PhD–FSTM–2022–012
The Faculty of Sciences, Technology and Medicine

DISSERTATION

Defence held on 14/01/2022 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG EN INFORMATIQUE

by

Saharnaz Esmaeilzadeh Dilmaghani

Born on 09 September 1992 in Salmas (Iran)

On Trustworthy AI and Localized Complex Network Analytics

Dissertation defense committee

Dr. Matthias Brust, dissertation supervisor
Research Scientist, Université du Luxembourg

Dr. Grégoire Danoy
Research Scientist, Université du Luxembourg

Dr. Pascal Bouvry, Chairman
Professor, Université du Luxembourg

Dr. Jean-Philippe Humbert
Vice-President of the Agence pour la Normalisation et l'Economie de la Connaissance (ANEC GIE) and Deputy Director of the Institut Luxembourgeois de la Normalisation, de l'Accréditation, de la Sécurité et qualité des produits et services (ILNAS)

Dr. Carlos H.C. Ribeiro, Vice Chairman
Professor, Instituto Tecnológico de Aeronáutica

"You cannot discover new oceans unless you have the courage to lose sight of the shore."

– André Gide
Author

To my husband, and my family.

Abstract

We live in a world where the interaction of many different entities results in the formation of complex systems. The communication between billions of smart devices, interactions of millions of people in social networks, and the existence of our biological life, which is based on seamless interactions between hundreds of genes and proteins within our cells, all are just a few examples of the complex systems surrounding us. At the core of these complex systems, there is clear evidence of a *complex network*, which symbolizes the interaction between the system's components. Analytical metrics and algorithms derived from graph theory are used in network analysis to understand the functionality of complex systems, anticipate system behavior, and control changes. Many of these global patterns in complex networks are generally influenced by decisions made by communities.

Communities are a tightly connected group of nodes with sparse connections to the rest of the network. These modular structures are crucial to understanding the complex network due to being closely tied to the system's functional and topological features. They can, for example, represent modules of proteins with similar functionality in a protein interaction network or influence dynamic network activities such as opinion and epidemic propagation. Local community detection methods have gained popularity among other strategies to discover communities in a complex network. The traditional methods are based on a top-down approach acquiring global information about the entire network; however, due to the growing size and complexity of existing networks, they often result in tangled communities, hence not providing functional information of the network.

The primary goal of this thesis is to provide methods and solutions for local network analysis. The following components comprise the thesis contribution:

1) Introduce a transformation approach to construct networks from relational data and describe how network structure affects community detection,

2) Provide a comprehensive evaluation of current local community detection techniques and suggest a locality exploration scheme (LES) for community detection algorithms,

3) Develop a local community detection Algorithm (LCDA) and employ it on real-world data,

4) Extend LCDA to LCDA-GO, which integrates biological functional information and detects protein communities in the cell on the PPI network.

Thereby, this thesis proposes a novel community detection algorithm that addresses the shortcomings of prior algorithms by presenting a local method. The applicability of the suggested algorithm is investigated by running it on real-world PPI networks.

Furthermore, this thesis contributes to industrial technical reports and whitepapers on the standardization and regulation of big data and Artificial Intelligence (AI). The thesis addresses the critical issues of digital trust in big data and AI by incorporating technical standardization and cutting-edge research solutions. The key contributions include, but are not limited to:

1) A comprehensive review of the state-of-the-art in numerous scientific and standard materials regarding privacy and trustworthiness concerns, including the introduction of privacy leaks and mitigation measures in big data and AI,

2) Investigating the societal implications of artificial intelligence standards in light of the recently initiated worldwide and European standardized processes,

3) Design and implementation of a scheme that connects scientific contributions and standardization efforts in the direction of AI conformity assessment.

The contribution of the thesis to standards demonstrates an impact on both scientific and standardization communities by contributing to both and offering recent outcomes from each.

Acknowledgements

My Ph.D. life at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, has been a wonderful experience equipped me with new skills both in my professional and personal life. I would like to express my sincere gratitude to many people, without whom I would not have completed this Ph.D. Thesis.

First, I would like to thank my supervisors Prof. Pascal Bouvry and Dr. Matthias Brust for giving me the opportunity as a Ph.D. student and for their continuous support and guidance during the past four years. I also had the privilege of working with the Institut Luxembourgeois de la Normalisation, de l'Accréditation, de la Sécurité et qualité des produits et services (ILNAS) with Dr. Jean-Philippe Humbert, Mr. Nicolas Domenjoud, and Mrs. Natalia Cassagnes as part of our collaboration in standardization. Their advice and support helped me to learn and gain knowledge in standardization and to successfully apply standardization in my Ph.D. thesis. I would also like to thank my CET committee member Prof. Martin Theobald and Dr. Grégoire Danoy for their valuable feedback and suggestions. During the last year of my Ph.D., I had the pleasure to working with Prof. Carlos H. C. Ribeiro from the Instituto Tecnológico de Aeronáutica, Brazil, whose valuable and constructive suggestions facilitated the completion of this thesis. I appreciate him for kindly accepting to participate in my dissertation defense committee.

I would like to extend my special gratitude to my collaborators at SnT, Dr. Grégoire Danoy and Dr. Emmanuel Kieffer for their constructive feedback and comments on this thesis.

I also wish to thank all my colleagues at SnT for providing such an amazing work environment and for their support throughout my Ph.D. Many thanks to Dr. Abdallah Ibrahim, Dr. Nader S. Labib, Mrs. Panissara Thanapol, Mr. Pierre-Yves Houitte, Mr. Gabriel Duflo, Mr. Clement Parisot, Dr. Rafael Bleuse for their support.

I am deeply grateful and indebted to my family, who set me off on the road to this Ph.D. a long time ago and never wavered in their support and understanding. And my special thanks to my husband, Dr. Alireza Haqiqatnejad who is a true friend, closest advisor, dedicated partner, and love of my life. I greatly appreciate his support, patience, and grateful insight in every step of my Ph.D. journey.

Finally, I would like to gratefully acknowledge the funding of my Ph.D. by the joint research programme University of Luxembourg/SnT-ILNAS on Digital Trust for Smart-ICT. The partial support from PCOG is also gratefully acknowledged.

List of Abbreviations

AI	Artificial Intelligence
CEN	European Committee for Electrotechnical Standardization
CENELEC	EuropeanEuropean Committee for Electrotechnical Standardization
ETSI	European Telecommunication Standards Institute
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineering
ILNAS	Institut Luxembourgeois de la Normalisation, de l'Accréditation, de la Sécurité et qualité des produits et services
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union (ITU) on Telecommunication Standardization Sector
JTC	Joint Technical Committee
ML	Machine Learning
PET	Privacy-Enhancing Techniques
SDO	Standards Developing Organization
SC	Subcommittee
TC	Technical Committee
TR	Technical Report
WG	Working Group
NLP	Natural Language Processing

List of Tables

List of Figures

Contents

Preface

This Ph.D. Thesis has been carried out from March, 2018 to November, 2021, at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, under the supervision of Prof. Pascal Bouvry and Dr. Mathias R. Brust. The time-to-time evaluations of the Ph.D. Thesis were duly performed by the CET members Prof. Pascal Bouvry, Dr. Mathias R. Brust, Dr. Grégoire Danoy, and Prof. Martin Theobald. Additionally, the European Committee for Electrotechnical Standardization reviewed the thesis and recognized it for its significant contribution to standardization through research and innovation. The thesis was rewarded in the category of Young Researchers at the Standards+Innovation Awards 2021.

Support of the Thesis

This Ph.D. Thesis has been fully supported by the joint research programme University of Luxembourg/SnT-ILNAS on Digital Trust for Smart-ICT. The partial support from PCOG is also gratefully acknowledged.

Publications

The original publications that have been produced during the period of Ph.D. candidacy is listed below. These publications are referred to in the text by J ≡ Journal, B ≡ Book chapter, C ≡ Conference, and T ≡ Technical report.

Journal Papers

[J1] **Saharnaz Dilmaghani**, Matthias R. Brust, Apivadee Piyatumrong, Grégoire Danoy, and Pascal Bouvry, "Link Definition ameliorating Community Detection in Collaboration Networks," Frontiers in Big Data 2, vol. 2, pp.22, Jun. 2019.

[J2] **Saharnaz Dilmaghani**, Matthias R. Brust, Carlos H.C. Ribeiro, Emmanuel Kieffer, Grégoire Danoy, and Pascal Bouvry, "From Communities to Protein Complexes: A Local Community Detection Algorithm on PPI Networks," Plos One, vol. 17, Jan. 2022.

[J3] Guillaume Avrin, Patrick Bezombes, Antonio Chella, Renaud Di Francesco, **Saharnaz Dilmaghani**, Sebastian Hallensleben, Thomas Hildebrandt, Valerie Livina, Aditya Mohan, and Meri Valtiala, "Standardisation of Artificial Intelligence: Making a "New

World" Brave, with Support of Human Requirements in a Machine Intelligence Environment," Springer, [Submitted].

Book Chapters

[B1] **Saharnaz Dilmaghani**, Apivadee Piyatumrong, Grégoire Danoy, Pascal Bouvry, and Matthias R. Brust, "Innovation Networks from Inter-organizational Research Collaborations," Heuristics for Optimization and Learning, Springer, Cham, pp. 361-375, Dec. 2020.

Conference Papers

[C1] **Saharnaz Dilmaghani**, Apivadee Piyatumrong, Grégoire Danoy, Pascal Bouvry, and Matthias R. Brust, "Transforming Collaboration Data into Network Layers for Enhanced Analytics," International Conference on Optimization and Learning, arXiv:1902.09364, Feb. 2019.

[C2] **Saharnaz Dilmaghani**, Matthias R Brust, Grégoire Danoy, Natalia Cassagnes, Johnatan Pecero, and Pascal Bouvry, "Privacy and Security of Big Data in AI Systems: A Research and Standards Perspective," 2019 IEEE International Conference on Big Data (Big Data), IEEE, pp. 5737-5743, Dec. 2019.

[C3] **Saharnaz Dilmaghani**, Matthias R Brust, Grégoire Danoy, and Pascal Bouvry, "Local Community Detection Algorithm with Self-defining Source Nodes," Complex Networks 2020, Complex Networks & Their Applications IX, Springer, Cham, vol. 943, pp. 200-210, Dec. 2020.

[C4] **Saharnaz Dilmaghani**, Matthias R Brust, Grégoire Danoy, and Pascal Bouvry, "Community Detection in Complex Networks: A Survey on Local Approaches" 13th Asian Conference on Intelligent Information and Database Systems (ACIIDS), Springer, Cham, vol. 12672, pp. 757-767, Apr. 2021. **[Nominated for the best Paper Award]**

Technical Reports

[T1] **Saharnaz Dilmaghani**, Nader Samir Labib, Chao Liu, Matthias R Brust, Grégoire Danoy, Pascal Bouvry, "White Paper: Data Protection and Privacy in Smart ICT-Scientific Research and Technical Standardization," ILNAS, Oct. 2018.

