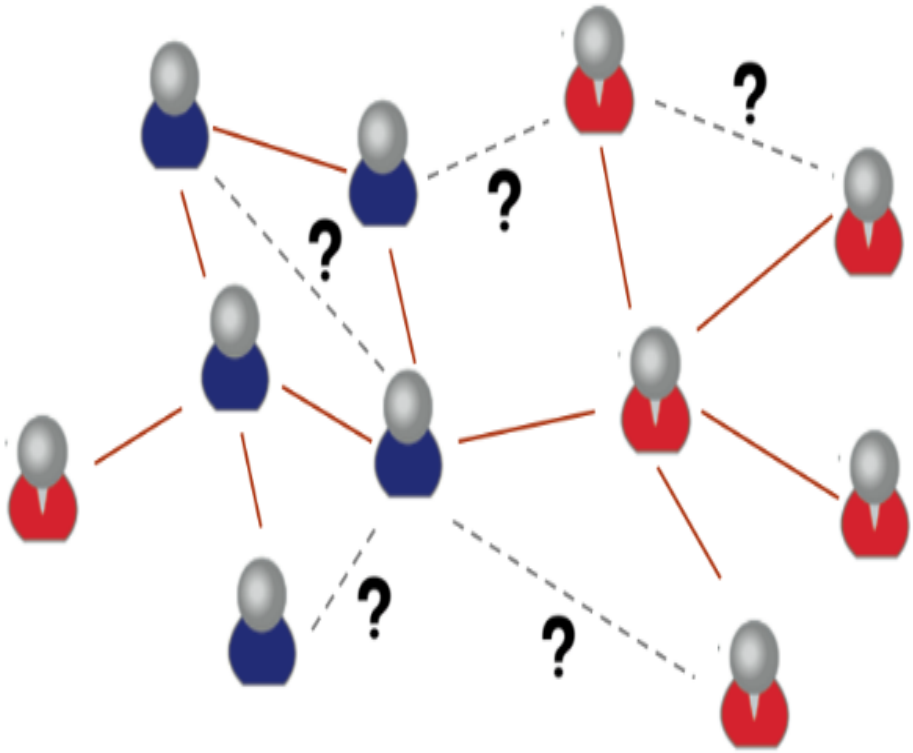
chancellor. Through this path, those persons could (during the demand) delay the demand or prevent it from getting through. This will give them power and strength in this small network.

Conclusion

In this chapter, we have seen what are Complex Networks, some of their examples, some of their common features and analysis that can be applied on them. In the next Chapter, we will present link prediction state of the art.

Chapter II

Link Prediction: State of The Art



1 Link Prediction Problem

Let us imagine we have a Social Network $G(V, E_t)$ at a particular time t, where V is the set of nodes, and E the set of links. As mentioned previously, the Link Prediction task has to predict new links that may appear or disappear in current network for a future time t' where $t' > t$. In Figure II.1 there is an example explaining the situation. It is a Social Network of 5 persons, where thick links represent interactions that are already existing at a given time t. In the other side, dashed links represent links that have recently appeared during the interval $[t, t']$. In this example, at time t, Alice and Bob are already friends, further, Alice and Nick are also friends. Thus, at time t', maybe Alice has introduced Bob to Nick, they (with a certain probability) became friends too. Using the same reasoning, Nick and Amy would become friends at time t'. That is the goal of Link Prediction which is the prediction of new links [3].

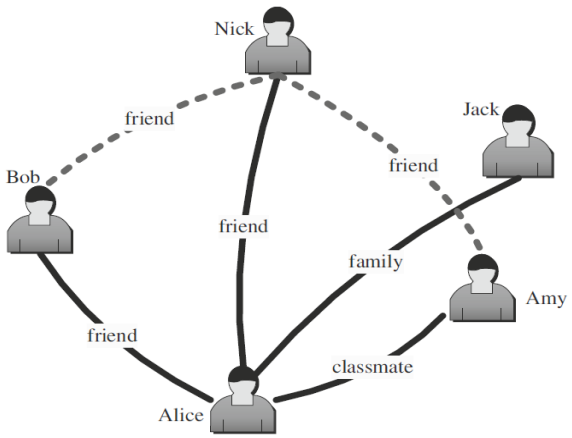


Figure II.1: An Example of Link Prediction Problem at a Given Time "t"

2 Link Prediction Applications

LP concept is wide (see next page), people may think that predictions are done just to know future relations between two Social Network users. It is true, but it is not restricted to that only. Most examples that have a strong

tie with Link Prediction concept are [38]:

- **Friends Recommendation in Social Networks**
  Always in Online Social Networks we see for example some propositions or suggestions of people who we may know and who we may –in the future- be friends with or probable followers. It is referred to the sentence *"you may know these people"*.

- **Reciprocal Relationships Prediction**
  Reciprocal relationship prediction can be defined as a two-way relationship that occurs in a Social Network. It represents the strength of trust between people. A well understood of these two-way relationships will provide a well knowledge of some network characteristics such as the underlying Communities, users' Influence on each other. And more other useful data that will be helpful to deep Network Analysis.

- **Collaborations and Experts Recommendation in Academic Social Networks**
  In Collaboration Networks nodes represent researchers, and edges represent collaborations. LP here, will help a researcher to find the expert that will match his own interest in a particular discipline.

- **Surveillance and Security Domain**
  Here, to control and monitor people, LP will be a sensitive task due to its influence in the of terrorist attacks (terrorist networks). Using LP may help to predict next contacts and plans that terrorists may do, this will be done by analyzing their relations in monitored network.

3 Link Prediction Metrics

There is a lot of techniques that use Link Prediction metrics. An example can be the use of nodes' information, topology or social theory to calculate the similarities of pair of nodes. In this work, we will give a brief review of these metrics starting with Node-based Metrics, then, Topology-based Metrics, and we will see Social Theory based Metrics [3]:

3.1 Node-based Metrics

As the name implies, this approach uses the following idea: "the more similar the pair is, the more likelihood a link appears between them, and vice versa", which means that it focuses on computing the similarity between a pair of nodes in order to predict a future link. This is due to social network nature, where users tend to create relationships with others who share the same education or religion tendencies, interest or location. This similarity measure is done for each non-connected pair of nodes (x_1, x_2) and it assigned a score for these pairs. Thus, it significates the similarity between them (the two nodes x_1 and x_2). High score here means high probability that x_1 and x_2 will be linked in the future (time t'). Actually, nodes have some attributes used to calculate similarity. To clarify the idea, we will take an online social network as an example, where a "profile" can be considered as an attribute for a node, or in email network, where mail name also act as an attribute for any node. This kind of information allows us to directly calculate similarity value between two nodes. While in general, attribute values take a textual form, thus, text-based and string-based similarity metrics are usually used here [2][39].

- Bahattacharyya et al.[39], defined a model "categorization tree model" which aims to study keywords of user profiles. That leads them to define distance between keywords to determine the similarity between a particular pair of users. One important remark appeared, it is that similarity between users is approximately tied without taking topological metrics in consideration. One other remark is that increasing the number of friends and keywords for a user implies lowering the similarity's average between him and his friends.
- Akcora et al. remarked that in current social networks, most user profiles are missing [2]. To solve this problem that affects calculating similarity value, they proposed a method to guess a portion of the missing value(s) of a particular profile before calculating the similarity. This interesting method is focusing on profile information of mutual friends and the majority voting schema.

3.2 Topology-based Metrics

These metrics focus on topological information. They are known as topology-based metrics. We will take a look on some popular and well-known Topology-based Metrics used in Link Prediction, but before doing so, we will set some standard notations by taking a Social Network type as an example, let's consider $G(V, E)$ where:

- G is the graph modeling the Social Network.
- V is the set of vertices (individuals).
- E is the set of edges (relationships).
- x is a node that belongs to V.
- A is the adjacency matrix of our graph G.
- $\Gamma(x)$ is the set of neighbors of the node x.
- $|\Gamma(x)|$ is the number of x's neighbors.

Topology-based Metrics are classified into : Neighbor-based Metrics, Path-based Metrics, and Random walk based Metrics.

3.2.1 Neighbor-based Metrics

In our lives, it is a reality that people always tend to make new relationships with other people who are near them. This fact leads to think that neighbors in a Social Network are the most close ones to a particular user. Thus, many Neighbor-based Metrics were designed for Link Prediction Problem. In this scope, we consider four metrics (Common-neighbors (CN), Jaccard Coefficient (JC), Adamic-Adar Coefficient (AA) and Preferential Attachment (PA)). They will be explained in more detail in the the third chapter.

3.2.2 Path-based Metrics

These metrics focus on "paths" between nodes. To make it clear, we saw previously that calculating similarity between pair of nodes can be done using node's information, or even neighbors' information. But in this kind of metrics, path has the main interest, and a set of methods of Path-based Metrics were done to allow similarity computing between pair of nodes [3]. Among them we have Local Path and Katz:

Local Path

Local Path metric uses information of 2 and 3 path lengths, moreover, Local Path exploits other additional information of neighbors that are far from the current node with length equal to 3. Paths of length equal to 3 are less relevant than nodes of length equal to 2, thus, an adjustment factor α must be assigned and applied in the measure. α must be a small number, close to 0. Local Path's metric is defined by [3]:

$$LP = [A]^2 + \alpha[A]^3 \tag{II.1}$$

Where:

- $[A]^2$ denotes the adjacency matrix of nodes having 2 as a distance length.
- $[A]^3$ denotes the adjacency matrix of nodes having 3 as a distance length.
- LP denotes the adjacency matrix that describes pair of nodes by taking lengths 2 and 3 into consideration.
- α denotes the adjustment factor.

Katz

This (" Exponentially Damped Path Counts" [38]) metric focuses on the whole graph paths, moreover, it counts all paths of pair of nodes. These paths are

damped exponentially by length to consider short paths more heavily than long ones. This measure is defined as follows:

$$Katz\,(x1, x2) = \sum_{l=1}^{\infty} \beta^l . \left|path^l_{x1,x2}\right| = \beta A + \beta^2[A]^2 + \beta^3[A]^3 + ... + \beta^{10}[A]^{10} + ... \quad \text{(II.2)}$$

Where:

- $\left|path^l_{x,y}\right|$ denotes the set of all length l paths from x to y.
- β^l denotes a value exponentially damped by length, also using a very small β involves prediction much similar to CN, this is due to the length, paths of three or upper won't increase the value so much (they contribute only a little to the summation) [3] [38].

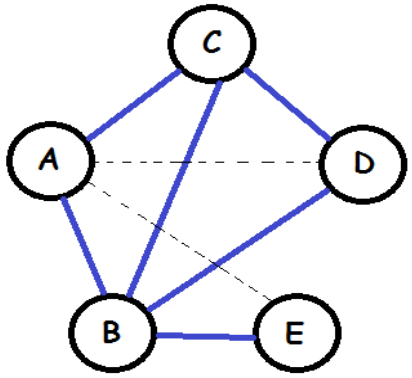


Figure II.2: Graph Example to Apply Katz metric

Figure (II.2) gives an example that clarify Katz's idea, let's take 2 pair of nodes, (A, E) and (A, D), while in (A, E) (using 2 or 3 hops) we have only one path. This involves:
$path^2_{A,E} = path^3_{A,E} = 1$, which means:
$Katz(A, E) = \frac{1}{2} * 1 + \frac{1}{4} * 1 + ...$
And from A to D (using 2 or 3 hops) we have two paths. This involves:
$path^2_{A,D} = path^3_{A,D} = 2$, which means:
$Katz(A, D) = \frac{1}{2} * 2 + \frac{1}{4} * 2 + ...$

3.2.3 Random walk based Metrics

In a social network, social interactions between nodes can be modeled by random walk that uses transition probabilities from a current node to all its neighbors to show us the random walker's destination from that node. And these Link Prediction metrics are used to calculate similarities between nodes using the random walk feature [3]. There is many random walk based metrics, but here, we explain only two metrics:

Hitting Time

The Hitting Time (HT) is the expected number of steps required for a random walk that starts at a node and reaching another one. Random walk mean that it starts from a source node and iteratively moves to one of its neighbors that was chosen uniformly and randomly. So we can now define HT as follows:

$$\mathrm{HT}\,(x_1, x_2) = 1 + \sum_{\omega \in \Gamma(x)} P_{x,\omega} \mathrm{HT}\,(\omega, y) \tag{II.3}$$

- where $P = D_A^{-1} A$ and:
  - D_A denotes the diagonal matrix of A which has as a value : $(D_A)_{i,i} = \sum_j A_{i,j}$
  - $P_{i,j}$ denotes the probability of stepping on node j.

Commute Time

Commute Time (CT) can be viewed as counting the expected steps from a particular node x_1 to another node x_2 and from x_2 to x_1 as well. It is the commute time to go from x_1 to x_2 then back to x_1 again [3]. Commute time between a pair of nodes can be denoted as follows:

$$\mathrm{CT}\,(x_1, x_2) = \mathrm{HT}\,(x_1, x_2) + \mathrm{HT}(x_2, x_1) \tag{II.4}$$

3.3 Social Theory based Metrics

Unlike the previous two metrics (Node-based and Topology-based Metrics) which only benefit from node and topology information, social theory based metrics make efficient LP's results, and this is due to collecting additional useful social interaction information. Also it is useful for large scale Social Networks in the first place. Many works were done last few years using classical social theories. They employed Community, triadic closure, strong and weak ties, and structural balance to solve Social Network issues, such as Social Network analyzing and mining.
In this work, we describe briefly two metrics done by Valverde-Rebaza and Lopes [40], and Liu et al. [41] respectively:

- Valverde-Rebaza and Lopes did a combination between topology and Community information with the help of users' interest and behaviors, then apply it to predict future links that may appear in Twitter. Results show that this new method can enhance the LP performance in a directed and asymmetric large scale Social Network [40].