[T2] **Saharnaz Dilmaghani**, Matthias R Brust, Grégoire Danoy, Pascal Bouvry, "Artificial Intelligence and Big Data: Gap Analysis Between Scientific Research and Technical Standardization," ILNAS, Oct. 2019.

Workshops

[W1] **Saharnaz Dilmaghani**, Matthias R Brust, Grégoire Danoy, Pascal Bouvry, "Network-based Machine Learning for Privacy Preservation in Bio-medical Big Data Analytics," NetSci18, Oct. 2018.

[W2] **Saharnaz Dilmaghani**, Matthias R Brust, Grégoire Danoy, Pascal Bouvry, "Locality in Community Detection Algorithms: Development of Local Approach and Analyzing the Existing Approaches," NetSci21, Oct. 2021.

Publications not Included in the Thesis

[J3] Guillaume Avrin, Patrick Bezombes, Antonio Chella, Renaud Di Francesco, **Saharnaz Dilmaghani**, Sebastian Hallensleben, Thomas Hildebrandt, Valerie Livina, Aditya Mohan, and Meri Valtiala, "Standardisation of Artificial Intelligence: Making a "New World" Brave, with Support of Human Requirements in a Machine Intelligence Environment," Springer, vol. XX, pp. XX, XXX 2021.

[C1] **Saharnaz Dilmaghani**, Apivadee Piyatumrong, Grégoire Danoy, Pascal Bouvry, and Matthias R. Brust, "Transforming Collaboration Data into Network Layers for Enhanced Analytics," International Conference on Optimization and Learning, arXiv:1902.09364, Feb. 2019.

[T1] **Saharnaz Dilmaghani**, Nader Samir Labib, Chao Liu, Matthias R Brust, Grégoire Danoy, Pascal Bouvry, "White Paper: Data Protection and Privacy in Smart ICT-Scientific Research and Technical Standardization," ILNAS, Oct. 2018.

Chapter 1

Introduction

This chapter introduces the problem of interest in this thesis. The primary focus of the thesis is artificial intelligence (AI) and data analytics developments. This thesis consolidates data science developments by introducing a community detection algorithm and addresses the data protection, and trustworthiness of AI on the other hand. The background, motivation and contributions, and organization of the thesis are presented in the subsequent sections.

1.1 Background

The rapid growth of data has triggered various fields to exploit analytical techniques on large, diverse datasets that includes structural and non-structural data generated by different sources. The importance of big data does not revolve solely on the volume, variety and velocity [2], but mainly on what we can do with it. Data analytics, Machine Learning (ML) and Artificial Intelligence (AI) combine different technologies to extract information from raw data with the goals of increasing the efficiency and the accuracy of prediction and decision making and also minimizing the risks and computational costs [3].

The developments of big data and AI thus could be paved from two perspectives: Facilitating the development of data analytics techniques by proposing advanced solutions on the omnipresent data, and taking actions towards developing rules and standards to control and harness AI and data analytics technologies against the yet-to-be-known risks.

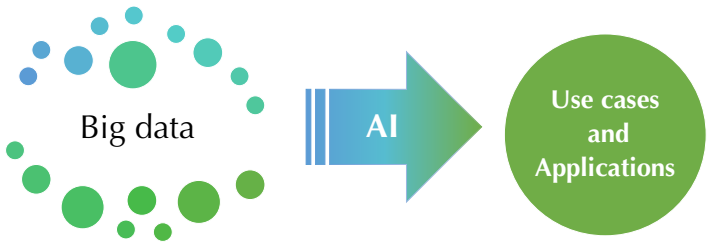

FIGURE 1.1: Utilizing AI on various big data for data analytics, decision making and prediction in different applications.

Complex Networks and Graphs

Many systems that we are surrounded with in the 21st century are part of complex systems that in fact are difficult to understand and infer knowledge. The underlying structure of most of the complex systems is a complex network that represents the connection of system's components [4]:

- The cellular network of body that integrates the interactions between genes, proteins, and metabolites into live cells.
- The connections between neurons, neuron network, derives functionality of the brain.
- The exchange of goods, services, and products are maintained through trade network.
- The social, family, and friend relationships are integrated in social networks representing societies' opinions.
- The connection and interaction of different devices through the internet connection in communication networks shows the communication system
- The transmission lines between generators are described in power grid networks.

Thus, any interaction and connection between components of a complex system can be encoded in a network which then represents a particular functionality of that complex system.

A network is an extract from *graph* introduced in graph theory. A network can be defined as a graph consisting of a set of nodes (e.g., vertices) and links (e.g., edges). The study of networks lies within graph theory and they have emerged in several disciplines. Networks are attached to real-world complex systems, thus, regarding the application and the nature of the nodes the represent different networks. While they can have the same graph representation if the same edges are connected to the same vertices. This way networks and graphs are distinguishable, however, they have been used interchangeably in the literature. Figure 1.2 shows graph representation of different networks that are extracted from real-world examples.

Networks are defined as $G = (V, E)$ with V including the nodes of the network and E, the list of links between each nodes of the network. Each network may represent different characteristics that often depend on the network attributes such as network density, degree centrality. Some of the attributes provide general information of the network by considering it as one global structure. Other attributes, however, provide specific local perspectives by spotlighting the nodes and links of a network. The use of these attributes play an important role to first understand the network, and next, to gain a deeper knowledge on the performance, structure, and connectivity of the network.

Community detection is part of the network analytics which provides information on specific structures of the network known as community structure. A *community* is defined as locally dense connected subgraphs in a network [4]. In a nutshell, nodes within a community are densely intra-connected and loosely inter-connected. Many real-world networks contain community structures. The social communities of social networks represents communities of people who are interested in similar beliefs and opinions. Protein-protein interaction network includes protein communities such that each community consists of set of proteins caring a particular biological functionality of cells. Communities of different networks may represent different insights and knowledge, however, they are all representing community structure properties.

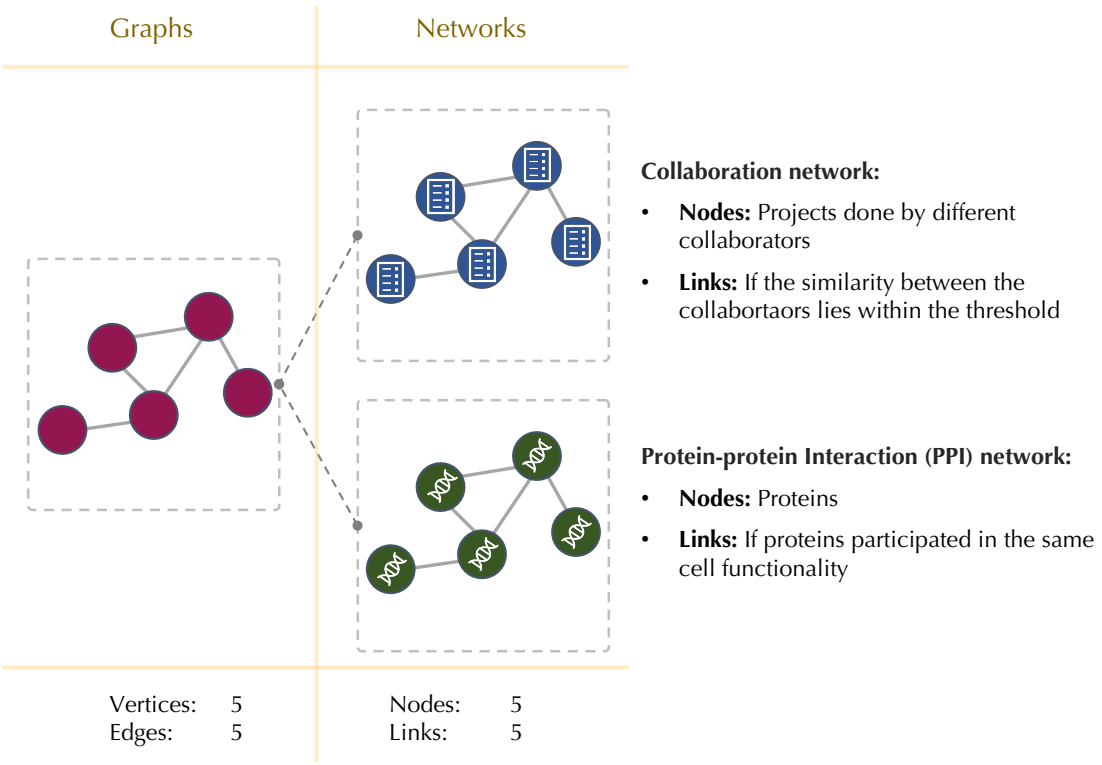

FIGURE 1.2: Graph and network representations.

The study of communities are important in several aspects. They often correspond to functional units of the network and hence represent a functional map of the network. Exploiting communities in large networks provides a meta-node information of the network that is easier to study rather than studding each individual node. Insights on the correlation between the topology of the network and its functionality is also often carried out by communities. Moreover, communities assist the understanding of different processes in the network. For instance, rumor spreading and epidemic spreading are such processes that are better understood by detecting communities in the network and analysing the impact of communities in such processes.

Data Privacy and AI Trustworthiness

The integration of AI in various domains [5] significantly increases concerns regarding the privacy and security of data. The data that actuates AI includes various sensitive information, particularly individuals' information, including: images, speech, comments and posts on social media [6,7], financial transactions, and health record information. Feeding such data in AI systems, they become vulnerable to privacy and security attacks that are even significantly increased recently [8,9].

Standards and regulations are playing an important role to reduce the risks raised by AI. In this thesis we also covered the technical standardization overview to provide additional insights on the conducted studies. Standards Developing Organization (SDO) SDOs develop technical standards and guidelines to address the needs and demands of particular adopters. Standards act an important role in achieving interoperability and portability of complex ICT technologies and platforms.

International Organization for Standardization (ISO) is one of the well-known SDOs in the world. Together with International Electrotechnical Commission (IEC), they initiated a Joint Technical Committee (JTC) as ISO/IEC JTC 1 to cover smart ICT related domains. AI, big data, privacy and data protection are among topics that are included in this TC. In the European level, European Committee for Standardization (CEN) and European Committee for Electrotechnical Standardization (CENELEC) are involved in the development of "Focus Group Artificial Intelligence" to provide standardization road-map for AI according to the European requirements [10]. Moreover, European Telecommunications Standards Institute (ETSI) has also initiated projects focused on the use cases, applications and security challenges of AI. In this thesis, we consider on the resources produced by ISO sub committees of SC 27 – "Information Security, Cybersecurity and Privacy Protection", and SC 42 – "Artificial Intelligence" and CEN and CENELEC AI working group, CEN/CLC/JTC 21 on Artificial Intelligence.

In this thesis, we consider two directions: First, we propose an algorithm that locally detects communities in big data and show its application on biological networks. Next, we investigate on a global picture by describing digital trust in big data and AI from standards and scientific research perspectives.

1.2 Motivation

The main motivations behind the work carried out in this thesis are explained in two main parts, first, the research contribution in complex networks and community detection, and then, the standardization collaboration in data protection and AI trustworthiness. Each part, addresses different aspects and answers research questions.

Part I: Complex Networks and Community Detection

- Investigating on the network construction form relational data that are not presented as networks, is there a unique solution and how would be the network representation different from each other
- Impact of the network construction on the network analysis algorithms particularly on community detection algorithms
- Investigated on community detection algorithms and dig into the taxonomies regarding locality of the algorithm? what is locality and what are the different levels of locality in community detection?
- Develop a local community detection algorithm and compare it with the existing algorithms
- The advantage of local algorithms over global algorithms? What are the applications and use cases?
- Developing a local community detection algorithm and its application on biological networks

Part II: Data Protection and AI Trustworthiness

- The impact of standardization in AI and big data

- Survey on the state-of-the-art scientific and standardization paper focusing on data protection and AI trustworthiness
- Developing a standard framework to implement scientific measures and investigate on the level of trustworthiness of different AI systems

1.3 Thesis Outline and Contributions

The contributions of this thesis can be categorized into two main parts, which are organized into 8 chapters. Briefly speaking, in this thesis, we address two different challenges in data analytics and AI. First, we focus on complex networks challenges and develop algorithms to detect communities in networks. In the next part, we consider the global picture of AI and big data and discuss data protection, and AI trustworthiness.

The organization of the thesis is structured as follows:

Part I: Complex Networks and Community Detection

Chapter 2: This chapter includes the discussion over the advantages of network analytics over relational data analytics. A data-to-network transformation approach is proposed using a real-world research collaboration data.

Chapter 3: The impact of network topology constructed from relational data is investigated in this chapter. We perform network analytics on the network layers constructed from research collaboration data to search for correlations between the level of data the needs to be involved in generating the networks. We, particularly, studied the the performance of a community detection algorithm on the generated networks.

Chapter 4: Focusing on community detection in complex networks, we notice a gap with respect to the taxonomy of community detection algorithms. This chapter, describes the difference between community detection algorithms in the literature from a different perspective. The *global* and *local* community detection algorithms are defined and discussed in this chapter.

Chapter 5: In this chapter, we propose a local community detection algorithm (LCDA) that relies on the information gathered from local neighbours of each node. Our algorithm follows a set of principles and modifies the nodes within communities repeatedly. We implement and test the algorithm on both synthetic and real-world networks.

Chapter 6: In this chapter, we apply our proposed algorithm on a biological, protein-protein interaction (PPI), network. We also define a new version of the previous LCDA algorithm as LCDA-GO that includes biological functional information in the process of identifying communities in the network.

Part II: Data Protection and AI Trustworthiness

Chapter 7: The vulnerability of big data and AI is discussed from two perspectives in this chapter, research and standardization. We first surveyed an described the existing attacks on big data and AI. Then, we investigated on the mitigation strategies proposed from both

standardization and research references.

Chapter 8: This chapter focuses on the necessity of AI certification and assessments. It provides a practice on analyzing the requirements for developing an AI assessment and gathers information on the existing knowledge gaps. We develop a questionnaire base on the key features of AI assessment and asked people with different backgrounds in AI and its application to fill it. We analyze and report the results in this chapter.

Finally, Chapter 9 concludes the thesis and provides insights on the future work.

Part I

Complex Networks and Community Detection

Chapter 2

Transforming into Networks

Exploring and analyzing data through the lens of networks has gained significant momentum in various domains from biology and neuroscience [11] (e.g., brain networks [12]) to modeling geographical systems [13], analyzing galaxy distributions [14], and quantifying reputation in art [15]. The popularity of using networks are due to the measurable properties networks possess from graph theory to reveal the complex characteristics of the data. While networks provide various advantages to analyze data the transformation from data to network is not a trivial task since for each dataset there are multiple ways to construct a network [16]. Scientific collaboration data is a type of datasets that represent a network structure due to the connectivity and the relationship between different co-authors that are collaborating on various tasks. The study of collaboration data as networks permits to extract knowledge on the structure and patterns of communities. Previous studies model the connectivity of collaboration data considering it as a binomial problem with respect to the existence of a collaboration between individuals [17]. However, such a data consists of a high diversity of features that describe the quality of the interaction such as the contribution amount of each individual.

In this chapter, we confronted with the challenge of network generation by proposing an atomized *data-to-network* approach for inter-organizational research collaboration data. The results of this contribution is published in a Springer chapter entitled as "**Innovation Networks from Inter-organizational Research Collaborations**" [18].

2.1 Introduction

Network structures are known to be essential in data analytics due to their potential to describe the underlying relationship between entities. They also offer various ways to represent the correlations within dataset that is easier compared to the traditional relational databases. Similar to graph theory, the study of networks (network science) relies on variety of properties and techniques that are applied in many applications such as visualization, link prediction, and clustering. One of the main concerns to study a relational data as a network is to construct network from the dataset. Thus, the network construction from a dataset plays an important role in data analytics. The step has been considered in various areas from biology and neuroscience [11] (e.g., brain networks [12]) to modeling and analyzing galaxy distributions [14], and quantifying reputation in art [15].

Transforming data to network often provides computationally efficient algorithms with lower complexity in comparison to a relational data tables [19]. Moreover, networks represent several properties that describes the data from different perspectives and different granularity levels. Numerous algorithms can be applied directly on networks such as the *Louvain*'s community detection algorithm [20] and *Page Rank*, that identifies the most influential object within a network [18]. Furthermore, data transformed into network layers can provide evidence for missing or omitted information [21, 22] as well as predicting the growth of the network in terms of nodes and links [23, 24].

With these advantages of networks at hand, we are confronted with the challenge of how to transform relational data into appropriate networks [18]. The challenge is twofold: It is not only on how to represent elements of a network but also the specific construction principles, since, for each dataset, there are numerous ways how to transform data into network [16]. Each network reveals a particular perspective on the input dataset - emphasizing some characteristics while diminishing the dominance of others [18, 25].

The state-of-the-art approaches are addressed the challenge of converting relational data to network from different perspectives and most of the time based on the application domain and type of the data. In a protocol-based approach, introduced by Karduni et al. [13], authors developed a set of rules to tackle this challenge in spatial data of geographical systems. They defined a protocol consisting of set of rules to extract the underlying network from the spatial data. In a different approach, authors formulated the problem as a *link prediction* problem to predict the links between entities of a relational data and constructed a network out of relational data. Casiraghi et al. [26] developed a generalized hypergeometric ensembles algorithm to predict and infer the connections within the objects in relational data. The study represents the perspective of link prediction while applying predictive analysis. Likewise, Xiang et al. [27] established a link-based latent variable model to infer the friendship relations within a social interaction. In [28] authors exploited a model based on the tensor factorization technique to exclude links from a dataset which consists different countries' news [18]. Moreover, Akbas et al. [29] proposed a model to construct a social network. Their proposed method is built on interactions between individuals, for instance phone calls. They defined a weight to each type of interaction exist in the dataset, then, authors measure a value as the combination of various interaction types and used it to construct a network. However, they did not have a relational data to convert into network. Akbas et al. followed up their study on network generation from interaction patterns by studying the social networks of animal groups [30–32]. In a similar approach Hong et al. [14] develop a linkage model and convert the cosmic web into three network models. The purpose of the study was to benefit from network representations to investigate the architecture of the universe.

In scientific collaboration data, the initiatives are taken by Newman [17] studies. He established networks from scientific paper publication datasets data and convert the authors' collaborations as links in the network. Thus, nodes act as the authors and two nodes are connected if two authors had a collaboration in a scientific paper. The network modeling introduced by Newman follows a binary approach when deciding on the links of the network. Scientific collaboration networks are also studied as *hypergraphs* by Ouvrard et. al. [33]. The authors emphasized on enhancing the visualization of these networks with respect to network properties.

Our approach is slightly different than the state-of-the-art such that it atomizes the process of network generation from the collaboration data. We also define a linkage measure to utilize all data features defining that specific relationship between the nodes.

2.2 Data to Network with Linkage Threshold

We propose a data-to-network approach to construct networks from an inter-organizational research collaboration dataset. The dataset contains information corresponding to various outcomes that are delivered by contribution of researchers within various teams. However, the data also includes features that represents the quality of contribution of researcher from some aspects. For instance, the level of contribution that each researcher is dedicated to the the delivery is one feature that is explained in the dataset. Thus, ignoring such features may impact the representation of data as a network. Exploiting such features, we define a *Linkage Threshold* measure to calculate the contribution strength between collaborators [18].

For this context we define network G as $G = (V, E)$ such that V is the set of nodes, and E is the list of tuples representing the link between each pair of nodes. Each node in the network represents research teams and links illustrate the common contribution of teams together. Each team (i.e., node) consists of a number of collaborators working within the team to deliver an outcome. Each collaborator is contributed in the team with a contribution level that may be different that his/her teammate. An illustration of the network structure is shown in Fig. 2.1. As shown in this figure, two teams are connected if they have common collaborators. Hence, if there is no collaboration between two teams no link will be presented between the corresponding nodes.