- In their article, Liu et al. made a proposition for a LP model that focuses on a concept of weak ties and the three node centralities of common neighbors (Degree, Closeness and Betweenness Centralities). According to their Centrality, each common neighbor plays a different role to the node connection. The weak tie has as a purpose : the amelioration of LP accuracy. The model is defined as follows [41]:

$$LCW(x_1, x_2) = \sum_z (w(z).f(z))^\beta, \quad f(z) = \begin{cases} 1, & z \in \Gamma(x_1) \cap \Gamma(x_2), \\ 2, & Otherwise, \end{cases} \tag{II.5}$$

  Where:

  - $w(z)$ denotes the weight of node centrality.
  - $f(z)$ denotes the switch function.
  - β has as a goal : the adjustment of each common neighbor's

contribution to the connection likelihood of two nodes. Using a big β that is greater than 1 will make the larger Centrality more significant than lower Centrality. In the other side, using a small β that is less than 0 gives the inverse effect. If β is between 0 and 1, here it will equally limit all nodes.

4 Link Prediction Techniques

LP is an active research field that let many scientists across the world searching for useful and powerful techniques to solve the LP problem. As a result, a set of techniques were found and gave an acceptable prediction. Each one is used in a particular domain and has its own characteristics. We have chosen to study three of them: Algebraic Method, UnSupervised Learning and Supervised Learning approaches . We will give a glance on the first ones, then we will take a wider look on the two remaining ones:

4.1 Algebraic Method

In a particular network, we don't have to consider immediate neighborhood of only two nodes, but we also can take the whole graph into account, by formulating it using algebraic graph theory, we mean here, the decomposition of graph's adjacency matrix and computing it. Using the new transformed adjacency matrix of a particular graph, Link Prediction methods can be easily defined and learned. Algebraic Link Prediction methods are characterized by their scalability and learnability. They are scalable because they rely on a model, and that model is built once. Hence, it makes fast computations. One other strongpoint is that algebraic models are dedicated for decomposed matrices (for huge networks), so they can be updated and gathered using iterative algorithms. In the other side, local Link Prediction algorithms are memory-based. Thus, they access the adjacency data directly during Link Prediction process [42].

4.2 UnSupervised Learning

We would like to talk (before the Supervised Learning) and briefly explain UnSupervised Learning (UL) . It is one of the machine learning (ML) techniques, which uses datasets and divide them into deferent clusters (using clustering algorithms). One example of clustering in UL is in "google news", where it gather hundreds of news per day, using clustering, will allow google to group them by topic. In addition, UL is used to organize large computer clusters –data center- where we have to predict and figure out which machines can work together, by doing so (putting these machines together), we will make the data center work better and more efficiently. An other example, in Social Network analysis, where it gave us knowledge about which friends we contact them frequently like in Facebook or Google+ where we add friends to circles, this allows google to automatically identify groups of people [43].

4.3 Supervised Learning

The last technique that takes our main interest is the Supervised Learning (SL) one of the ML types as the previous one (UL), which can be seen as the problem of taking labeled datasets (the input), then picking up information from this data in order to label new datasets (unseen datasets). The program doing so is called "learner", it is a function that takes a set of features of an example then performs prediction [44].

4.3.1 Link Prediction as Binary Classification

It is also a classification problem, where we have two or more output classes. Starting with the training datasets which contains some seen situations, then, using these situations will allow us to label other unseen ones. This is done by calculating similarity score using features [45]. In general, SL that focuses on classification uses a function with some features in order to classify an unseen data into one class (labeling it). In general, LP under SL follows a set of steps,

beginning with preparing the labeled datasets, finding training instances and assigning class labels, then using some specific features.

4.3.2 Links as Elements to be Classified

In LP domain, the SL task aims to predict new links by looking up for nodes' similarity, we mean here that used features are gleaned from the node itself, and this needs a knowledge of node's own characteristics (sometimes hard or impossible due to the network privacy). What we are going to do here is something different, unlike calculating node's similarity, we will try to predict links by calculating edge similarity itself. In other words, we are going to inferring new relationships (in a Social Network). This is done by calculating features that characterize each pair of nodes.

4.3.3 Example of Data Representation

We mentioned features that represent our distance vector, also labeled them (either there is a link –yes- or not –no-), and this is done using datamining techniques. This is an important step toward predicting new links that may appear in the next instant *t'*.

Test_0_A_18_51_12_9_5_2016.csv

```
First,Last,Edge,Link,CN,JC,AA,PA
1,2,"(1,2)",1,0.2933333,0.6864608,0.3158572,0.1646342
1,3,"(1,3)",1,0.5866666,0.8218527,0.7103093,0.4756097
1,4,"(1,4)",1,0.4933333,0.7767221,0.5756637,0.3902439
1,5,"(1,5)",0,0.08,0.2422803,0.08331347,0.06097561
1,6,"(1,6)",1,0.4266667,0.7672209,0.4936847,0.3048781
1,7,"(1,7)",1,0.12,0.3349169,0.1210619,0.09146342
1,8,"(1,8)",1,0.4133333,0.7672209,0.4599876,0.2865854
1,9,"(1,9)",1,0.2533333,0.6104513,0.2791573,0.152439
1,10,"(1,10)",1,0.2533333,0.5653207,0.2703398,0.1890244
1,11,"(1,11)",1,0.05333333,0.1638955,0.05123683,0.05487805
1,12,"(1,12)",1,0.4,0.7909739,0.4584624,0.25
1,13,"(1,13)",1,0.2,0.5083135,0.2108098,0.1280488
1,14,"(1,14)",1,0.32,0.695962,0.3445022,0.2012195
1,15,"(1,15)",0,0.1733333,0.4536817,0.1783518,0.1158537
```

Figure II.3: A Data File

In Figure (II.3), we produced a *".csv"* file to make data understandable by other applications.

4.3.4 Data Mining Techniques

Data Mining Techniques, (called also knowledge data discovery KDD), is the process of analyzing data from different sides or angles to find correlation between data used on many fields in relational databases, then transform it into useful information that match our needs [46]. Also, data mining can be seen as a powerful new technology that leads to the extraction of hidden predictive information from large databases. Many Data Mining Techniques are used in this scope, these techniques are implemented in the aim of enhancing the value of existing information resources [47]. Data Mining Techniques provide answers in LP under CplxNs discipline field such as: *"in a future instant, in a particular Social Networks, who may make new relationships, and why?"*.
In this work, we will see two datamining techniques (algorithms), Decision Trees and k-nearest neighbor . A detailed explanation will be in the third Chapter. Also, these algorithms are used in the experimental part (using Weka data mining tool (see Appendix D). We have chosen them due to their simplicity.

Conclusion

In this Chapter, we have seen the Link Prediction Problem and some of its metrics and techniques. In the next Chapter we will explain our Approaches used to achieve the Link Prediction goal.

Chapter III

Our Link Prediction Approaches

1 Global View of Our Approaches

Our approaches of link prediction focus on machine learning techniques, we proposed in this scope two approaches, one is supervised (supervised learning link prediction "SLLP") based on classification and the other one is unsupervised (unsupervised learning link prediction "ULLP") based on clustering. Both approaches are described as follows:

1.1 Supervised Approach

As seen in the state of the art chapter, SL is one of the learning machine techniques. In our design, we use features to characterize a link as it will be described next. The learner is built using classification technique in which discrete class assignment (label, either link or not) is performed. A Supervised Learning algorithm analyzes the training data and produces an inferred function which is called *"classifier"* [48].
Figure III.1 sketches our SL approach:

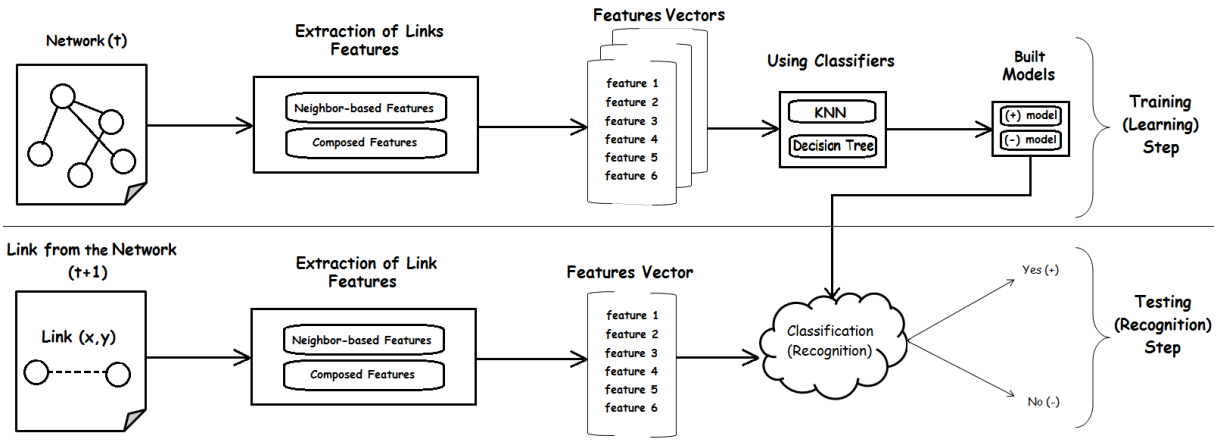


Figure III.1: Illustration of Our Supervised Approach

Once the features extracted from the network, we will use a vector representation in which each pair of nodes is represented by a vector of features. This step is similar for UnSupervised Learning approach.

1.2 UnSupervised Approach

UnSupervised Learning is also one of the machine learning techniques. Link Prediction under UL (ULLP) can be done also by applying a ranking strategy by taking some measures and metrics like what we have described previously in LP Metrics. However, features are taken into consideration without any knowledge on their labels (the output). Thus, UL has to organize and categorize data according to their similarities.
In our work, the similarity is measured between pair of nodes in a Complex Network.
by taking the *"k"* first elements we can predict new links that may appear in the future [49].

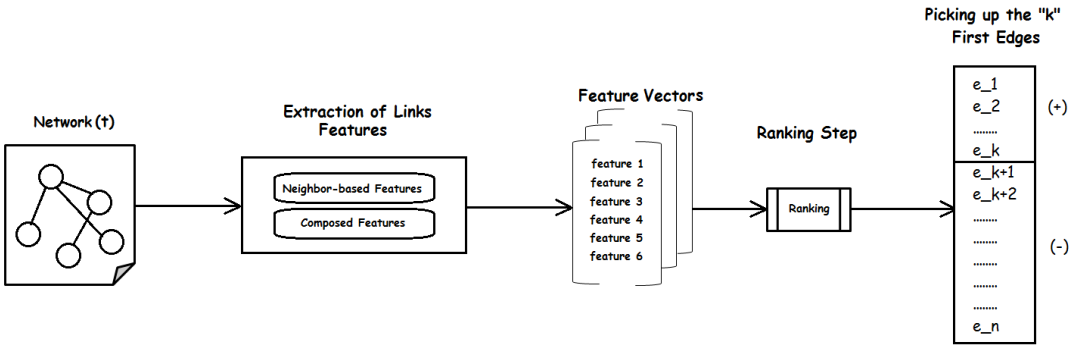


Figure III.2: Illustration of Our UnSupervised Approach

Figure (III.2) illustrates ULLP approach. It shows that by taking a network in an actual time (t), it allows us to extract edge features focusing on either neighbor-based or combined features (will be described next), then, doing the ranking task on the chosen feature, after that, picking up the "k" first edges and supposing them as future links.

2 Which Features?

Many types of features can be used. From the Topology-based, they can be either Neighbor-based, Path-based or Random walk based Metrics

features. We chose Neighbor-based features to achieve our LP purpose. We chose them according to their extraction simplicity and their brief time consuming. In fact, this is an important factor when we are dealing with big Networks. These features are namely Common-neighbors (CN), Jaccard Coefficient (JC), Adamic-Adar Coefficient (AA) and Preferential Attachment (PA). In addition, we have added some other ones (Z and Z_c) that represent the combination of Neighbor-based features.

For SLLP, we use the neighbor-based features in a Gathered form. Features are not always normalized, the normalization is performed to avoid specific features' promotion. In order to normalize them, we try to make all features between 0 and 1 scale (relatively with the original values). Also, we have to weight features to get an efficient prediction Precision (this is a task of objective function maximization).

Let us return to the adapted features, we use some metrics to characterize a pair of nodes (both + and -). This is done by three ways (Single, Combined and Gathered ones) as follows:

2.1 Single Features

Here, we characterized pair of nodes using one of the "Neighbor-based Metrics", we are talking about CN, JC, AA, PA separately.

2.1.1 Common Neighbors

Due to its absolute simplicity, we can say that CN metric is the most popular metric used in LP [38]. By considering two nodes x_1 and x_2, CN is the number of nodes that both x_1 and x_2 have a direct (one link) interaction with. The CN measure formula is defined as follows:

$$CN(x_1, x_2) = \mid \Gamma(x_1) \cap \Gamma(x_2) \mid \tag{III.1}$$

where $\Gamma(x)$ represents x's neighbors. One of the weak points of this measure is represented in the relative reflection of similarity between pair of nodes, its time complexity equal to $O(n^2)$ [3].

2.1.2 Jaccard Coefficient

As we saw previously, *CN* is an interesting metric, but it doesn't take care of common neighbors' size, Here, *JC* came to make normalization for the size problem. It gives higher values to pair of nodes which share higher proportion of common neighbors, it works with the total number of neighbors they have, its time complexity is equal to $O(2n^2)$ [3]. The *JC* measure is defined as follows:

$$JC(x_1, x_2) = \frac{\mid \Gamma(x_1) \cap \Gamma(x_2) \mid}{\mid \Gamma(x_1) \cup \Gamma(x_2) \mid} \tag{III.2}$$

where $|\Gamma(x_1) \cap \Gamma(x_2)|$ represents the number of common friends, and $|\Gamma(x_1) \cup \Gamma(x_2)|$ represents the total number of friends [38]. Here is an example that may clarify the idea, let's imagine that we have node x_1 that have 5 as a total number of neighbors and share 4 neighbors with another node x_2 (which has 7 as a total number of nodes), and let's take also two other nodes X_3 and x_4, the values are:

Table III.1: *CN* and *JC* Illustration

	Total Friends Number	**Common Friends**	**CN**	**JC**	**Observation**
x_1	5	4	4	0.42	a reasonable result for CN and JC
x_2	7				
x_3	75	30	30	0.18	an unreasonable result for CN
x_4	96				

We see that *CN* does not reflect the real similarity, where it gives a high value in the case of high common friends (x_3 and x_4) without taking all their neighbors in consideration, in the other hand, we saw that *JC* -using normalisation- gives a better evaluation that reflects the reality.

2.1.3 Adamic-Adar Coefficient

One other metric is *AA*, it is proposed by Adamic and Adar, in the purpose of computing similarity between two Web pages historically [3]. Now, it is widely

used in Social Networks. This measure is formulated according to Jaccard's Coefficient. One different thing is that common neighbors which have fewer neighbors are weighted more heavily than the others. Many neighbors only doesn't mean that they have high similarity value, like in Social Networks, for example in Facebook, users that make 1000 friendship relations are not like those who make only 20. The formula is mentioned below:

$$AA\left(x_1, x_2\right) = \sum_{z \in \Gamma(x_1) \cap \Gamma(x_2)} \frac{1}{log|\Gamma(z)|} \tag{III.3}$$

2.1.4 Preferential Attachment

In Social Networks, users who have many friends tend to create more connections in the future, from here, PA metric emerged. It is under the rule "*rich get richer*" in Social Networks. This metric estimates how rich two individuals are by calculating the multiplication between the number of their friends or followers. One advantage is that similarity value does not require node neighbors information, thus, it has the lowest computational complexity.

$$PA(x_1, x_2) = |\Gamma(x_1)|.|\Gamma(x_2)| \tag{III.4}$$

2.2 Combined Features

As Single Features present separate information on links, we have characterized edges using Combined metrics. We used in this scope two new metrics. The first one, so-called "Z", it is the sum of CN, JC, AA and PA as follows:

$$Z = CN + JC + AA + PA \tag{III.5}$$

The second feature, more general, so-called "Z_c" (For Z customized), it is customized feature in which we weighted features used (in the range [0..1] real values). It is defined as follows:

$$Z\_C = \alpha * CN + \beta * JC + \gamma * AA + \Delta * PA \tag{III.6}$$

By varying these coefficients, we obtained a promising results (vs. normal "Z"). We can also get Z by making all coefficients equal to 1.

2.3 Gathered Features

In the previous feature formulation, we use a feature alone. Here, the four features will be gathered as a vector (features vector). These four features are taken from the Neighbor-based Metrics. In other words, the Simple Features are Gathered (and can be used only in the SLLP approach). We have called this characterization: Z_All (where "All" refers to our simple Neighbor-based features).

3 Which ML Techniques?

In order to built models,we use many techniques. For the SL approach we use Decision Trees and KNN. For UL approach we use Ranking Strategy. These three techniques will be explained in what follows:

3.1 Decision Trees

A Decision Tree (DT) uses classification or regression models in a tree structure form. It divides datasets into smaller subsets, in the same time, an associated DT is incrementally developed. This will produce as a result a tree with decision nodes and leaf nodes. A decision node has two or more branches (depending on the dataset and its results). Leaf node represents the taken decision or the appropriate class. The topmost node in a DT is called root node, also, it corresponds to the best predictor in this tree. The core algorithm behind DT is named *"ID3"* done by J.R.Quinlan which employs a top-down greedy search that explore the possible branches without backtracking [50].

Figure III.3: A Generated DT From a Weather Dataset Example [50]

Figure III.3 represents a weather dataset example which contains attributes like outlook, temperature, humidity and windy, these are predictors for the question: *"play golf or not?"*, by analyzing data, the DT can be built, then, it will help in taking decisions (either yes or no).
In our work, the used attributes are the four Simple (Single) features employed separately (each time one feature is used) or Gathered (all features will appear exactly like the given example) or Combined (a sum of features).

3.2 Nearest Neighbor Method

Nearest Neighbor Method (KNN) classifies each tuple or record in a dataset focusing on classes' combination of the k most similar records in its historical dataset. Also, it is sometimes called the "*KNN technique*" [47]. So we can say that KNN is a simple and basic algorithm that stores all available cases and classifies new cases based on similarity measures (like distance functions and hamming distances). KNN has been used in statistical estimation and pattern recognition (in about 1970) as a non-parametric technique [51].

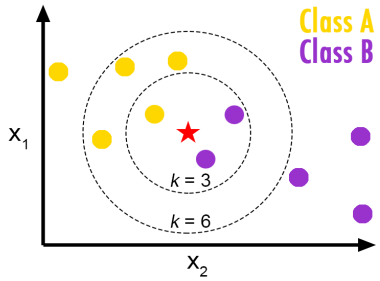


Figure III.4: KNN example in 2-dimensional feature-space [52]

Figure (III.4) illustrates a data set represented in an (x, y) axis (2 features because the space is 2-dimensional). There is a training set consisting of 10 instances (seen data) and 2 distinct classes A and B. Yellow ones for A class, and purple ones for B class. The unlabeled observation (unseen data) is the red star. KNN aims to classify the red star in either A or B class, by looking to its "3" (the value of k) nearest neighbors, KNN had placed the red star as a B-Class, because there is 2 B-class neighbors among 3. In the case of $k = 6$, it was classified as an A-Class because there is 4 A-class neighbors among 6) [52].

3.3 Ranking Strategy

As we said previously in the ULLP approach, Ranking Strategy sorts features' values, then it considers first pair of nodes and it will predicts that they will be links in the future. It works with Simple Features and Composed ones.

Chapter IV

Experiments

In order to evaluate our approaches, we have performed a range of experiments. The aim of these experiments is to measure the effect of different ML techniques, KNN and DT in the SLLP approach, and Ranking in the ULLP approach, and the different combinations of features (alone, Gathered in the SL, or Combined).
Our main script deals with all these techniques, and has the task of preprocessing the network benchmarks.
Before showing and commenting our results, let us describe the context of the experiments which are explained as follows:

- The chosen programming language for implementation.
- The employed ML toolkit.
- The used performance metrics.
- Benchmark of networks

We mention that we have automated the almost of these tasks (It is described in Appendix B).

1 Visual C# (Sharp)

C# ("see sharp") is one of the .Net programming languages. It is also object-oriented language and allows us to build reusable components for wide variety of applications. It was introduced by Microsoft in 2000. Moreover, C# is an evolution of the C and C++ languages family. However, it uses features from other programming languages, such as Delphi and Java [53]. Some of the used algorithms like calculating Simple, Combined and Gathered Features (extracting features in general) and normalization are shown in Appendix A. Also, the full description of our application's functionality is explained in Appendix C.

2 Classification Toolkit

Weka is a collection of machine learning algorithms used for data mining tasks. It also contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. In our case, we are going to use the two first tools (pre-processing and classification) [54].
We explain briefly how to use Weka in Appendix D.

3 Used Metrics

3.1 Confusion Matrix

In LP discipline, there is a way to represent classifier's results, it is called Confusion Matrix [55]. In our work that handle the existence or not of a particular link in the future, we have only two classes (we are talking about Binary Classification problem). Our Confusion Matrix will use the following table:

Table IV.1: Truth Table Confusion Matrix

		Appeared in the Future as	
		Positive	Negative
Predicted as	Positive	True Positive (TP)	False Positive(FP)
	Negative	False Negative(FN)	True Negative(TN)

In Table IV.1, a classifier's results will – absolutely- fall into one of these four cells, and this is done by knowing the actual predictions (instance $t1$) and the new network snapshot (instance $t2$). The four cells' components are described as follows:

- **TP:** True Positive is the correct prediction about the appearing of a new link in the future.
- **FP:** False Positive is the incorrect prediction about the appearing of a new link in the future.
- **FN:** False Negative is the incorrect prediction about the non-appearing (not to be confused with disappearing because we suppose that we don't have lost links) of a new link in the future.
- **TN:** True Negative is the correct prediction about the non-appearing (not to be confused with disappearing) of a new link in the future.

Figure IV.1 illustrates this 4 situations:

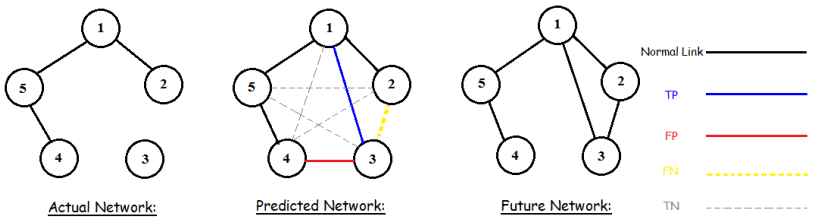


Figure IV.1: An Example of Links Types in a small network

3.2 Performance Measures

Performance Measures are very important and useful to say that our predictions are either good, bad or acceptable. It also gives information

about the correctness of the used classifier for an unseen dataset. According to these measures we can confirm that the classifier can be used or not on another datasets from this kind. We explain some commonly used Performance Measures: Classification Accuracy, Precision, Recall and F-measure [55].

Classification Accuracy

Classification Accuracy is one of the simple Performance Measures, it can be seen as the number of correct made predictions (TP and TN) divided by the total number of all predictions made (TP, TN, FP and FN):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{IV.1}$$

Precision

Precision is the number of True Positives (TPs) divided by both the number of True Positives (TPs) and False Positives (FPs). In other words, it is the number of positive predictions divided by the total number of positive class predicted values:

$$Precision = \frac{TP}{TP + FP} \tag{IV.2}$$

Recall

Recall or Sensitivity is the number of True Positives (TPs) divided by the number of True Positives (TPs) and the number of False Negatives (FNs). In other words, it is the number of positive predictions divided by the number of positive class values in the test data:

$$Recall = \frac{TP}{TP + FN} \tag{IV.3}$$

F-measure

The F-measure or F Score is also one of the Performance Measures, it reports the balance between the Precision and the Recall:

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{IV.4}$$

4 Benchmark Networks Selection

In the aim of testing and performing the LP tasks (SLLP and ULLP), we have chosen two datasets from the konect.uni-koblenz.de Website [56], which is a collection of network datasets project in the area of network science. It contains a lot of data that can be modeled as networks. Thus, it provides us with the possibility of studying them systematically. It contains 160 network datasets (directed, undirected, weighted...).
These datasets are represented as text files, each file contains pair of nodes in a single line that contains the first and the second nodes followed by a timestamp. From these datasets we have chosen two benchmarks: Haggle and Hypertext 2009.
Before dscribing these benchmarks, let us explain the used strategy to get our time division. In fact, we have split edges into two times, t and $t+1$, we have taken the first third as an actual network (t), and the remaining two thirds as a future network ($t+1$), any existing edges represent the positive class (+), in the other case, we have completed and assigned their class as negative (-). Figure IV.2 illustrates this splitting strategy.

Also, while our task considers new links that may appear. We put some hypothesis for this version of LP approaches. We assumed that we do not have new vanishing vertices (static), and we do not have vanishing edges too, thus, an edge that appears in t will still remain at the "t+1" network. Then we added characteristics for both (+) and (-) edges.

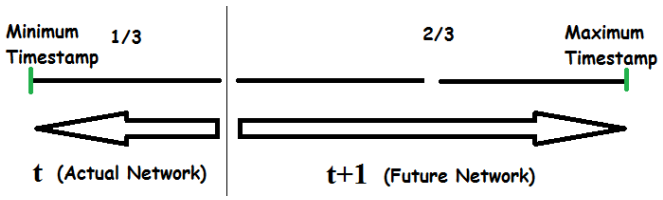


Figure IV.2: Strategy of Splitting Datasets (Actual and Future Instances)

4.1 Haggle Network Dataset

The Haggle Network Dataset (HG) is an undirected network in a TSV (Tab Separated Values) file, that represents contacts between people (person-person contacts) measured by carried wireless devices. A node here represents a person, an edge (link) between two persons shows that there was a contact between them. It was gathered over 5 days [57].

4.2 Hypertext 2009 Network Dataset

The Hypertext 2009 Network Dataset (HY) is a face-to-face (visitor-to-visitor) contacts of the the ACM Hypertext 2009 conference attendees. This conference was held in Turin, Italy over 3 days (from 29 June to 1 July 2009). It is a network in a TSV file, where nodes represent conference visitors, and edges represent face-to-face contacts (each one was active for at least 20 seconds). Multiple edges denote multiple contacts. Each edge is annotated with the time when contact was done [58].

Table IV.2 gives more details and characterizes both chosen datasets. We have extracted these features using our own scripts.

Table IV.2: Technical Sheet of the Used Datasets

	First Dataset	**Second Dataset**
Dataset Name	Haggle Network	Hypertext 2009 Network
Number of Vertices	274	113
Number of Links (+)	1540	946
Numer of non-Links (-)	35861	5382
Number of Future Links	584	1250
Minimum Timestamp	20733 (Day Number 1)	1246255220 (Mon, 29 Jun 2009 06:00:20 GMT)
Maximum Timestamp	364094 (Day Number 5)	1246467560 (Wed, 01 Jul 2009 16:59:20 GMT)
Splitting on	135187 "1/3" (Day Number 2)	1246326000 "1/3" (Tue, 30 Jun 2009 01:40:00 GMT)

5 Results and Discussion

Afer describing the experiment context, this section is dedecated to explain and comment our results. Indeed, we have performed a set of experiments that aim to measure the performance of our LP approaches in the following situations:

- Characterizing links by features taken alone, Gathered or Combined.
- The impact of a chosen classifier on performances in the SLLP approach.
- Comparing the performance of SLLP vs. ULLP.