To formulate the definition of link in the network, we leverage the participation level represented as a percentage in the dataset. We define LT considering two features which describe collaboration 1) the number of common collaborators within each pair of teams, and 2) the contribution percentage of the common collaborators. Assume $Team_i$ and $Team_j$ have n common collaborators such that each collaborator contributes up to a certain level within the team. We define p_i^m as the contribution percentage of collaborator m in $Team_i$. We determine $\mathcal{M} = \{Team_i \cap Team_j\}$ such that it identifies the list of common collaborators between $Team_i$ and $Team_j$. We thus formalize LT between each $Team_i$ and $Team_j$ as [18]

$$LT = \frac{1}{2n} \sum_{\forall m \in \mathcal{M}} p_i^m + p_j^m, \tag{2.1}$$

where, LT is the *Linkage Threshold* which measures the contribution strength between two teams. The value range of LT starts from 0% to 100%. Each value within this range represents the average of contribution percentage of collaborators within a pair of teams. For instance, with LT equals to 20%, two teams in the network are connected if the average contribution

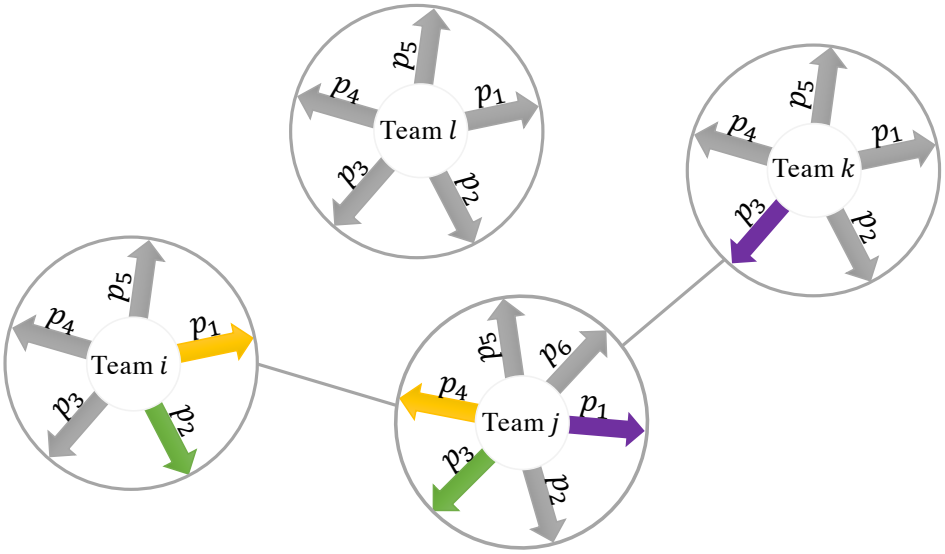

FIGURE 2.1: The network structure is identified by nodes (gray circles) which illustrate teams and contain collaborators. The links are describing the collaboration between the nodes. Each pair of teams may have common collaborators (illustrated as non-gray arrows) and other members who has not collaborated (grey arrows) with other teams. In this example, *Team i* and *Team j* have two common collaborators with different contribution levels (p_i), and *Team j* and *Team k* have only one collaborator in common. However, there is not a common collaborator between *Team l* and other teams.

percentage of the common members between those projects is equal or greater than 20%. Thus, those two teams are neighbors in the network. In other words, a particular LT identifies teams who have the highest contribution of collaborators. In a nutshell, increasing LT results in a sparse network since the teams are obliged to contain a high level of collaboration with other teams, and lowering the LT range will increase the density of the network since naturally people are collaborated within many teams but with lower contribution level in each. In a few cases the defined threshold could not be met by the nodes, hence, there will be no links between that node and the rest of the nodes and it remains as isolated node in the network.

Description of the algorithm

Let's assume $\mathcal{D}$ as the relational collaboration dataset. We need to identify the *nodesList* and *linksList* to construct network $G = (nodeList, linkList)$. We first extract the research teams from $\mathcal{D}$ and store it as node's list (see line 2). For each tuple of the teams, LT is calculated as described in Equation (2.1). Given the value of LT (see line 3), those links that satisfy the condition specified in the algorithm (see line 6) are appended to the *linkList* (see line 7). In this stage, we have extracted both lists of nodes and links for the network. Finally, considering the different given values for LT, a set of networks are constructed and stored in a vector $\mathcal{G}$ (see line 9) [18].

Complexity analysis

For constructing network G with n nodes and m links, our proposed algorithm, Algorithm 3.1 operates as follow. The complexity analysis depends mainly on two stages: 1) Comparing each pair of nodes to find those that serves the identified condition for threshold. This is the

most expensive part in case of computational costs with $O(n^2)$. 2) The complexity of the network generation is linear such that for n nodes and m links the complexity is $O(n+m)$.

Algorithm 2.1 Data-to-Network Layers

Input: $\mathcal{D}$, a dataset of research collaboration.
Output: $\mathcal{G}$, a vector of generated network layers.

1: **procedure** DATA-TO-NETWORKS($\mathcal{D}$)
2: $nodesList \leftarrow$ teams within $\mathcal{D}$
3: **for** each $threshold$ in $range(0, 100)$ **do**
4: **for** each tuple of $team$ in $nodesList$ **do**
5: $LT \leftarrow LinkageThreshold(team_i, team_j)$
6: **if** $LT \geq threshold$ **then**
7: $linksList \leftarrow (team_i, team_j)$
8: $Network\ G \leftarrow GenerateNetwork(nodesList, linksList)$
9: Insert G to $\mathcal{G}$
10: return $\mathcal{G}$

2.3 Experiment Setup

2.3.1 Dataset

We use collaboration data of researchers derived from the *National Electronics and Computer Technology Center* (NECTEC) from Thailand. The organization consists of different R&D departments and researchers from electronics and computer science topics (e.g., AI and advanced electronic sensing, intelligent systems and networks). NECTEC is organized such that experts collaborating within teams which may comprise various deliverables: *intellectual property (IP)*, *papers*, or *prototypes*. The information of collaborations and contributors has stored in a relational database consisting of collaborations conducted between July 2013 and July 2018 [18].

The dataset has been retrieved from NECTEC's knowledge management system with two key information: 1) the type of the deliverable, and 2) team contributors and contributions. The dataset of combined deliverables consists of almost $8,000$ records for more than $3,000$ teams. Table 3.1 represents details regarding the statistics of the dataset [18].

TABLE 2.1: Detail of NECTEC collaboration dataset.

Deliverable Type	# Researchers	# Teams
Paper	576	1717
Prototype	524	539
IP	489	630

Overall, the dataset includes more than $1,000$ researchers contributing within different teams to deliver various outcomes. NECTEC evaluates each collaborator with two parameters defined internally: 1) contribution percentage, and 2) IC-score. The contribution percentage describes a member's contribution to the particular team of each member and varies from 0% to 100%. The IC-score is a measure defined based on the type of delivery. For instance,

TABLE 2.2: Statistics from NECTEC dataset, mean (μ), standard deviation (σ) and variance (σ^2) of IC-score and contribution percentage is calculated.

Features	μ	σ	σ^2
IC-score	3.16	4.24	1.79
Contribution percentage	23.30	22.80	5.20

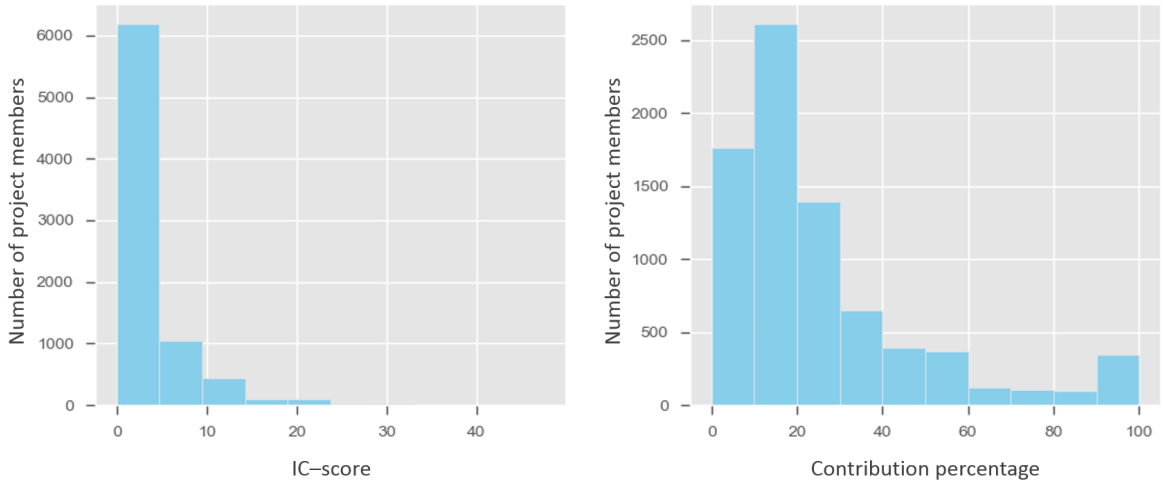

FIGURE 2.2: Histograms of IC-score and contribution percentage in the collaboration dataset. They represent the number of members within a certain value of IC-score (or contribution percentage) in the dataset.

an industrial-level prototype delivery of a team has a higher IC-score than a lab-level one. To obtain the IC-score for a member, the total IC-score value of the deliverable has divided to the contribution percentage of the member [18].

We perform data analytics on NECTEC dataset to acquire information regarding the features describing the collaboration. We first explore the data by examining the statistical details related to the features. The results are shown in Table 3.1. As shown in the table, IC-score represents a small value in average that is equal to 3.16. Considering the definition behind this value, IC-score relies highly on the type of deliverable (e.g., prototype) and the contribution of members [18]. The contribution percentage of members, however, is distributed with the mean value of 23.30.

We plot the distribution of the IC-score and contribution percentage (cf. Fig. 2.2) in a histogram. Both histograms illustrate that a large number of members participated in teams with lower IC-score and contribution percentage [18].

2.3.2 Network Metrics

Each network, $G = (V, E)$, could be described by its properties representing its nodes and links arrangement in the network. A set of measures are extracted from graph theory and used in networks science to analyze the corresponding network. We discriminate the measures in two categories of *global* and *local* based on the level of information they capture for the computations. The *global* measures require a wider structure information and explore the

whole network, instead, the *local* metrics solely focus on the information obtained from an individual node or its neighbours. Each category of metrics provides a different perspective from the network and thus valuable to understand the network [18].

We choose a set of metrics both from local and global measures to analyzed the obtained networks from the proposed *Data-to-Network* algorithm in Section 3.3. We choose the network density, centrality measures, and connected components to analyze the generated networks both in *global* and *local* levels. The metrics are defined as the following.

Definition 1. *Network density. The network density d of a network G is measured as*

$$d = \frac{2m}{n(n-1)},$$

where, n is the number of nodes and m is the number of links in network G.

Network density describes the ratio of the existing links to the potential ones in the network (i.e., the number of links in a complete graph with the same number of nodes). It varies form 0, if there is no link between the nodes, to 1, if all possible links exist in the network. Hence, a network density close to 0 indicates a sparse network while a higher density describes a dense network. Network density depends on the information form the whole network structure, hence, it is a global metric.

Definition 2. *Degree Centrality. Degree centrality is calculated for each node as:*

$$C_D(v) = deg(v),$$

where, deg(v) is the number of direct links connected to v from its neighbours in the network.

The degree centrality is the first centrality measure that explains the importance of a node in the network. It simply calculates as the number of links connected to that node. Node degree also implies the number v's neighbours, and thus, the metric is local.