In more details, this set of experiments aim to achieve the following points:

For ULLP approach:

- Measuring the effect of features taken alone.
- parameterizing Z_c coefficients.
- Comparing between Single Features, Z and Z_c.

For SLLP approach:

- Measuring the effect of features taken alone and gathered (Z_All).
- Forming Z_c after setting its coefficients.
- Comparing between the 7 previous features.

- Comparing between the used classifiers (DT vs KNN).

SLLP vs. ULLP:

- Confirming that SLLP is better than ULLP.
- Comparing between Simple Features, Z and Z_c for both datasets in DT and KNN and Ranking approaches.
- Determining if our suggestion about making a Composed Feature (Z_c) with coefficient had improved (or not) the LP results in DT and KNN and Ranking approaches.
- Determining which is the best approach and the best feature.

5.1 ULLP Approach

Here, first performed the experiment using the ULLPapproach. Figure IV.3 shows our Graphical User Interface (GUI) for choosing the feature and the first number of pair of nodes.

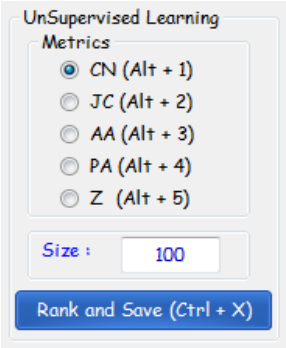


Figure IV.3: ULLP Settings Menu

In what follows, we report the performances of obtained results by ULLP approach for HG then HY (see Table 13 in Appendix E) where "k" column represents the considered pair of nodes that are supposed to be linked among the 100 first non-linked pair of nodes. In what follows, all charts are created from tables in Appendix E.

5.1.1 ULLP Results for HG

Figure IV.5 shows curves of these previous data. In term of F-measure, *PA* gives the best results for all "k" values, followed by *Z_c*, while the worst case is observed for *JC* with a "0" value for all given "k" values (remark that by rising "k" over 87 we observed that performances became higher than "0" value).

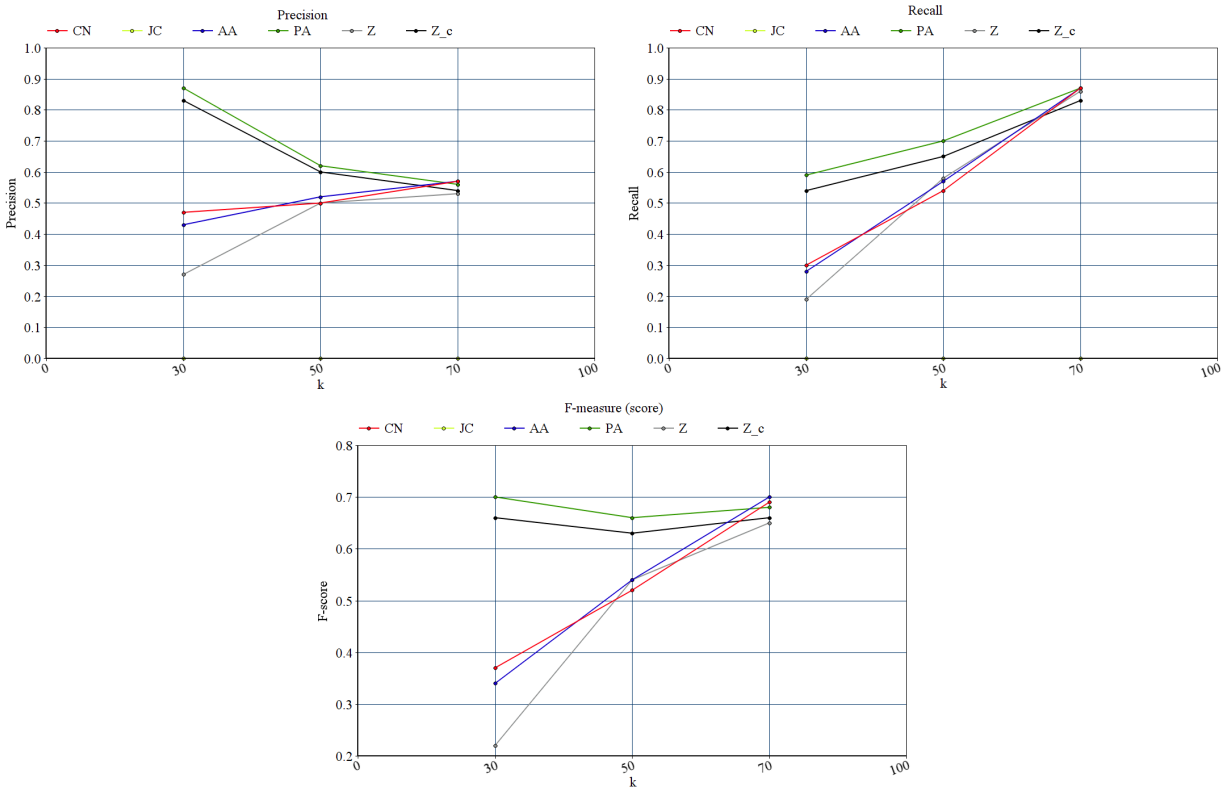


Figure IV.4: ULLP Performance Measures (HG)

Concerning *Z_c*, we have fixed (α, β, Γ, Δ) coefficients according to the obtained results from the Simple Features taken alone. We gave the feature one two thirds $\left(\frac{2}{3}\right)$ from (1), and from the remaining fraction, we gave also two thirds until the last feature which takes the remaining value. In our case, we got 0.66 for the first, 0.22 for the second, 0.08 for the third and 0.04 for the last (and this process is used also in the SLLP approach to form *Z_c*). The chosen coefficients are shown in Figure IV.5.

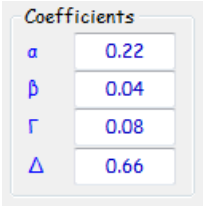


Figure IV.5: ULLP Setting Coefficients for Z_c (HG)

Table IV.3 shows the average of used features. We varied the value of k ($k = 30, 50, 70$) evaluate each feature in the ULLP approach.
Also we made a bar chart that reflects the same information.

Table IV.3: ULLP Performance Measures Average (HG)

	Precision	**Recall**	**F-measure**
PA	0,68	0,73	0,68
Z_c	0,65	0,67	0,64
CN	0,51	0,57	0,53
AA	0,51	0,57	0,52
Z	0,43	0,54	0,47
JC	0	0	0

From Figure IV.13 it is clear that PA and Z_c are the best features, followed by CN, AA and Z. While JC came in the last place with "0" which mean that it is not a good feature in such dataset.

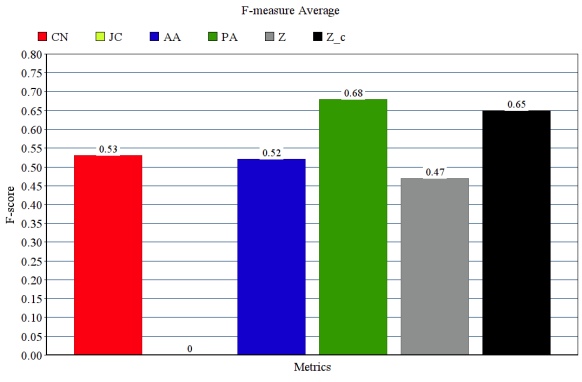


Figure IV.6: ULLP F-measure Average (HG)

5.1.2 ULLP Results for HY

Figure IV.8 shows that In term of F-measure, *AA* comes in the first place, followed by *Z_c*, *Z*, *CN* and *PA*, while *JC* (like in HG) comes in the last place.

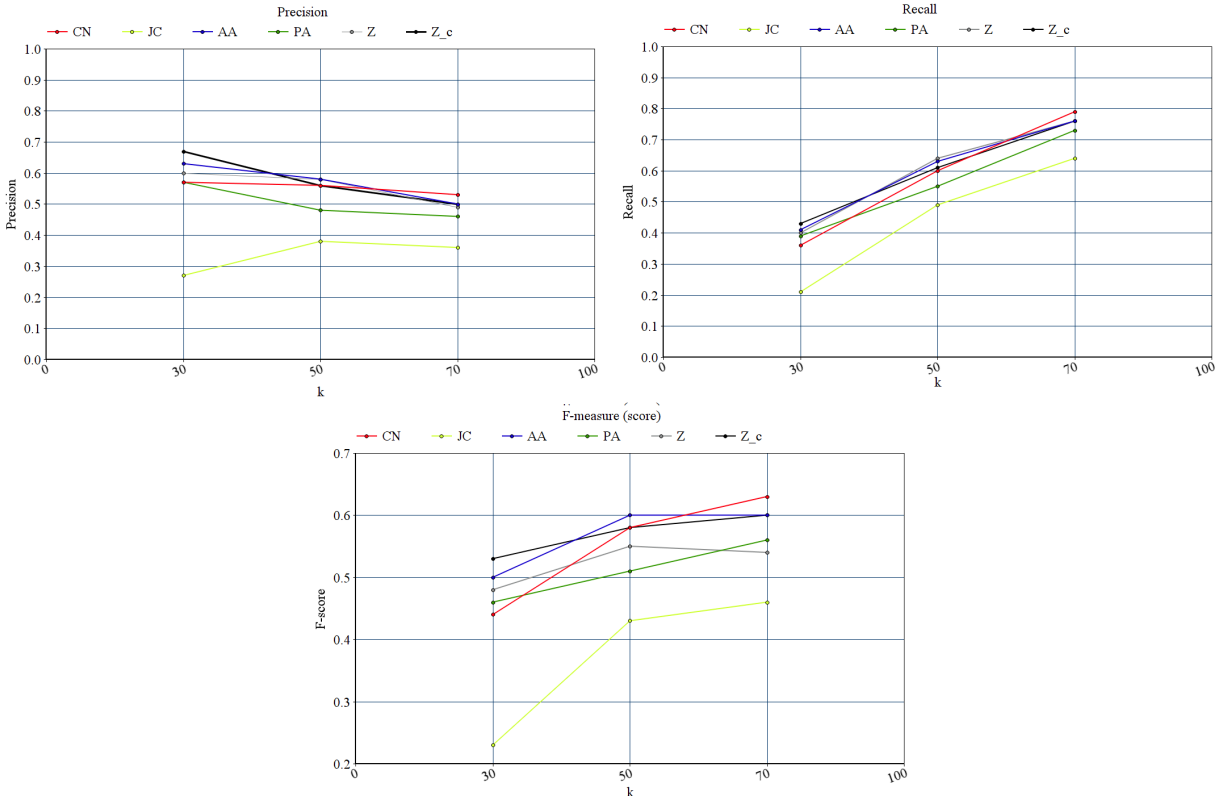


Figure IV.7: ULLP Performance Measures (HY)

Concerning *Z_c*, we have also performed a parameterizing for its coefficients according to the results of ULLP obtained when taking features alone. They are taken as in Figure IV.8.

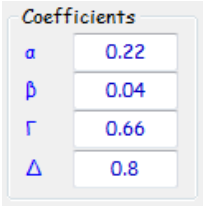


Figure IV.8: ULLP Setting Coefficients for Z_c (HY)

Table IV.4 reports the average of used features where we varied the value of k for HY in the ULLP approach.

Table IV.4: ULLP Performance Measures Average (HY)

	Precision	**Recall**	**F-measure**
AA	0,57	0,60	0,57
Z_c	0,57	0,57	0,55
CN	0,55	0,58	0,55
Z	0,56	0,52	0,53
PA	0,50	0,55	0,51
JC	0,33	0,44	0,37

Also we made a bar chart that reflects the same information. From Figure IV.15, we observe that both *AA* and *Z_c* give good results in HY, while *JC* is the worst one among all features (like in HG).

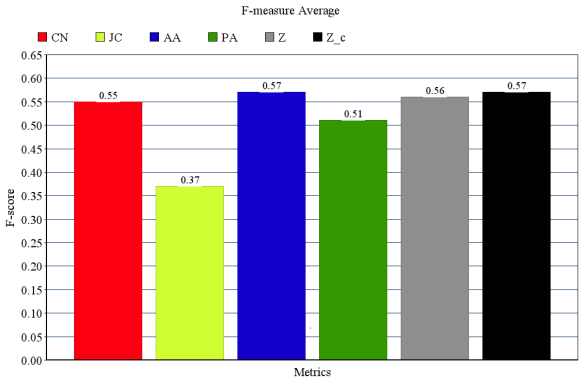


Figure IV.9: ULLP F-measure Average (HY)

5.1.3 ULLP Datasets Results

Let us now evaluate our SLLP approach for both datasets. To get these results, we have computed the average values of performance measures for each feature. The obtained results are shown in Table IV.5. We also represented these data using a bar chart.

Table IV.5: ULLP Performance Measures Average (HG and HY)

	Precision	**Recall**	**F-measure**
Z_c	0,62	0,64	0,61
PA	0,59	0,64	0,60
AA	0,54	0,59	0,55
CN	0,53	0,58	0,54
Z	0,49	0,57	0,52
JC	0,17	0,22	0,19

From Figure IV.16, it is clear that our Composed Feature Z_c comes in the first place which means that combining features enhances ULLP's performances. Also, PA is a good feature for such networks and such approach, while JC is not suggested to be used due to the low obtained performance measures.

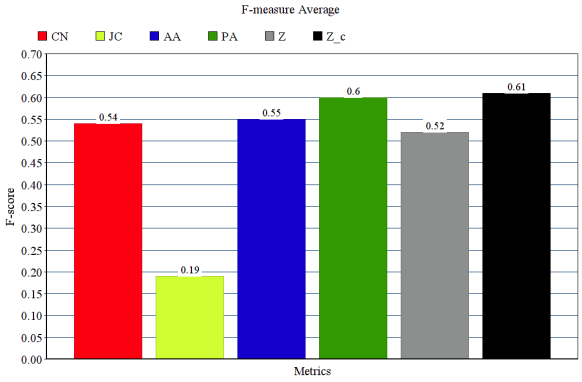


Figure IV.10: ULLP Performance Measures Average (HG and HY)

5.2 SLLP Approach

In this part, we perform experiments using the SLLP approach. We have performed our splitting as defined in "Benchmark section". Table IV.6 reports details of this splitting:

Table IV.6: SLLP Details on Train and Test Sets

	Haggle Network	**Hypertext Network**
Training Set's Length	3080 (1540 + , 1540 -)	1892 (946 + , 946 -)
Testing Set's Length	1168 (584 + , 584 -)	2500 (1250 + , 1250 -)
Picked Up Tests Length Each	100	100

Figure IV.11 shows a snapshot of our GUI for choosing one feature to be used in ranking, and the first number of pair of nodes.

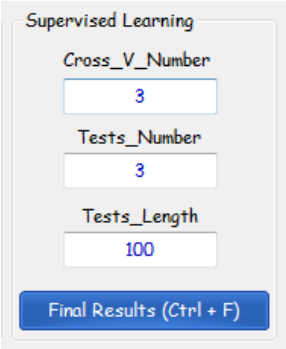


Figure IV.11: SLLP Settings Menu

In order to validate our results, we have used the k-fold technique (k=9). For balancing class purpose, we have selected an equal number of (+) class and (-) class. We also reassured that the test set is disjoint with the train set (pair of nodes of future and actual networks). For all obtained results (see Appendix E).

5.2.1 SLLP Results for HG

By analogy to the ULLP, here, we have also parameterizing Z_c coefficients. Figure IV.12 reports the fixed coefficients (using the same strategy as in ULLP).

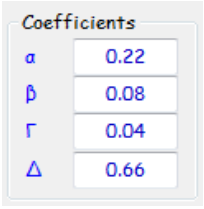


Figure IV.12: SLLP Setting Coefficients for Z_c (HG)

Table IV.7 represent results of performance measure for each feature gotten from "Weka" tool after loading train and test data produced by our application. The obtained results are from two classifiers, DT (J48), and KNN (IBK). We also represent this data as bar chart.

Table IV.7: SLLP Performance Measures Average (HG)

	J48 (Decision Tree)			**IBK (KNN/k-Nearest Neighbor)**		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
CN	0,955	0,617	0,747	0,949	0,622	0,749
JC	0,789	0,736	0,758	0,791	0,666	0,720
AA	0,946	0,635	0,757	0,953	0,539	0,686
PA	0,926	0,780	0,844	0,912	0,790	0,846
Z	0,747	0,775	0,745	0,848	0,705	0,768
Z_All	0,846	0,931	0,871	0,969	0,953	0,960
Z_c	0,783	0,909	0,832	0,913	0,844	0,876

Figure IV.13 shows that when we proceed Single Features it gives good results, and using Composed Features (Z_c) gives better results. Moreover, using Gathered Features (Z_All) enhances the obtained results for the best in both DT and KNN. We also remark that all these results are acceptable (compared to other works where it is about 70% [59]).

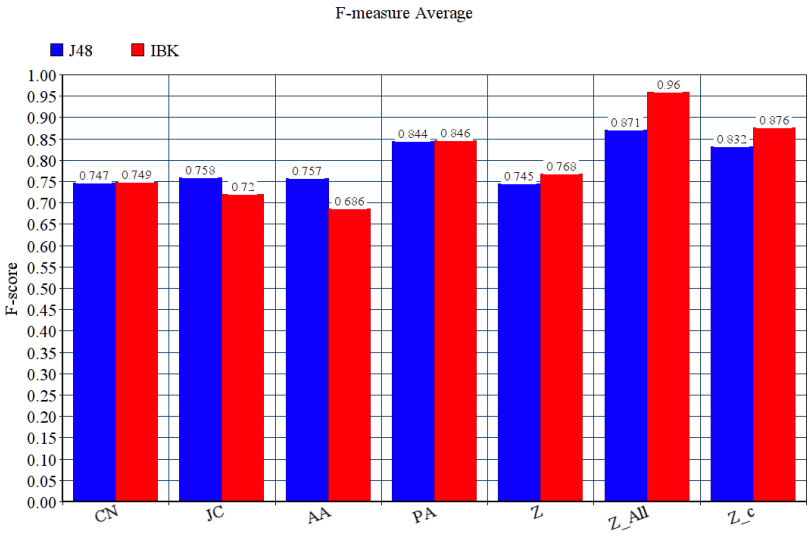


Figure IV.13: SLLP F-measure Average (HG)

5.2.2 SLLP Results for HY

As we did before in HG we have also parameterized *Z_c* coefficients. Figure IV.14 shows the fixed coefficients (using the same strategy, where a good feature will take a higher coefficient as explained before in the ULLP approach).

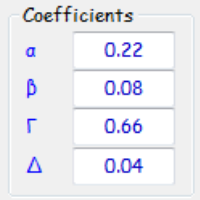


Figure IV.14: SLLP Setting Coefficients for Z_c (HY)

Table IV.8 shows that *Z_c* comes on the top, it achieved a high value in the DT technique, but it achieved the worst one in the KNN technique. Also, we remark that *Z_All* did not achieve a high score, while *Z* gave a good and balanced results in both DT and KNN. Other features are various but we can say that they came with acceptable results (compared to other works done in this scope [59]).

Table IV.8: SLLP Performance Measures Average (HY)

	J48 (Decision Tree)			IBK (KNN/k-Nearest Neighbor)		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
CN	0,573	0,978	0,722	0,571	0,979	0,721
JC	0,608	0,875	0,717	0,585	0,915	0,712
AA	0,603	0,954	0,738	0,581	0,944	0,718
PA	0,554	0,998	0,710	0,567	0,985	0,718
Z	0,610	0,935	0,737	0,598	0,960	0,737
Z_All	0,580	0,977	0,726	0,570	0,971	0,717
Z_c	0,757	0,784	0,750	0,816	0,604	0,692

Figure IV.15 shows the same information given by Table IV.8. Where it is clear that *Z_c* gives the highest value in the DT technique, and it gives the lowest one in the KNN technique.

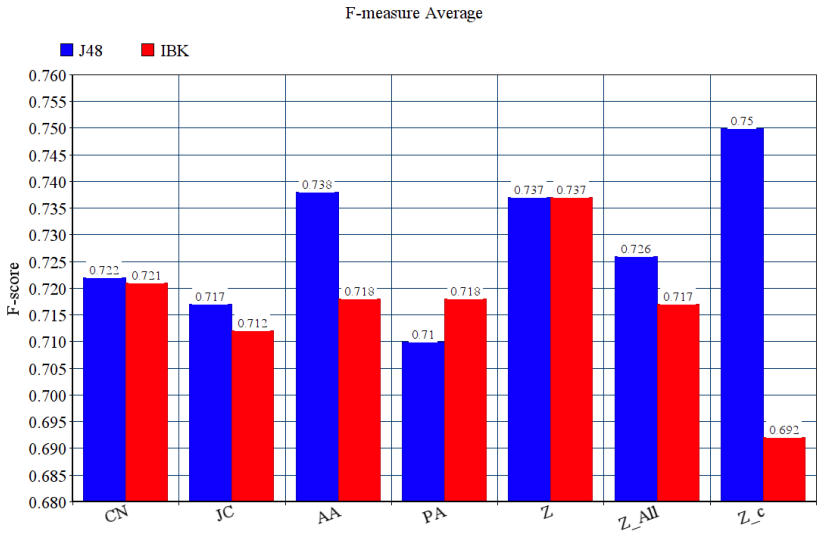


Figure IV.15: SLLP F-measure Average (HY)

5.2.3 SLLP Datasets Results

Now, it is time to measure performances achieved with both datasets. To do so, we have calculated average of these datasets for all features and both classifiers. The obtained results are shown in Table IV.9. Also, we made a bar chart in the aim of visual observations. We remark that in both DT and KNN, *Z_All* has got the best results followed by *Z_c* which comes in the

second place, then we get the remaining features with acceptable results.

Table IV.9: SLLP Performance Measures Average (HG and HY)

	J48 (Decision Tree)			IBK (KNN/k-Nearest Neighbor)		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
CN	0,764	0,798	0,734	0,760	0,801	0,735
JC	0,698	0,806	0,738	0,688	0,790	0,716
AA	0,774	0,795	0,747	0,767	0,742	0,702
PA	0,740	0,889	0,777	0,739	0,887	0,782
Z	0,678	0,855	0,741	0,723	0,832	0,753
Z_All	0,713	0,954	0,799	0,769	0,962	0,839
Z_c	0,770	0,847	0,791	0,864	0,724	0,784

Figure IV.16 confirms what we are saying, where it is clear that *Z_All* is the best feature among them with both classifiers.

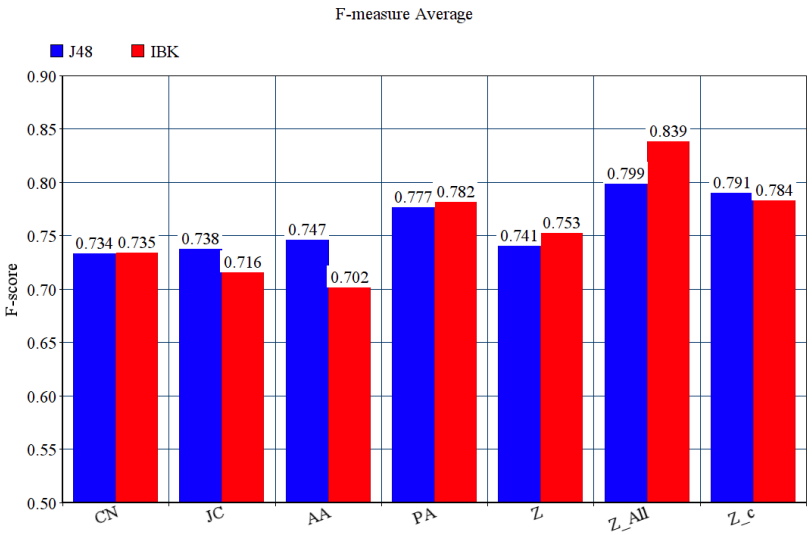


Figure IV.16: SLLP F-measure Average (HG and HY)

5.3 SLLP Vs ULLP Results

In this part, we have taken all obtained results for both datasets (HG and HY) and all techniques (DT KNN and Ranking). Table IV.10 shows that in the SLLP (both DT and KNN), *Z_All* is the best feature in all SLLP features, followed by *Z_c* which has also given high results. And in the ULLP, *Z_c* has achieved high results. We can also remark that *JC* is not suggested in the

ULLP approach due to its bad obtained results. Moreover, we can observe that SLLP in all its features are better than ULLP.

Table IV.10: SLLP vs. ULLP Performance Measures Average (HG and HY)

	Supervised L						**Unsupervised L**		
	J48 (Decision Tree)			**IBK (KNN)**			**Ranking**		
	Pr	**Rc**	**F-m**	**Pr**	**Rc**	**F-m**	**Pr**	**Rc**	**F-m**
CN	0,764	0,798	0,734	0,760	0,801	0,735	0,532	0,577	0,538
JC	0,698	0,806	0,738	0,688	0,790	0,716	0,167	0,222	0,186
AA	0,774	0,795	0,747	0,767	0,742	0,702	0,540	0,587	0,547
PA	0,740	0,889	0,777	0,739	0,887	0,782	0,591	0,640	0,596
Z	0,678	0,855	0,741	0,723	0,832	0,753	0,493	0,571	0,516
Z_c	0,770	0,847	0,791	0,864	0,724	0,784	**0,617**	**0,638**	**0,609**
Z_All	**0.713**	**0.945**	**0.799**	**0.769**	**0.962**	**0.839**			

Figure IV.17 summarizes all what we have discussed, where it is clear that *Z_All* is the best feature, and that SLLP is better than ULLP.