Definition 3. *Closeness Centrality. The closeness centrality [34] of node v in network G is calculated as*

$$C_C(v) = \sum_{\forall u \in V} \frac{1}{d(v,u)},$$

where, $d(v,u)$ is the distance between to pairs of nodes, v and u. Hence, $C_C(v)$ is calculated as the average shortest distance length from v to every other node in the network.

Closeness centrality also captures the importance of individual nodes in the network. It explains how close a node is to all other nodes. Hence, the more central a node is, the closer it is to all other nodes. Closeness centrality is measure for individual nodes in the network, nevertheless, it requires information from the whole network. Thus, it is considered as a global metric.

Definition 4. *Betweenness Centrality. The betweenness centrality [35] of node v in G is measured as follows*

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}},$$

where, σ_{st} total number of shortest paths length from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.

Betweenness centrality indicates the importance of a node by describing how many times a node acts as a bridge along the shortest path between two other nodes. Nodes with high betweenness may have a significant influence in a network due to their control over the flow of information passing between others through them. Similar to closeness centrality describes an individual node's influence in the network, however, it requires global network information. Therefore, it is a global measure.

Definition 5. *Clustering Coefficient. The clustering coefficient [36] of node v in network G is formulated as*

$$CC(v) = \frac{2T(v)}{deg(v)(deg(v) - 1)}$$

where, $T(v)$ identifies the number of triangles through node v and $deg(v)$ is the degree of v.

The clustering coefficient represents the degree to which nodes are strongly grouped together. The value of the clustering coefficient lies between 0 for a *star* network - in which a node's neighbors are not connected to each other, and 1 for a *clique* network - in which every two distinct nodes are adjacent. Clustering coefficient is a local measure since it relies on the local information of a node only.

Definition 6. *Connected Components. The connected components of G are calculated as the number of sub-networks including at least two directly connected nodes connected.*

In a connected component, two nodes are in the same sub-network if there is a path between them in the network. The identification of connected components relies on the global information of the network, hence, it is a global measure.

Exploiting the above-mentioned set of standard metrics, we are able to describe and analyze the networks we constructed with different *LinkageThreshold*s with the *Data-to-Network* approach.

2.4 Network Analysis

We employ our *Data-to-Network*, on the NECTEC dataset to transform the collaboration dataset to a set of network layers. As described in Section 3.3, our methodology provides a team-based perspective where certain members are collaborating within. Hence, the generated networks illustrate teams that are connected if they satisfy the defined threshold of LT. As a final result, we obtained a vector of network layers, each represents a certain level of connection defined by LT. For each network we calculated the set of standard network metrics as already described in Section 2.3.

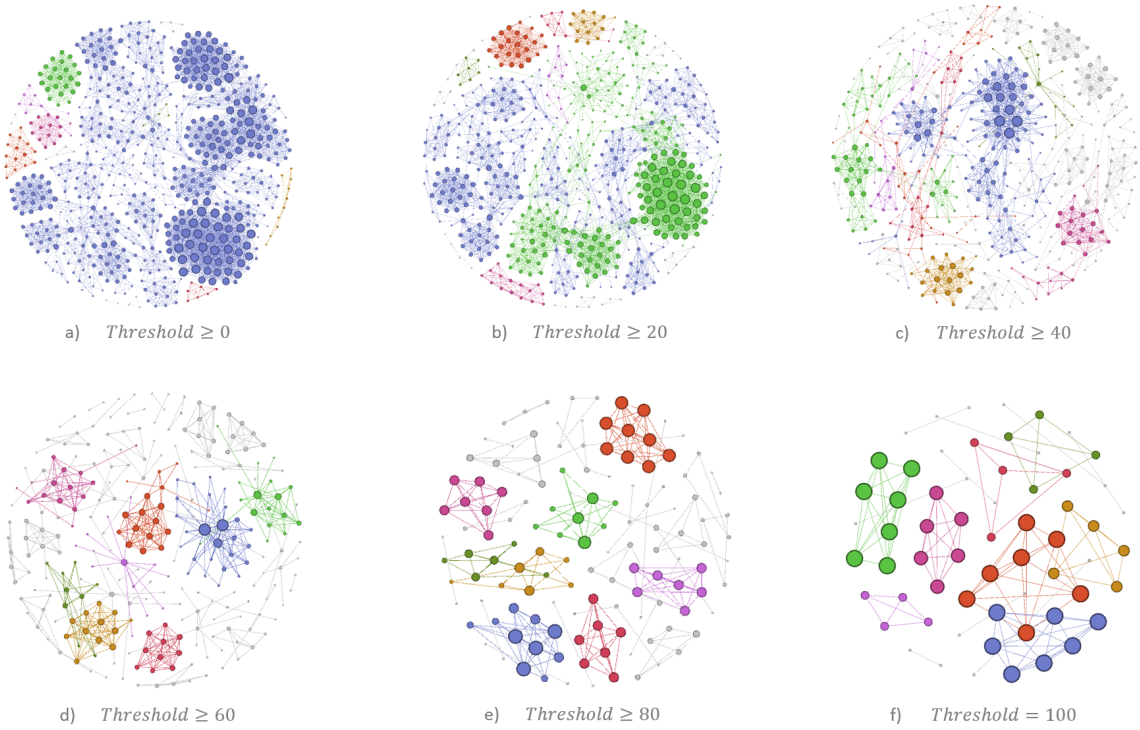

FIGURE 2.3: Visualization of the generated networks. In each network, the size of the nodes represents the degree of a node and the color illustrates the components. Such that blue shows components with the highest number of nodes, whereas gray represents the smallest components of a network. Moreover, green and red describe components which have a number of nodes within the range of previous cases. The networks are visualized using Gephi [1] and the Fruchterman-Reingold layout.

We perform *Data-to-Network* approach on the NECTEC teams by defining 11 thresholds for LT, starting from 0 to 100 percentage. We obtain one network for each LT. Fig.2.3 illustrates the final networks for some of LT thresholds. The networks are colored to show the connected components. As we expected, increasing the threshold level highly impacts the number of isolated nodes in the networks since the number of nodes in networks that could fit in the high restricted conditions ($LT > 80$) was less than other cases with more relaxed conditions. Therefore, we eliminate the isolated nodes during the experiments to analyze the networks regardless of the influence of these nodes.

The LT controls the level of contribution and provides insights on different level of collaborations. The first network is generated by $LT = 0$ where the connections are created if there is any collaboration between teams. In case of $LT = 100$, teams are only connected if they have a full collaborations (i.e., contribution percentage). Therefore, this layer of network only represents strong collaboration in which the members are fully participated in projects [18].

We first analyze the generated networks using the global metrics which provide a general perspective about the networks. We, then, analyze networks by exploiting local measures for a detailed perspective. The global metrics are explained in Table 2.3.

Table 2.3 provides general network information with respect to the number of nodes and links and density of each networks. As shown in the table, the number of teams who can satisfy the certain LT is decreased. In addition, the number of connected components has first increased such that in $LT = 50$ the generated network is the most fragmented layer with

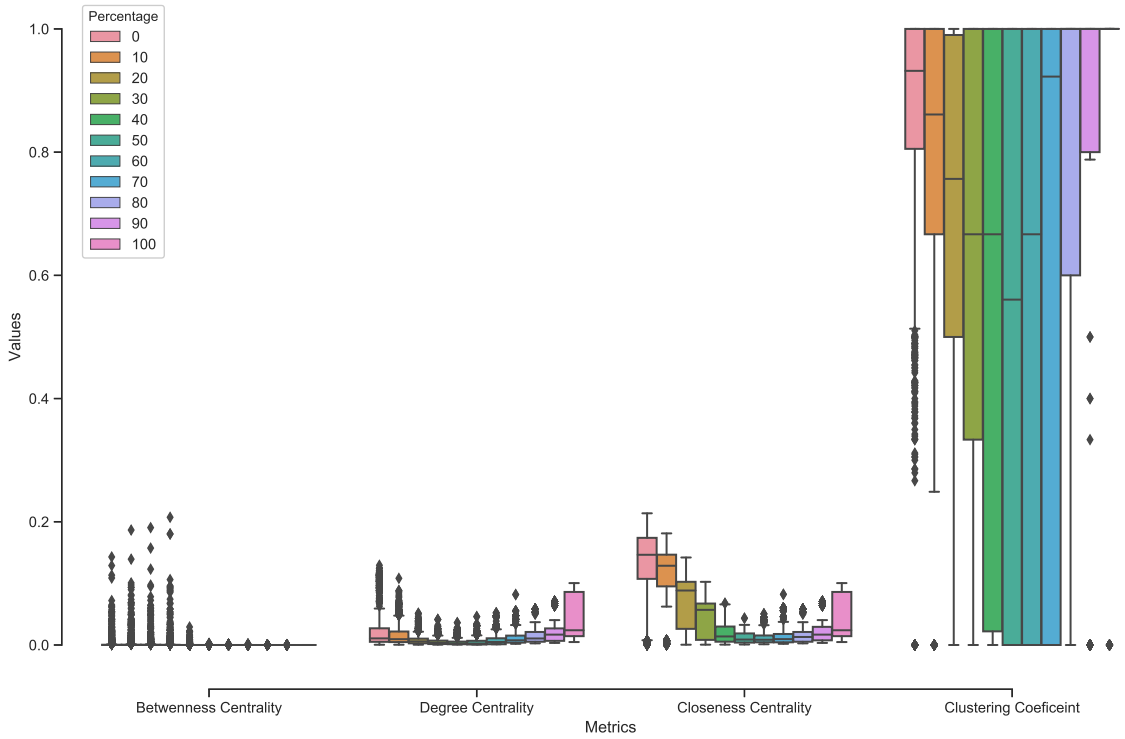

FIGURE 2.4: Measuring a set of network metrics to analyze the nodes' behavior from the generated network layers.

112 components, and then it is degraded. The density of networks is also providing additional information where the networks generated with higher LT (> 70) are shown to have a denser structure compare to the lower LTs.

Fig. 3.2 provides a closer look on the networks by showing the results of centrality metrics (betweenness, degree, and closeness), and clustering coefficient. As shown in the plot, the values of betweenness centrality of nodes are quite low. This is expected from the datasets because the teams of NECTEC are organized such that they work on certain domains of their specialty. Hence, the number of teams that lie in the shortest path within other nodes are very small. The degree centrality of the nodes have not changed dramatically while changing the threshold. The network layers that are not constructed with the very small or high value of LT are shown to be only collaborating with a consistent set of teams, although with different level of collaboration. Besides in the both extreme cases where the collaboration is very low, i.e., $LT = 0$, and very high, i.e., $LT = 100$, the maximum number of teams that a particular team is in collaboration with is higher than the other the other networks. The closeness centrality reveals another insight of nodes, where in our set of networks with lower contribution level the maximum number of teams who are acting as broadcasters in the network is relatively high. Clustering coefficient hast the most variant values in the networks, where in $LT = 50$ it reveals a normal distribution covering the full spectrum. Even though this network is the most fragmented layer within the set of network layers, there are well-cluster shaped components within the network. Overall, all networks reveal a high clustering coefficient which matches with the nature of the networks as collaboration data. In particular, networks with $LT < 30$ and $LT > 80$ the clustering coefficients are considerably high (> 0.65)

TABLE 2.3: Analyzing the generated networks for a global perspective. The number of nodes (#Nodes), number of links (#Links), number of connected components (n_{comp}) as well as network density (d) are calculated.