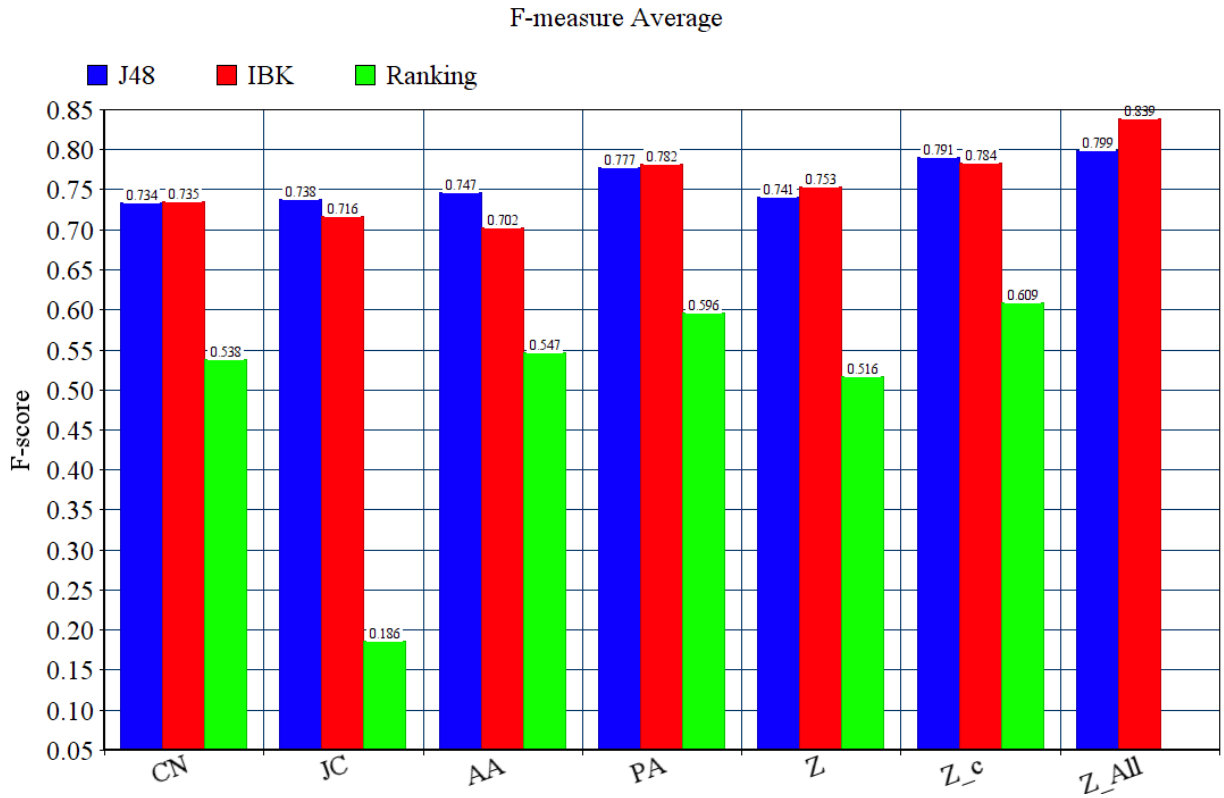


Figure IV.17: SLLP vs. ULLP F-measure Average (HG and HY)

Conclusion and Future Work

In this work, we have adressed complex networks and how they evolve in term of links appearing (edges dynamicity). Our main goal was to use a non-classic edges representation and to see if this will improve the link prediction. Our motivation is that most works in link prediction use features alone, and they did not investigate their combination.

We have performed the Link prediction task using two approaches, Unsupervised Learning Link Prediction (ULLP) and Supervised Learning Link Prediction (SLLP). In ULLP approach, we did the ranking strategy using four features from the node-based features each one alone. In addition, we investigated their combination (weighted sum). In SLLP approach, we characterized edges as elements to be classified to either link or non-link using simple node-based features with three scenarios (each one alone, then, combined and gathered as a vector of features).

In our experiments, we used two real network benchmarks. As a result, we found that using combined features enhance predictions rate in both approaches. Furthermore, we found that using gathered features in the SLLP approach gives the best results among all previous results with an average of about 84% (96% in the second dataset) which is a high value comparing to other results done in this scope [59]. Also, we confirmed that SLLP is better than ULLP.

More investigations - in the future - can be done in combining features, by choosing the best combination and coefficients and to solve the problem with a non-empirical method. Also, using other than neighbor-based features is another suggestion to be investigated like the use of content and node profile.

Appendices

A - Used Algorithms

1 Common Neighbors Algorithm:

Algorithm 1 Common Neighbors (CN)

1: **function** CN(A, $n1$, $n2$)
2: **Input:** $AdjacencyList : A$
$int : n1, n2$
3: **Output:** $int : cn$
4:
$$cn \leftarrow sharedNeighborsNumber(a, n1, n2)$$
5: Return cn ▷ Methods used will be explained in the auxiliary functions part
6: **end function**

2 Jaccard Coefficient Algorithm:

Algorithm 2 Jaccard Coefficient (JC)

1: **function** JC(A, $n1$, $n2$)
2: **Input:** $AdjacencyList : A$
$int : n1, n2$
3: **Output:** $float : jc$
4:
$$jc \leftarrow \frac{sharedNeighborsNumber(A, n1, n2)}{countNeighbors(A, n1) + countNeighbors(A, n2)}$$
5: Return jc ▷ Methods used will be explained in the auxiliary functions part
6: **end function**

3 Adamic-Adar Coefficient Algorithm:

Algorithm 3 Adamic-Adar Coefficient (AA)

function AA(A, $n1$, $n2$)
  Input: $AdjacencyList : A$
    $int : n1, n2$
  Output: $float : aa \leftarrow 0$
  $LinkedList < int >: L \leftarrow sharedNeighbors(A, n1, n2)$
  foreach $intedge$ in L **do**

$$aa \leftarrow aa + \frac{1}{(float)Math.Log(countNeighbors(A, edge))}$$

  end for
  Return aa ▷ Methods used will be explained in the auxiliary functions part
end function

4 Preferential Attachment Algorithm:

Algorithm 4 Preferential Attachment (PA)

function PA(A, $n1$, $n2$)
  Input: $AdjacencyList : A$
    $int : n1, n2$
  Output: $int : pa$

$$pa \leftarrow countNeighbors(A, n1) * countNeighbors(A, n2)$$

  Return pa ▷ Methods used will be explained in the auxiliary functions part
end function

4 Z construction Algorithm:

Algorithm 5 Combination of All Features (Z)

function Z(α, β, γ, δ, cn, jc, aa, pa)
    Input: $float : \alpha, \beta, \gamma, \delta, jc, aa$
    $int : cn, pa$
    Output: $float : z$

$$z \leftarrow (\alpha * cn + \beta * jc + \gamma * aa + \delta * pa)$$

    Return z ▷ a composite value done by the combination of all other metrics (that are weighted)
end function

5 Rescaling Algorithm:

Algorithm 6 Rescaling (Normalisation)

function RESCALING(A, $feature\_type$, $feature$, $out\, feature$‘)
    Input: $AdjacencyList : A$
    $int : feature\_type$
    $float : feature$
    Output: $float : feature$‘

$$feature‘ \leftarrow \frac{feature - Min(A, feature\_type)}{Max(A, feature\_type) - Min(A, feature\_type)}$$

    Return $feature$‘ ▷ this is done for normalization $(0- > 1)$ for every feature, where "feature_type" refers to the calculated feature
end function

6 Auxiliary Functions:

Algorithm 7 Auxiliary Functions

```
1: function SharedNeighbors(AdjacencyList a, int n1, int n2)
2:     Input: AdjacencyList : A
                int : n1, n2
3:     Output: LinkedList < int >: L
4:     bool[] : t1, t2                    ▷ both have A's number of vertices (A.getNumberOfVertices()) as a size and are initialized to false
5:     foreach Tuple < int > edge in A[n1] do
6:         t1[edge.Item1] ← true
7:     end for
8:     foreach each Tuple < int > edge in A[n2] do
9:         t2[edge.Item1] ← true
10:    end for
11:    for i ← 1 to A.getNumberOfVertices() do
12:        if (t1[i] ∧ t2[i]) then
13:            L.Add(i)    ▷ this will add the shared neighbor between n1 and n2 to the adjacency list if both t1[i] and t2[i] are true
14:        end if
15:    end for
16:    Return L
17: end function
18:
19: function sharedNeighborsNumber(AdjacencyList a, int n1, int n2)
20:    Input: AdjacencyList : A
                int : n1, n2
21:    Output: int : snn
22:    snn ← sharedNeighbors(A, n1, n2).Count()
23:    Return cn      ▷ it will return the number of common neighbors between n1 and n2 which is the number of elements in A
24: end function
25:
26: function countNeighbors(AdjacencyList A, int n)
27:    Input: AdjacencyList : A
                int : n
28:    Output: int : cnt
29:    cnt ← A[n].Count()
30:    Return cnt          ▷ it will return the number of neighbors that node n has
31: end function
```

B - The Flowchart of Our Work

- 1: reading the file that contains the network graph representation.
- 2-A: file type 1 (matrix representation), fill the adjacency list "adj1" with the source, destination and the class (link 1 or non-link 0) for the actual network "t", then read the "t+1" network graph and fill "adj2".
- 2-B: file type 2 (edge list representation). fill both "adj1" and "adj2" according to the timestamp in the read file.
- 3: fill the two "MyTableStructure" "mts1" and "mts2" with the source, destination, class, cn, jc, aa, pa and the composite metric "z" and "z_c".
- 4: choose between the supervised learning method or the unsupervised one.
- 5-A-1: supervised learning, setting the number of times that we change the training set, testing set and the tests' length.
- 5-A-2: supervised learning, saving the training data in an arff file extension for the machine learning software.
- 5-A-3: supervised learning, saving the testing data in an arff file extension for the machine learning software "weka".
- 5-A-4: supervised learning, proceeding both training and testing data to "weka".
- 5-B: unsupervised learning, setting the size "k" of the sets (that will be in a descending order).
- 6: if "reExperiment" equal true, repeat step "4", else, end of the work.

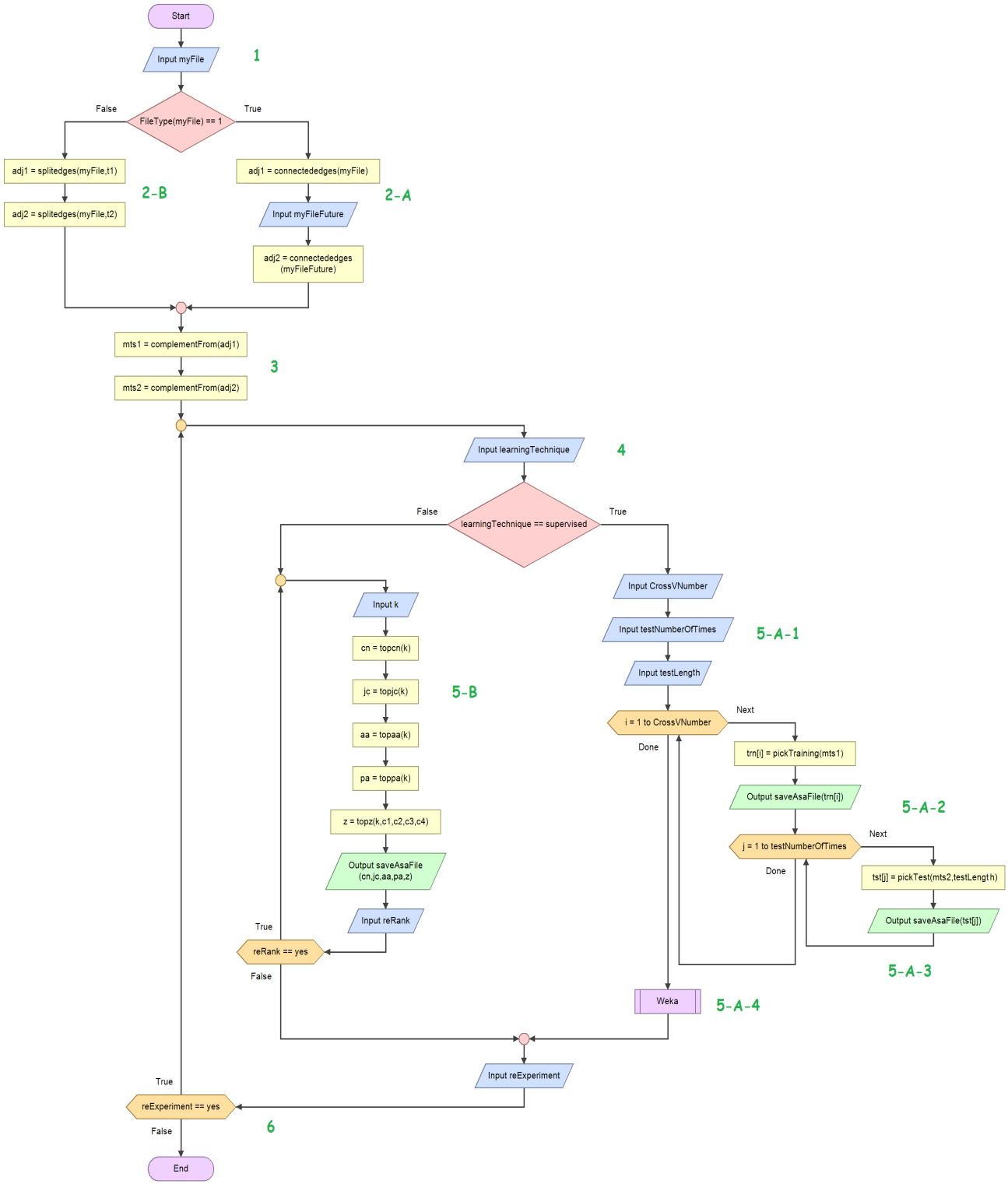


Figure 18: Flowchart Diagram of our [60]

C - Our Graphical User Interface (GUI)

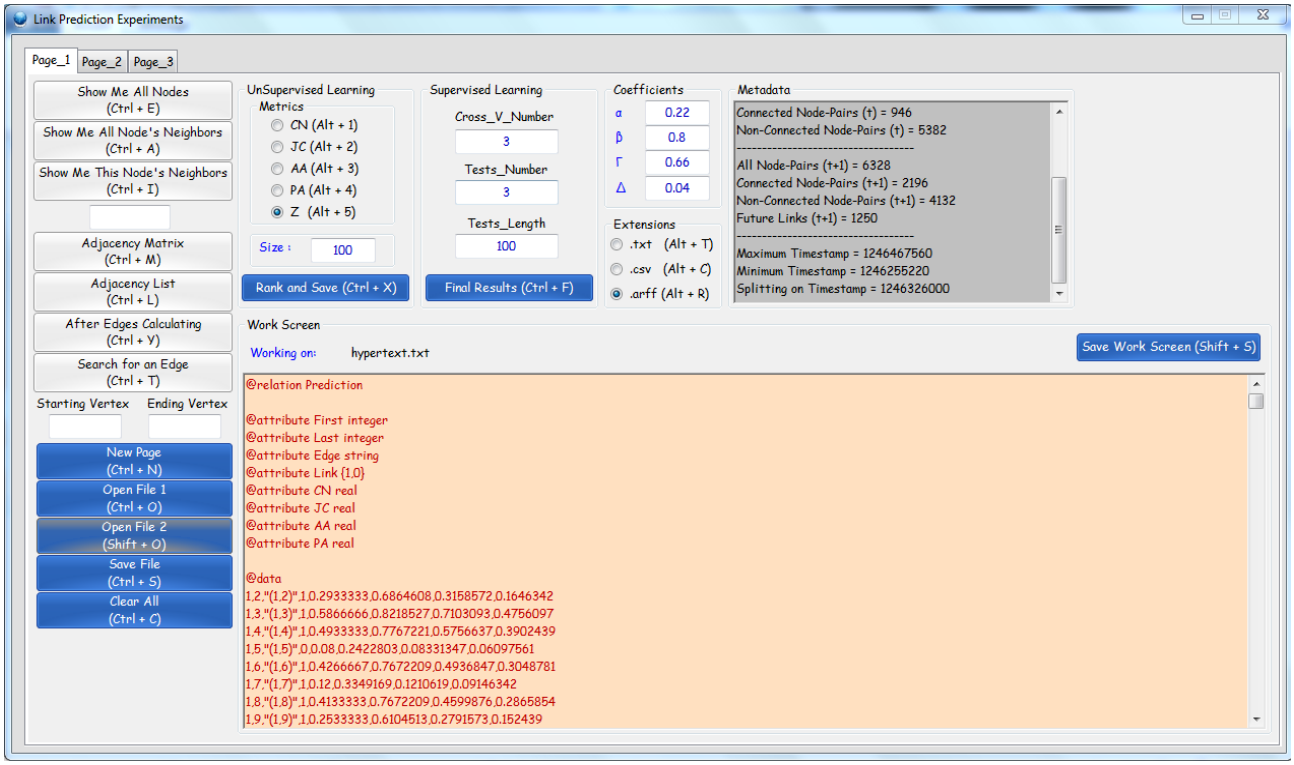


Figure 19: A snapshot of our Application (GUI)

- **Left Buttons:**
  - **Show me all nodes:** This will show all existing nodes (connected or non-connected ones).
  - **Show me all node's neighbors:** This will show all nodes with their neighbors (empty if no neighbor was found).
  - **Show me this node's neighbors:** This will show a specific node's all neighbors (empty if no neighbor was found).
  - **Adjacency matrix:** This will show the full Adjacency Matrix for the actual network (used for small networks).
  - **Adjacency list:** This will show the full adjacency list for the actual network (node1-node2-arc-class-cn-jc-aa-pa.)
  - **Search for an edge:** This will show a specific edge from the adjacency list defined by the first and the second node.
  - **New page:** To create a new TabPage in order to do another experiment (up to 10 pages per launch).
  - **Open file 1:** To open the dataset that contains our network representation(adjacency matrix and list types).
  - **Open file 2:** To open the futur dataset (if the first is an adjacency matrix type).
  - **Save file:** To save the actual and the future networks after the processing (calculating features).
  - **Clear all:** To clear all data done in the actual tabpage (including app's variables).
- **UnSupervised Learning:** With up to 5 choices, we can rank a "size" elements of the specific feature, and auto-save them in a folder followed by the chosen "size" and the feature used as a name.
- **Supervised Learning:** With the capability of choosing a "cross_v_number" number of times, and a "tests_number" number

of times, and a "tests_length" size, the process will generate many training sets ("cross_v_number" time) and in each one, "tests_number" files that contains the testing sets of a size equal to "tests_length", the final results will be saved in a folder that contains all the fills with their folder (all are named with the feature and the settings used).

- **Coefficients:** α , β , γ and Δ are the coefficient of CN, JC, AA and PA respectively to make the new feature "z_c" which is the combination of the previous features multiplied by their coefficients.
- **Extensions:** There is three deferent extensions used in this program, a simple one "txt" that contains the pair-node with its features separated with tab spaces, a "csv" one, which is similar to the previous one but separated with commas, an "arff" one, which is similar to the "csv" but it contains a specific header that "weka" can understand.
- **Dispalying screens:** There is two screens used in this program, one for metadata, which is the most important actions done (save, open, edges detected . . .), and a main one which contains the network's representation according to what has the user did, also we can save the observed screen.
- **Shortcuts ability and controls:** To improve the tests' speed, we added shortcuts everywhere, this will help the user to work faster, also, some controls are added to protect the incorrect data filled by the user.
- **Application's limits:** Due of time, and due to the experiments importance against the application's interface and additional functionalities, it is still incomplete (like showing matrix representation in big datasets).

D - Weka's Interface

Figure 20: Weka's Main Interface [61]

1: To start the work, we need to choose the "**Explorer**" option.

2: "**Open file**" will let us load one of the training set files generated by our c# application.

3: "**Choose**" will allow us to choose the classifier that will be used in the classification step.

4: "**Set. . .**" will let us load one of the testing set files generated by our c# application.

5: Here, we can alter the classifier's default settings, in our case, we modified the k values to 5 instead of 1 (of the k_{th} nearest neighbors).

6: "**More options. . .**" will give us more options, like outputting results (labels) or showing/hiding fields (like Confusion Matrix).

7: By clicking on "**Start**", weka will classify the testing set (the unseen data) according to the training set (the seen data) using the chosen classifier with the user's settings, then, it will show the generated results and some performance measures (Precision/Recall/F-measure).

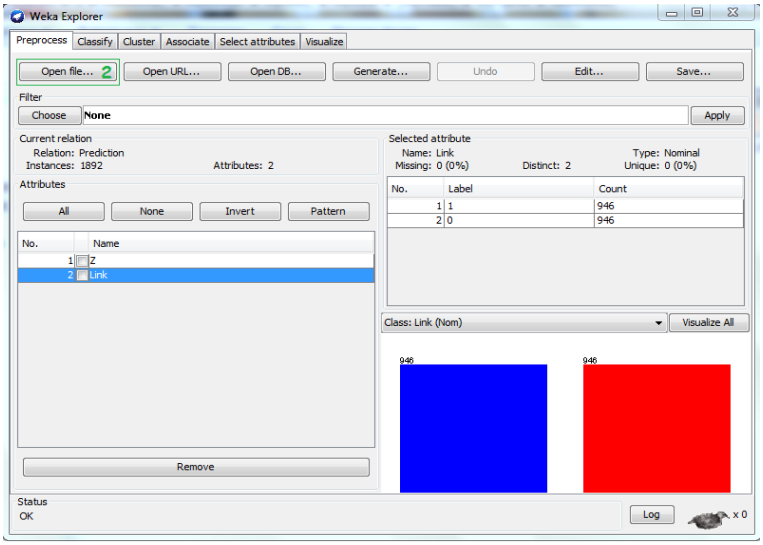


Figure 21: Loading a Training Set in Weka

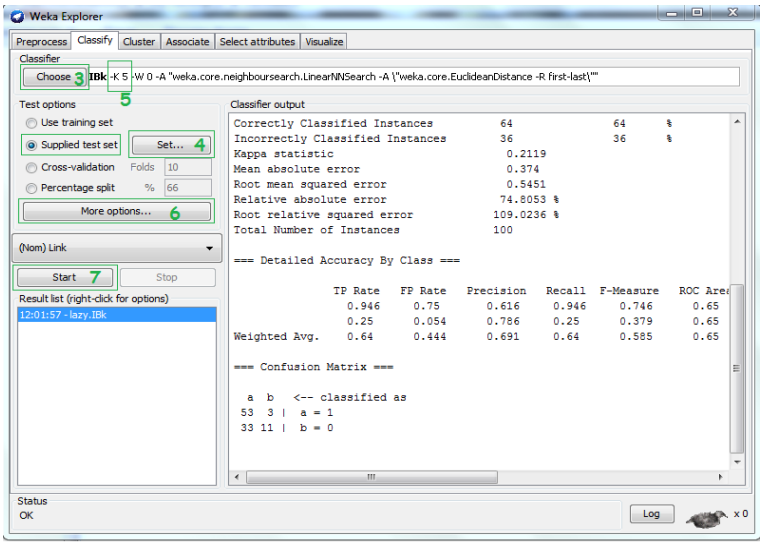


Figure 22: Loading and Applying Classification on a Test set

E - Full Tests

Table 11: SLLP All Performance Measures of Z_c (HG)

			J48 (Decision Tree)			IBK (KNN)		
	Cv	**Tn**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Z_c	**1**	**1**	0,754	1	0,860	0,776	0,918	0,841
		2	0,776	0,95	0,854	0,814	0,875	0,843
		3	0,833	0,98	0,901	0,887	0,922	0,904
	2	**1**	0,911	0,911	0,911	0,883	0,946	0,913
		2	0,852	0,902	0,876	0,846	0,863	0,854
		3	0,814	0,941	0,873	0,807	0,902	0,852
	3	**1**	0,5	1	0,667	0,786	0,957	0,863
		2	0,489	1	0,657	0,815	0,978	0,889
		3	0,5	0,957	0,657	0,786	0,957	0,863

Table 12: SLLP All Performance Measures of Z_c (HY)

			J48 (Decision Tree)			IBK (KNN)		
	Cv	**Tn**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Z_c	**1**	**1**	0,953	0,695	0,804	0,854	0,593	0,700
		2	0,936	0,83	0,880	0,731	0,717	0,724
		3	0,917	0,702	0,795	0,767	0,489	0,597
	2	**1**	0,83	0,765	0,796	0,806	0,569	0,667
		2	0,83	0,78	0,804	0,846	0,66	0,742
		3	0,898	0,8	0,846	0,921	0,636	0,752
	3	**1**	0,494	0,891	0,636	0,844	0,587	0,692
		2	0,453	0,739	0,562	0,8	0,609	0,692
		3	0,5	0,851	0,630	0,771	0,574	0,658

Table 13: ULLP All Performance Measures (HG)

	k	Precision	Recall	F-measure
CN	**30**	0,47	0,30	0,37
	50	0,50	0,54	0,52
	70	0,57	0,87	0,69
JC	**30**	0	0	0
	50	0	0	0
	70	0	0	0
AA	**30**	0,43	0,28	0,34
	50	0,52	0,56	0,54
	70	0,57	0,86	0,69
PA	**30**	0,87	0,59	0,70
	50	0,62	0,70	0,66
	70	0,56	0,89	0,68
Z	**30**	0,27	0,19	0,22
	50	0,50	0,58	0,53
	70	0,53	0,86	0,65
Z_c	**30**	0,83	0,54	0,66
	50	0,60	0,65	0,63
	70	0,54	0,83	0,66

Table 14: ULLP All Performance Measures (HY)

	k	Precision	Recall	F-measure
CN	**30**	0,57	0,36	0,44
	50	0,56	0,6	0,58
	70	0,53	0,79	0,63
JC	**30**	0,27	0,21	0,23
	50	0,38	0,49	0,43
	70	0,36	0,64	0,46
AA	**30**	0,63	0,41	0,5
	50	0,58	0,63	0,60
	70	0,50	0,76	0,60
PA	**30**	0,57	0,39	0,46
	50	0,48	0,55	0,51
	70	0,46	0,73	0,56
Z	**30**	0,60	0,40	0,48
	50	0,58	0,64	0,61
	70	0,49	0,76	0,59
Z_c	**30**	0,67	0,43	0,53
	50	0,56	0,61	0,58
	70	0,5	0,76	0,60

Table 15: SLLP All Performance Measures (HG)

			J48 (Decision Tree)			IBK (KNN)		
	Cv	**Tn**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
CN	**1**	**1**	0,944	0,607	0,739	0,951	0,696	0,804
		2	0,893	0,532	0,667	0,839	0,553	0,667
		3	0,957	0,449	0,611	0,958	0,469	0,630
	2	**1**	1	0,667	0,800	1	0,667	0,800
		2	0,944	0,694	0,800	0,944	0,694	0,800
		3	0,971	0,688	0,805	0,97	0,667	0,790
	3	**1**	0,886	0,633	0,738	0,882	0,612	0,723
		2	1	0,667	0,800	1	0,667	0,800
		3	1	0,615	0,762	1	0,577	0,732
JC	**1**	**1**	0,721	0,875	0,791	0,683	0,732	0,707
		2	0,729	0,745	0,737	0,723	0,723	0,723
		3	0,765	0,796	0,780	0,791	0,694	0,739
	2	**1**	0,805	0,733	0,767	0,806	0,644	0,716
		2	0,761	0,714	0,737	0,786	0,673	0,725
		3	0,861	0,646	0,738	0,889	0,667	0,762
	3	**1**	0,778	0,714	0,745	0,78	0,653	0,711
		2	0,842	0,711	0,771	0,833	0,556	0,667
		3	0,837	0,692	0,758	0,829	0,654	0,731
AA	**1**	**1**	0,949	0,661	0,779	0,921	0,625	0,745
		2	0,844	0,574	0,683	0,88	0,468	0,611
		3	0,96	0,49	0,649	0,964	0,551	0,701
	2	**1**	1	0,6	0,750	1	0,533	0,695
		2	0,943	0,673	0,785	0,879	0,592	0,708
		3	1	0,604	0,753	1	0,583	0,737
	3	**1**	0,895	0,694	0,782	0,931	0,551	0,692
		2	0,97	0,711	0,821	1	0,467	0,637
		3	0,949	0,712	0,814	1	0,481	0,650
PA	**1**	**1**	0,845	0,875	0,860	0,855	0,839	0,847
		2	0,813	0,83	0,821	0,886	0,83	0,857
		3	0,927	0,776	0,845	0,921	0,714	0,804
	2	**1**	0,972	0,778	0,864	0,973	0,8	0,878
		2	0,907	0,796	0,848	0,889	0,816	0,851
		3	1	0,75	0,857	0,929	0,813	0,867
	3	**1**	0,921	0,714	0,804	0,881	0,755	0,813
		2	1	0,733	0,846	0,919	0,756	0,830
		3	0,952	0,769	0,851	0,953	0,788	0,863
Z	**1**	**1**	0,875	0,75	0,808	0,857	0,75	0,800
		2	0,81	0,723	0,764	0,833	0,745	0,787
		3	0,921	0,714	0,804	0,825	0,673	0,741
	2	**1**	0,842	0,711	0,771	0,839	0,578	0,684
		2	0,87	0,816	0,842	0,902	0,755	0,822
		3	0,932	0,854	0,891	0,975	0,813	0,887
	3	**1**	0,5	0,816	0,620	0,814	0,714	0,761
		2	0,45	0,8	0,576	0,8	0,622	0,700
		3	0,519	0,788	0,626	0,783	0,692	0,735
Z_All	**1**	**1**	0,965	0,982	0,973	1	0,982	0,991
		2	0,957	0,957	0,957	0,979	0,968	0,973
		3	0,92	0,939	0,929	0,977	0,878	0,925
	2	**1**	1	0,822	0,902	0,975	0,867	0,918
		2	1	0,816	0,899	0,979	0,969	0,974
		3	0,979	0,958	0,968	0,958	0,958	0,958
	3	**1**	0,597	0,939	0,730	0,959	0,959	0,959
		2	0,563	1	0,720	0,978	1	0,989
		3	0,633	0,962	0,764	0,912	1	0,954