LT	#Nodes	#Links	n_{comp}	d
0	2334	65162	36	0.024
10	2330	43614	38	0.016
20	2228	19083	45	0.008
30	1960	10146	65	0.005
40	1631	6442	93	0.005
50	1282	4562	112	0.005
60	826	2950	99	0.009
70	526	1898	91	0.014
80	379	1299	77	0.018
90	298	1084	60	0.024
100	210	875	43	0.04

and the corresponding number of connected components and network density are small and high respectively. Hence, among all network layers these networks present components in which the teams are intended to have more stronger collaborations. The correlation between the each metric and LT is well represented in Fig. 2.5 [18].

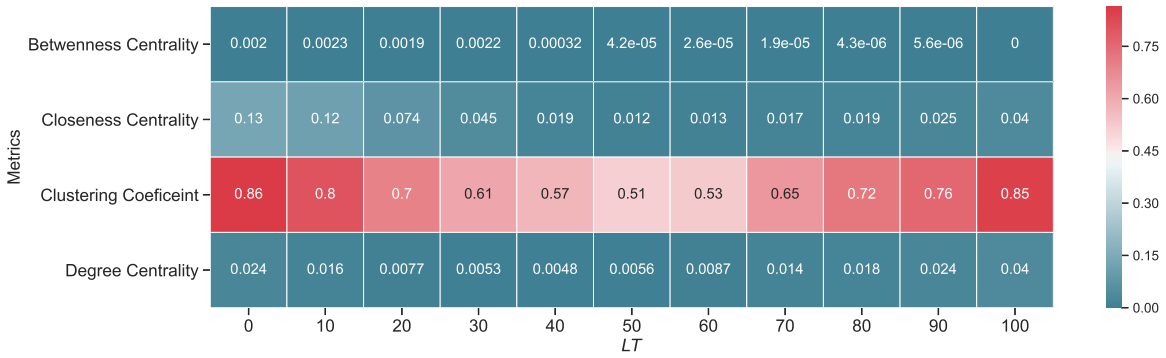

FIGURE 2.5: The correlation between the measured metrics and LT values in the generated networks.

We also generate a correlation matrix, illustrated in Fig. 2.6, to investigate on the correlation between the values of the metrics we have extracted from different generated network layers. The high correlation between the metrics reveal that each network layer is informative and reveals a particular aspect of the collaboration structure that we have constructed.

2.5 Conclusion

In this chapter, we introduced an approach that automatically transforms relational collaboration data into network layers. In the network layers, nodes represent collaborative teams, and connections are created under certain conditions that is defined in the approach. The

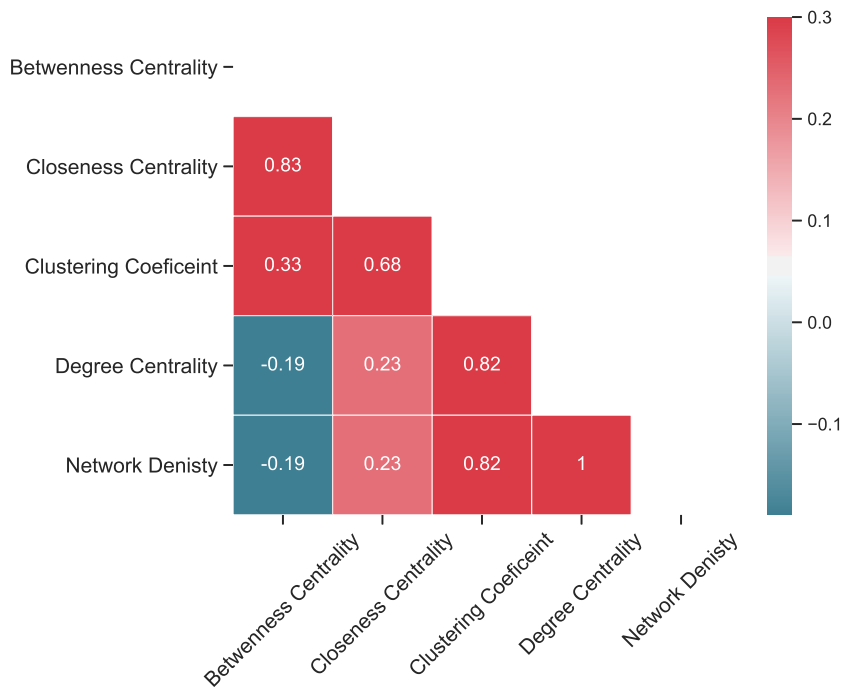

FIGURE 2.6: The correlation matrix of the metrics from the generated networks with different LTs.

condition depends on the *Linkage Threshold* (LT) parameter that is defined over the number of collaborators in teams and their contribution percentage. LT control the level of contribution in the process of constructing network such that we can evaluate different range of contributions. As results, we present different network layers each presenting an organizational perspetive from the collaboration data. We conducted netwrok analysis using metrics such as clustering coefficient, closeness and betweenness centrality, and illustrate their impact on the different network layers. The network analysis on the generated network layers shows different behaviour in each layer. We, then, utilize the results of the metrics as an important input to visualize the generated graphs in each configuration. We conclude that the LT has a crucial impact on the network properties and must be chosen with caution.

Our methodology reveals several optimization criteria. The influence of the LT on the results of the metrics indicates that the network representation can be optimized. Moreover, the LT can be generalized to a utility function to be performed on any given collaboration dataset. In addition, deciding on an optimal LT based on predefined criteria and conditions could further improve the performance, but also widen the applicability, of our algorithm.

Chapter 3

Impact of Network Topology on Community Detection

Transforming data into networks may have impacts in the results of network analysis. In this chapter, we followed our investigation on generating the networks from the real-world collaboration dataset. we investigated the definition of the fundamental research question of how and which network representation to choose for a given set of data. The drawback of previous studies is that they only consider the existence of a collaboration between individuals to connect them in the network. However, our work proposes a standardized method to produce networks from large and complex datasets.

In this chapter, we define a method to construct scientific collaboration networks from the data considering different features describing the collaboration. Furthermore, we benefit from the scientific collaboration dataset of *National Electronics and Computer Technology Center (NECTEC)* to examine our method. Interestingly, our results indicate that identifying a network construction model leads to a less noisy yet well–shaped community structure network with high modularity score. These results are published in Frontiers journal as "**Link definition ameliorating community detection in collaboration networks**" [25].

3.1 Introduction

Collaboration networks are social structures which indicate the relationship between collaborators who perform on the same tasks. Collaboration is an essential component to define the success of today's knowledge sharing ecosystem [37] and establishment of innovation.In collaboration networks, nodes represent individuals (aka collaborators) and links between them imply a collaboration. The analysis of collaboration networks can reveal information about the most likely behavior of individuals and groups in the network [38] such as discovering the interaction patterns [39], the evolution of collaboration communities [40] and predictive models on the productivity and longevity of collaborations [41].

One prominent property studied in the context of collaboration networks is the community structure of nodes [42]. The discovery of communities, with dense intra-connections and comparatively sparse inter-cluster, can be beneficial for various applications such as discovering common research area of potential collaborators [43]. Various network-based community detection algorithms are used for this purpose, e.g., *Louvain*'s algorithm [20], Label Propagation Algorithm (LPA) [44].

Most collaboration data are stored in relational databases which are used to extract the collaboration networks to perform network analysis. The context of scientific collaboration networks has been initiated with the studies of [17], [45]. The network is defined such that the researchers are represented as nodes and the links constructed if at least one paper happened to be published by them. Other studies such as [41] have followed a similar generative approach to construct the collaboration network from the dataset. In a recent study [46], a weighted scientific collaboration network has been proposed such that links are weighted by the number of papers. One drawback of previous studies is the elimination of other potential features that represent the collaborations (e.g., date, number of citations). The information which is attached to the data can substantially impact the underlying network representation and, therefore, the outcomes of network analysis (e.g., community detection). Thus the appropriate use of network analysis, substantially depends on choosing the right network representation [47], i.e., the definition of nodes and links [16]. Besides, in some cases, the definition of the link also requires determining a *threshold* which can significantly alter the outcomes of network properties, e.g., network density [48].

3.2 Collaboration Dataset

We benefit from a particular collaboration database provided by the *National Electronics and Computer Technology Center (NECTEC)* that presents different projects and collaborations in the area of R&D[1]. The whole database is the knowledge management about projects within distinct deliverables where the key information is to know project contributors and contributions. The database consists of three datasets, each indicates a particular deliverable: *PAPER*, *PROTOTYPE*, and *IP* (intellectual property) conducted between July 2013 and July 2018.

The datasets of combined research teams information consist of approximately 8000 records which correspond to the information of more than 2300 projects. Detailed statistical information regarding each dataset is provided in Table 3.1. Overall, NECTEC has more than 1000 members who are contributing to different deliverables with certain features that have been evaluated by the organization. For each researcher who collaborated on a

[1]National Electronics and Computer Technology Center (NECTEC) (https://www.nectec.or.th/en/)

contribution, a contribution percentage has been recorded. Another feature named IC–score which is designed by NECTEC, evaluates the scientific value and the outcome of contributions. For instance, producing a prototype in an industrial stage has a higher impact than one in the laboratory stage. For each project, the IC–score is divided between each contributor considering their individual participation in the project. Overall, each dataset of the deliverables contains a) project ID, b) collaborator's ID, c) contribution percentage of a collaborator for each project d) IC–score of a collaborator for each project.

Deliverable Type	# Researchers	# Projects	Cont. percentage	IC–score
PAPER	576	1717	$\mu = 22.22, \sigma = 19.73$	$\mu = 3.89, \sigma = 4.61$
PROTOTYPE	524	539	$\mu = 15.54, \sigma = 13.73$	$\mu = 9.41, \sigma = 10.75$
IP	489	630	$\mu = 25.15, \sigma = 24.42$	$\mu = 4.08, \sigma = 4.63$
Total	1056	2347	$\mu = 20.78, \sigma = 19.82$	$\mu = 5.81, \sigma = 7.73$

TABLE 3.1: General overview of the datasets from NECTEC. Contribution percentage (Cont. percentage) and IC–score are feature extracted from the dataset and describe the collaboration.

3.3 Methodology for Link Construction

We propose a *collaboration score* function that takes into account the combination of features extracted from the dataset. The purpose is to quantify the contribution of researchers considering features describing the collaborations. The collaboration score is the key element to define the link in the network while nodes are co–authors. We introduce a *linkage threshold* (LT) on obtained collaboration scores. Thus, multiple networks are produced using various LT values.

We define the *collaboration score* function based on the features extracted from the NECTEC datasets which includes a) the number of projects, b) the contribution percentage of researchers, and c) the IC–score of researchers. Given two researchers i and j worked on a mutual project p, i.e., (i, j), let n be the number of projects that i and j have collaborated, and $p_{k,i}$ and $p_{k,j}$ represent the contribution percentage of researcher i and j, respectively for the k^{th} project. Likewise, $s_{k,i}$ and $s_{k,j}$ indicate the IC–score of each researcher on the k^{th} project. Hence, we determine the *collaboration score* function as follows.

$$f_{i,j} = \frac{1}{n}\Big(\frac{1}{2}\sum_{k=1}^{n}(p_{k,i} + p_{k,j}) + \frac{1}{2}\sum_{k=1}^{n}(s_{k,i} + s_{k,j})\Big) \tag{3.1}$$

The function takes into account the average of IC–score and contribution percentage between any tuple of collaborators. The LT, then, is defined such that it determines different levels of collaboration score in the network. The range of LT varies from 0 to 1, which is the normalized range of collaboration score. In a nutshell, increasing LT enlarges the number of collaborations.

The threshold values indicate links in the network between the nodes. We produce a set of networks considering various LTs. Algorithm 3.1 shows the pseudocode of the data transformation to networks. A relational dataset of collaborations is the input of the algorithm. The researchers are determined as nodes of the network. For each tuple of researchers, the collaboration score is measured (see line 4). In order to generate a network, links are produced considering a particular LT value. All collaborations that are less or equal than the level of the chosen threshold are determined as links in the network (see line 7). Considering various levels of LT, a set of networks are generated by the algorithm which is examined in Sect. 3.4.

Algorithm 3.1 Network Extraction from Data

Input: D, scientific collaboration dataset
Output: $\mathcal{G}$, a vector of generated networks

procedure TRANSFORM-TO-NETWORK(D)
  $colList \leftarrow$ researchers from D
  for $tuple(i, j)$ in $colList$ **do**
    f.append $\leftarrow collaborationScore(tuple(i, j))$
    $collaboration$.append$\leftarrow$ Concatenate $tuple(i, j)$ and $normalize(f)$
  for LT in range($normalize(f)$) **do**
    if $collaboration.normalize(f) \leq LT$ **then**
      $nodes$.append($[i, j]$)
      $links$.append($[tuple(i, j)]$)
    $G \leftarrow Network(nodes, links)$
    $\mathcal{G}$.append G
  return $\mathcal{G}$

3.4 Results

Our proposed method has been employed on different deliverable types of the previously described NECTEC collaboration data. As a result of the extraction process, our method returns a set of corresponding collaboration networks. In the first stage, we exploit the distribution of the collaboration score (f) within each dataset. Next, we analyze the topology of the extracted networks given the different values of LT by measuring a set of network metrics. Furthermore, for each generated network, we identify the communities using the *Louvain* algorithm and evaluate their quality.

3.4.1 Data Processing

We exploit the histogram and cumulative distribution function (CDF) of f for each dataset of deliverables from NECTEC. Figure 3.1 describes the frequency and distribution of the obtained f after normalization. The average (μ) of f for *PAPER*, *PROTOTYPE*, and *IP* are 0.24 (standard deviation ($\sigma = 0.16$)), 0.18 ($\sigma = 0.12$), and 0.3 ($\sigma = 0.21$) respectively. Furthermore, the figure also shows that the majority of collaborators have relatively low number of contribution. Nevertheless a small number of collaborators are highly collaborated in various projects.

3.4.2 Topological Analysis

We analyze the topology and structure of extracted networks from each dataset by calculating a set of network metrics: degree, network density, transitivity, clustering coefficient, betweenness centrality, and closeness centrality. Figure 3.2 describes the evolution of these metrics on a set of 41 networks while increasing LT from 0 to 1 with the step of 0.025.

The degree of a node in collaboration networks represents the number of direct collaborations for each individual. The average node degree of networks obtained from *PAPER* is 6.59, *PROTOTYPE* is 11.46, and *IP* is 5.71 which indicates that on average, teams in *PROTOTYPE* had significantly higher collaborations compared to others. As illustrated in Figure 3.2, the degree of extracted networks does not change significantly. The reason is after a certain threshold of LT, the number of new links which have been added to the network does not grow significantly while the number of nodes stays constant. A similar scenario occurs when measuring network density. The network density calculates the ratio of existing links to the number of all possible links in a network such that a density close to 0 identifies a sparse network while a density equal to 1 is a complete network. With LT close to zero, the network

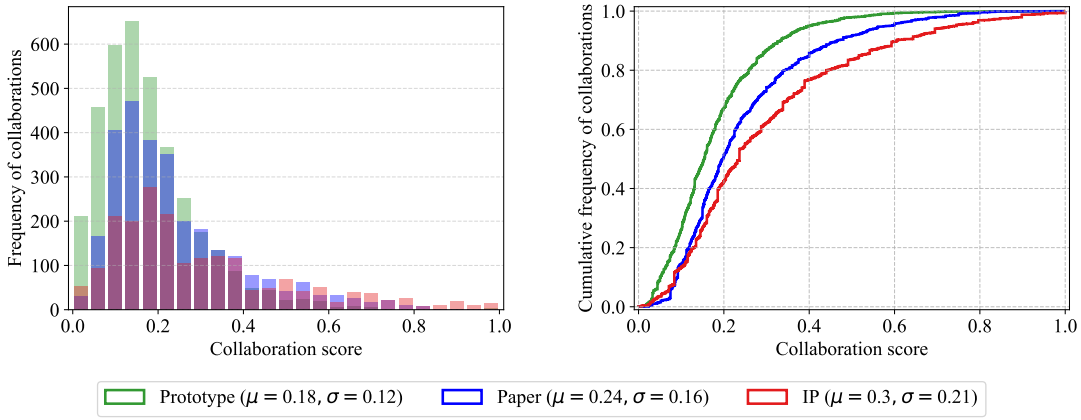

FIGURE 3.1: The histogram and cumulative distribution function (CDF) of generated collaboration score (f).

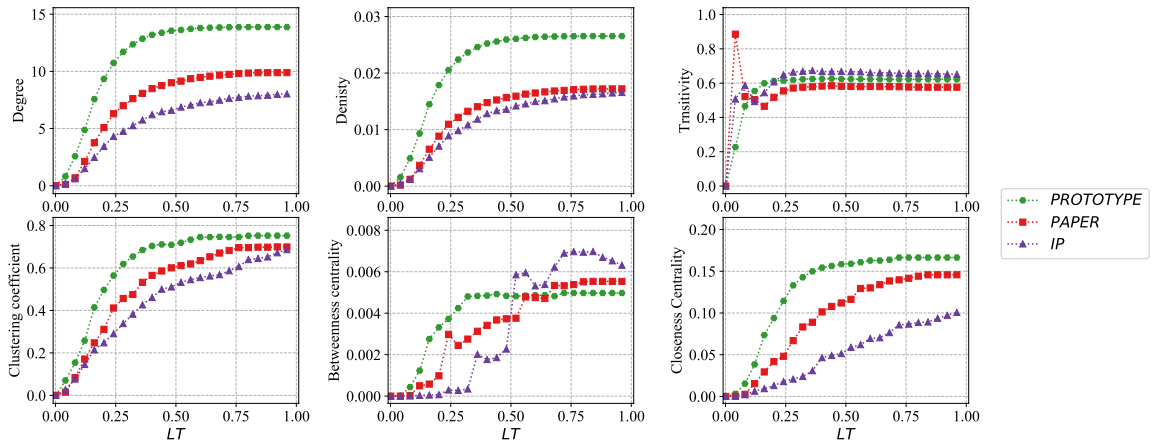

FIGURE 3.2: Topological analysis of a set of 41 produced networks from each dataset while increasing LT from 0 to 1 by 0.025.

mostly consists of isolated nodes which explains why in all three datasets the network density is close to zero. Eventually, the density of the network increases slowly and remains steady. The reason is due to the high number of nodes compared to the number of collaborations between the nodes. This indicates the fact that in real-world collaboration networks each collaborator may only collaborate with a small number of collaborators, hence, the networks are considered as rather sparse.

In order to get knowledge on the complexity of collaborations of each dataset, we calculate the transitivity and clustering coefficient of networks. Transitivity refers to the extent to which the relation that relates two nodes in a network that are connected by a link is transitive. Thus, it represents the symmetry of collaborations in our networks and forms triangles of collaborations. Figure 3.2 illustrates fluctuations for networks constructed with lower LT, however, quickly it approaches a consistent value.

On the other hand, clustering coefficient describes the likelihood of nodes in a network that tend to cluster together [49]. The average clustering coefficient of produced networks is 0.44 for *PAPER*, 0.61 for *PROTOTYPE*, and 0.45 for *IP*. For a relatively high LT the clustering coefficient approaches approximately to 0.7. A possible explanation can be that contribution of at least three people happens often in scientific collaboration teams [50]. Therefore, every collaboration that has three or more co–authors increases the clustering coefficient significantly.

Centrality measures indicate the importance of nodes in the network. We measure betweenness centrality and closeness centrality to analyze datasets. For a node, the betweenness is defined as the total number of shortest paths between every pair of individuals in the network which pass through the node [35]. In other terms, it highlights collaborators who act as a bridge between different groups in a network.

Moreover, closeness centrality defines the closeness of a node to other nodes by measuring the average shortest path from that node to all other nodes within the network. Hence, the more central a node is, the closer it is to all other nodes [34]. All three datasets reach the highest closeness centrality after a certain threshold. However, each dataset reflects a considerably different growth function, such that *IP* follows a linear function after each evolution, *PROTOTYPE*, and *PAPER* are growing exponentially.

3.4.3 Community Detection Analysis

We imply *Louvain* community detection algorithm to evaluate LT on *collaboration score*. We extract communities of each network and measure the modularity and number of clusters. The modularity of communities illustrates the strength of connected nodes inside the same community compare to the community of a random graph (with the same size and average degree). The higher the modularity, the more the network is closer to a well–shaped community structure.