Table 16: SLLP All Performance Measures (HY)

			J48 (Decision Tree)			IBK (KNN)		
	Cv	**Tn**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
CN	**1**	**1**	0,621	0,964	0,755	0,621	0,964	0,755
		2	0,56	1	0,718	0,534	1	0,696
		3	0,55	0,936	0,693	0,536	0,957	0,687
	2	**1**	0,542	0,963	0,694	0,542	0,963	0,694
		2	0,596	1	0,747	0,596	1	0,747
		3	0,495	0,958	0,653	0,495	0,958	0,653
	3	**1**	0,645	0,984	0,779	0,652	0,984	0,784
		2	0,589	1	0,741	0,609	1	0,757
		3	0,557	1	0,715	0,558	0,981	0,711
JC	**1**	**1**	0,635	0,839	0,723	0,617	0,893	0,730
		2	0,614	0,915	0,735	0,548	0,979	0,703
		3	0,576	0,809	0,673	0,553	0,894	0,683
	2	**1**	0,627	0,87	0,729	0,641	0,926	0,758
		2	0,608	0,906	0,728	0,576	0,925	0,710
		3	0,569	0,854	0,683	0,538	0,875	0,666
	3	**1**	0,643	0,885	0,745	0,635	0,885	0,739
		2	0,602	0,893	0,719	0,578	0,929	0,713
		3	0,598	0,907	0,721	0,575	0,926	0,709
AA	**1**	**1**	0,675	0,964	0,794	0,659	0,964	0,783
		2	0,595	0,936	0,728	0,524	0,936	0,672
		3	0,581	0,915	0,711	0,512	0,894	0,651
	2	**1**	0,628	0,907	0,742	0,581	0,926	0,714
		2	0,617	0,943	0,746	0,6	0,962	0,739
		3	0,536	0,938	0,682	0,511	0,938	0,662
	3	**1**	0,645	0,984	0,779	0,648	0,967	0,776
		2	0,589	1	0,741	0,609	0,946	0,741
		3	0,557	1	0,715	0,584	0,963	0,727
PA	**1**	**1**	0,609	1	0,757	0,611	0,982	0,753
		2	0,48	1	0,649	0,5	0,979	0,662
		3	0,485	1	0,653	0,516	1	0,681
	2	**1**	0,545	1	0,706	0,571	0,963	0,717
		2	0,576	1	0,731	0,582	1	0,736
		3	0,485	0,979	0,649	0,479	0,958	0,639
	3	**1**	0,649	1	0,787	0,67	1	0,802
		2	0,596	1	0,747	0,602	1	0,752
		3	0,557	1	0,715	0,57	0,981	0,721
Z	**1**	**1**	0,667	0,893	0,764	0,647	0,982	0,780
		2	0,623	0,915	0,741	0,541	0,979	0,697
		3	0,594	0,872	0,707	0,592	0,957	0,731
	2	**1**	0,645	0,907	0,754	0,6	0,944	0,734
		2	0,617	0,943	0,746	0,58	0,962	0,724
		3	0,536	0,938	0,682	0,549	0,938	0,693
	3	**1**	0,652	0,984	0,784	0,663	0,967	0,787
		2	0,602	1	0,752	0,616	0,946	0,746
		3	0,553	0,963	0,703	0,598	0,963	0,738
Z_All	**1**	**1**	0,679	0,982	0,803	0,632	0,982	0,769
		2	0,557	0,936	0,698	0,549	0,957	0,698
		3	0,55	0,936	0,693	0,495	0,957	0,653
	2	**1**	0,545	1	0,706	0,543	0,944	0,689
		2	0,576	1	0,731	0,602	1	0,752
		3	0,485	0,979	0,649	0,495	0,938	0,648
	3	**1**	0,663	1	0,797	0,656	1	0,792
		2	0,598	0,982	0,743	0,587	0,964	0,730
		3	0,564	0,981	0,716	0,568	1	0,724

References

[1] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesus Gómez-Gardeñes, Miguel Romance, Irene Sendiña-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014. 1, 4

[2] Cuneyt Gurcan Akcora, Barbara Carminati, and Elena Ferrari. User similarities on social networks. *Social Network Analysis and Mining*, 3(3):475–495, 2013. 1, 20

[3] 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. 1, 18, 19, 22, 23, 24, 33, 34

[4] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002. 4

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

[6] Complex Networks and Graphs. https://www.teachengineering.org/view_lesson.php?url=collection/jhu_/lessons/jhu_cnetworks_lesson01.xml. Accessed: 2016-03-03. 5

[7] What is Epidemiology? http://www.bmj.com/about-bmj/resources-readers/publications/epidemiology-uninitiated/1-what-epidemiology. Accessed: 2016-03-03. 5

[8] Health topics, Epidemiology. www.who.int/topics/epidemiology/en/. Accessed: 2016-03-03. 5

[9] An Epidemiology Certificate Program. http://asegrad.tufts.edu/academics/explore-graduate-programs/epidemiology. Accessed: 2016-03-03. xi, 5

[10] Christian Bachmaier, Ulrik Brandes, and Falk Schreiber. Biological networks. 2010. 6

[11] The Structure of Scientific Collaboration Networks. http://www.pnas.org/content/98/2/404.full. Accessed: 2016-03-03. 6

[12] Collaboration Network on the Field of Network Science. http://users.dimi.uniud.it/~massimo.franceschet/bibliometrics/. Accessed: 2016-03-03. xi, 6

[13] Norman P Hummon and Patrick Dereian. Connectivity in a citation network: The development of DNA theory. *Social networks*, 11(1):39–63, 1989. 7

[14] Network of Academic Citations. http://blogs.lse.ac.uk/impactofsocialsciences/the-handbook/chapter-3-key-measures-of-academic-influence/. Accessed: 2016-03-03. xi, 7

[15] Social Networking. http://whatis.techtarget.com/definition/social-networking. Accessed: 2016-03-03. 7

[16] Social Networks. http://www.journals.elsevier.com/social-networks. Accessed: 2016-03-03. 7

[17] Visual Representation of the Social Network Theory. http://socialnetworking.lovetoknow.com/What_is_Social_Network_Theory. Accessed: 2016-03-03. xi, 8

[18] Definition of Online Social Networking. http://socialnetworking.lovetoknow.com/about-social-networking/definition-online-social-networking. Accessed: 2016-03-03. 8

[19] Top 15 Most Popular Websites. http://www.ebizmba.com/articles/social-networking-websites. Accessed: 2016-03-04. 8, 10

[20] How to Write a Bio for a Social Media Profile. http://www.blog.dedicaces.us/?p=223. Accessed: 2016-03-03. xi, 9

[21] Facebook Definition. http://whatis.techtarget.com/definition/Facebook. Accessed: 2016-03-04. 9

[22] Twitter Definition. http://whatis.techtarget.com/definition/Twitter. Accessed: 2016-03-04. 9

[23] LinkedIn Definition. http://whatis.techtarget.com/definition/LinkedIn. Accessed: 2016-03-04. 10

[24] Pinterest Definition. http://whatis.techtarget.com/definition/Pinterest. Accessed: 2016-03-04. 10

[25] Google+ Definition. http://whatis.techtarget.com/definition/Google-plus. Accessed: 2016-03-04. 10

[26] Remco Van Der Hofstad. Random graphs and complex networks. *Available on http://www. win. tue. nl/rhofstad/NotesRGCN. pdf*, page 11, 2009. 11

[27] Scale-Free Network. http://mathworld.wolfram.com/Scale-FreeNetwork.html. Accessed: 2016-03-04. 11

[28] Real Networks and Small Worlds. https://sciencetonnante.wordpress.com/2013/02/25/propagation-depidemies-et-graphes-aleatoires/. Accessed: 2016-03-03. xi, 11

[29] Small World Networks. http://mathinsight.org/small_world_network. Accessed: 2016-03-04. 12

[30] Small World Graph. http://pimediaonline.co.uk/science-tech/six-degrees-of-separation-2/. Accessed: 2016-03-03. xi, 12

[31] Simulating Retrieval From a Highly Clustered Network implications for spoken Word Recognition. http://journal.frontiersin.org/article/10.3389/fpsyg.2011.00369/full. Accessed: 2016-03-03. 12

[32] Community Structure in Social and Biological Networks. http://www.pnas.org/content/99/12/7821.full. Accessed: 2016-03-04. 12

[33] Example Network Showing Community Structure. http://www.nature.com/nphys/journal/v8/n1/fig_tab/nphys2162_F1.html. Accessed: 2016-03-03. xi, 13

[34] James Peter Bagrow. *Analysis and applications of complex networks*. 2008. 13

[35] David Liben-Nowell. *An algorithmic approach to social networks*. PhD thesis, Massachusetts Institute of Technology, 2005. 14

[36] Link Prediction Algorithms. http://be.amazd.com/link-prediction/, . Accessed: 2016-03-04. xi, 14

[37] Degree Centrality. http://faculty.ucr.edu/~hanneman/nettext/C10_Centrality.html#Degree. Accessed: 2016-03-05. 15

[38] Link Prediction Algorithms. http://be.amazd.com/link-prediction/, . Accessed: 2016-03-06. 19, 22, 23, 33, 34

[39] Prantik Bhattacharyya, Ankush Garg, and Shyhtsun Felix Wu. Analysis of user keyword similarity in online social networks. *Social network analysis and mining*, 1 (3):143–158, 2011. 20

[40] Jorge Valverde-Rebaza and Alneu de Andrade Lopes. Exploiting behaviors of communities of twitter users for link prediction. *Social Network Analysis and Mining*, 3(4):1063–1074, 2013. 25

[41] Haifeng Liu, Zheng Hu, Hamed Haddadi, and Hui Tian. Hidden link prediction based on node centrality and weak ties. *EPL (Europhysics Letters)*, 101(1):18004, 2013. 25

[42] Jérôme Kunegis, Ernesto W De Luca, and Sahin Albayrak. The link prediction problem in bipartite networks. In *Computational intelligence for knowledge-based systems design*, pages 380–389. Springer, 2010. 26

[43] Unsupervised Learning. https://www.coursera.org/learn/machine-learning/lecture/olRZo/unsupervised-learning. Accessed: 2016-03-30. 27

[44] Machine Learning: Supervised Learning. https://www.udacity.com/course/viewer#!/c-ud675/l-684818868/m-640579190. Accessed: 2016-04-12. 27

[45] Link Prediction Using Supervised Learning. http://www.powershow.com/view1/d2293-ZDc1Z/Link_Prediction_Using_Supervised_Learning_powerpoint_ppt_presentation. Accessed: 2016-04-12. 27

[46] Data Mining: What is Data Mining? http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm. Accessed: 2016-04-12. 29

[47] An Introduction to Data Mining. http://www.thearling.com/text/dmwhite/dmwhite.htm. Accessed: 2016-04-12. 29, 37

[48] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012. 31

[49] Emöke-Ágnes Horvát. Modelling and inferring connections in complex networks. 2013. 32

[50] Decision Tree - Classification. http://www.saedsayad.com/decision_tree.htm. Accessed: 2016-04-12. xi, 36, 37

[51] K Nearest Neighbors - Classification. http://www.saedsayad.com/k_nearest_neighbors.htm. Accessed: 2016-04-12. 37

[52] Classification of Hand-written Digits (3). http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/. Accessed: 2016-04-12. xi, 38

[53] C# Station. http://www.csharp-station.com/. Accessed: 2016-04-12. 40

[54] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/. Accessed: 2016-05-04. 40

[55] Classification Accuracy is Not Enough: More Performance Measures You Can Use. http://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/. Accessed: 2016-04-12. 40, 42

[56] Jérôme Kunegis. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion*, pages 1343–1350, 2013. URL http://userpages.uni-koblenz.de/~kunegis/paper/kunegis-koblenz-network-collection.pdf. 43

[57] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. on Mobile Computing*, 6(6):606–620, 2007. 44

[58] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *J. of Theoretical Biology*, 271(1):166–180, 2011. 44

[59] Hially Rodrigues De Sá and Ricardo BC Prudêncio. Supervised link prediction in weighted networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2281–2288. IEEE, 2011. 53, 54, 58

[60] Flowgorithm. http://www.flowgorithm.org/. Accessed: 2016-04-29. xii, 65

[61] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009. xii, 69