Figure 3.3 shows the average results of 200 experiments on each dataset with the error bars which are too small. The figure shows that the modularity of all three datasets converges to relatively a high score of approximately 0.7 after a certain LT. It indicates that the produced collaboration networks have well–defined community structure compare to the random network of the same size. As illustrated in this figure, increasing LT does not affect the modularity after a particular point. For the lower LT (< 0.4), as also shown in Figure 3.2 networks have a considerably lower density, thus, they are sparse. However, the score increases exponentially and becomes steady for all three datasets for $LT > 0.4$. On the other hand, increasing LT decreases the number of communities considerably. When networks are sparse (i.e., $LT \leq 0.2$) the number of communities is almost equal to the number of nodes.

Moreover, as illustrated in Figure 3.3, the modularity score increases significantly even for the low values of LT and reaches to its highest value before it decreases and becomes steady. On the other hand, the number of communities exponentially decreases. Therefore, the network obtained from $LT < 0.2$ has an extremely high number of communities. In a particular case for *PROTOTYPE*, the modularity increases and becomes steady with $LT > 0.4$, and similarly the number of communities become constant ($= 22$) with $LT > 0.5$. Furthermore, considering the growth of metrics for *PROTOTYPE* from Figure 3.2, all metrics are constant with $LT > 0.4$.

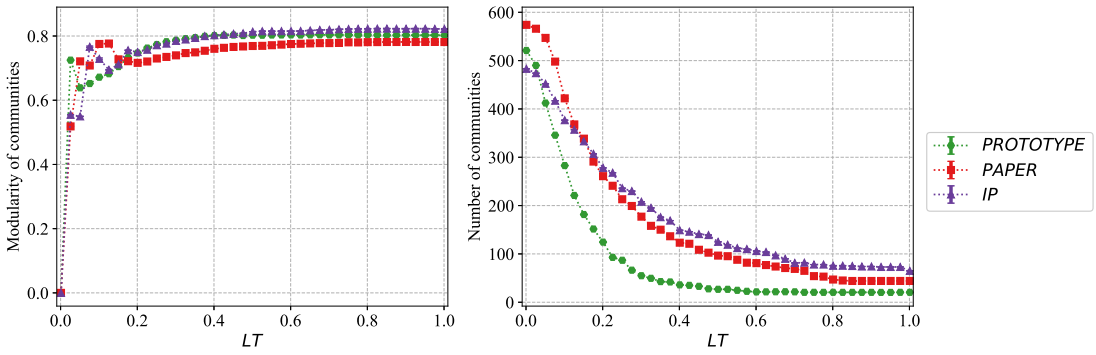

FIGURE 3.3: Community detection analysis after implying *Louvain* algorithm on networks produced with different LT values. The Community modularity score, and the number of clusters are the average of 200 experiments for 41 data points. The error bars are not visible because the standard error is very small.

3.5 Conclusion

The approach outlined in this chapter infers collaboration networks of researchers within projects of an organization. Our method uses the features describing the collaborations of a research institute and quantifies them by applying a proposed *collaboration score* function.

Our results show that the quality of the detection of communities from the extracted collaboration networks can differ significantly by the choice of the linkage threshold. It turns out that a greedy increase of links and connections can lead to a noisy network structure where the *identity* of nodes could be affected by a large amount of superfluous connections. Consequently, our future work has to focus on the understanding of a networks preference towards a rich network while avoiding a noisy structure [51]. Moreover, our experiments on the execution time of community detection indicate that increasing LT impacts the execution time of the algorithm. Hence, one option is to generate the network choosing a considerably low threshold while the modularity of communities is still at the highest possible value.

In this study we use a set of network metrics and the modularity score to evaluate communities of obtained networks. However, as future work we are looking at advancing our collaboration score model for network construction from relational data. Moreover, we consider identifying the optimum LT in order to recognize high quality communities within the obtained networks.

Chapter 4

Locality in Community Detection

Early approaches of community detection algorithms often depend on the network's global structure with a time complexity correlated to the network size. Local algorithms emerged as a more efficient solution to deal with large-scale networks with millions to billions of nodes. This methodology has shifted the attention from global structure towards the local level to deal with a network using only a portion of nodes. Investigating the state-of-the-art, we notice the absence of a standard definition of *locality* between community detection algorithms. Different goals have been explored under the *local* terminology of community detection approaches that can be misunderstood.

This chapter probes existing contributions to extract the scopes where an algorithm performs locally. Our purpose is to interpret the concept of locality in community detection algorithms. We propose a *locality exploration scheme* to investigate the concept of locality at each stage of an existing community detection workflow. We summarized terminologies concerning the locality in the state-of-the-art community detection approaches. In some cases, we observe how different terms are used for the same concept. We demonstrate the applicability of our algorithm by providing a review of some algorithms using our proposed scheme. Our review highlights a research gap in community detection algorithms and initiates new research topics in this domain. The results are published in the ACIIDS 2021 conference proceedings as "**Community Detection in Complex Networks: A Survey on Local Approaches**" [52]

4.1 Introduction

Densely connected components are inseparable from networks providing structural or functional roles of the applications represented by the network. Community detection algorithms aim to identify these densely connected components within a network. Each community consists of nodes that are similar or close to each other more than other nodes outside the community. The existing community detection algorithms can be differentiated into categories of global and local approaches. Unlike global approaches, local methods are known to discover communities without the integral global structural information of the complex networks [53–55].

The primary goal of developing local community detection algorithms is to find a local community of a given node in the absence of global information of the network [56, 57]. Utilizing the traditional global algorithms, that requires fetching a large-scale network, often produce structural hairballs and not meaningful communities [58] as studied in protein folding networks [59]. The initial solution of finding a local community structure for a given node has been further developed to detect all network communities. Therefore, locally detecting communities turn to answer today's large-scale networks that is one of the drawbacks of the global algorithms that tend to find all network communities using complete network information.

Motivation. The question is, then, what is defined as locality when it comes to community detection algorithms? Among various interpretations, one may define it as finding local community(s) [56, 57] of a given node(s). In contrast, others infer it as a local approach by incorporating local information of a network to find all network communities [24,58,60]. The majority of the studies still lay down in a spectrum within these two classes. For instance, they exploit the entire network to extract information used in the core community detection operation, whereas the objectives that define a community are determined locally [61–64]. Considering all the above-mentioned points, we noted the absence of a comprehensive study that supports the need to define a standard terminology regarding the locality of community detection algorithms to analyze the existing approaches deeply in this field. Our goal is to address these gaps in this chapter.

Contribution. We raise a new research challenge on community detection approaches concerning the *locality* in different stages of the algorithm. We explore the corresponding concepts and terminologies in various references, yet often with different terms. We also investigate the working flow of community detection approaches that benefit from a locality level in their approach. We developed Locality Exploration Scheme (LES) to incorporate research questions on the *locality level* of an approach in each step of the algorithm. Our scheme surveys existing approaches and countermeasures from a broad perspective respecting the algorithm's input, the core workflow of community detection, and the resulting output. Employing our model, we analyze some of the references concentrating on the stages defined in our scheme and discuss the applied locality level.

To the best of our knowledge, no studies have previously addressed the mentioned challenges. Our scheme is the first model to assemble strategies and associated locality levels to develop a community detection algorithm with a predetermined level of locality.

4.2 Preliminaries and Background

We assume G is a network denoted by $G = (V, E)$, where V is the set of nodes, and E is the set of edges representing links within pairs of nodes (v, u) such that $v, u \in V$. Each node $v \in V$ has a degree of k_v representing the number of its neighbours from $\Gamma(v)$, the neighbour list of v.

Definition 1. *Community Structure.* We define a community structure c as a sub-network of G, where the intra-connectivity is maximized compared to the inter-connections such that $c_i \cap c_j = \varnothing$ and $\bigcup c_i = V$.

Definition 2. *Local Community Structure.* As introduced by [56], a local community has no knowledge from outside of the community. It consists of core node(s) that are internal to the community such that they have no connection to the outside of the community, and border nodes that connect those core nodes to the unknown portion of the network (i.e., other communities).

Definition 3. *Community Detection Algorithm.* The algorithm that detect densely connected components of community structures in a network is known as community detection algorithms. *Local*

community detection algorithms tend to discover local community structures of G. We define *Local Detection of Communities* as an algorithm that associate a level of locality in its process to detect all communities of a network.

Definition 4. *Source Node.* A set of nodes chosen according to a score (e.g., similarity and centrality) to represent a community structure are identified as source nodes. They are also sometimes referred as core, seed and central nodes in the literature. In most of the cases, identifying a source node initializes a community of a network.

Definition 5. *Locality Level.* Adopted from [65], we define a three-level spectrum of locality. Starting from the most relaxed level, *global-level* that has no constraints, then *community-level* that is limited to the information within the community, and finally, *node-level* locality as the most restricted level which incorporates only local information of a node (up to certain extension, e.g., second-neighborhood).

Definition 6. *Auxiliary Information.* Some approaches require extra information to operate, for instance a threshold value, or a node/link weight. In this chapter, the extra information added to the process are considered as *auxiliary information*.

Definition 7. *Community Expansion.* A community detection algorithm often needs an expansion strategy to enlarge the initial source nodes or preliminary detected communities. It can be a fitness function to evaluate the membership of a node to a community or a modularity objective to measure the inter-connectivity of the community.

4.3 Locality Exploration Scheme (LES)

The question of how much information algorithms need from a network for their operations is not a recent research question [66, 67]. Stein et al. [68] have provided a classification of local algorithms in network research. They define a four-level model based on auxiliary information, non-constant run time, and functionality. We find this classification comprehensive in communication networks, however, it is limited when it comes to community detection criteria. Other comprehensive studies on community detection are confined to the two main categories of *local* and *global* algorithms [53–55, 69–72] emphasizing global methods. Thus, we identify this absence of attention to local community detection in complex networks.

We provide Locality Exploration Scheme (LES) illustrated in Fig. 4.1 and combine the challenges raised in Section 7.1. The scheme considers a three-level model for community detection algorithms: *Input data*, *Community detection flow*, and *Output communities*. In each step, we collect the possible solutions from the literature and sort them with the locality level described in Section 4.2. In the following subsections, we explain each stage provided in Fig. 4.1 for the possible solutions investigated from the literature.

4.3.1 Input Data

The initial step towards any community detection algorithm is the input provided to the algorithm. One of the main drawbacks of the global community detection is being dependent on the entire network to discover the communities. Although the community detection flow in local algorithms does not depend on the global structure, the input data includes the whole network for preliminary operations in several cases.

Considering large-scale networks, it is impossible to fetch the whole network for the next operations. Therefore, even if an algorithm offers an adequate locality level during some steps of the community detection flow (i.e., community expansion), it may fail to operate on a network if the input is the entire network. Hence, this is an essential stage when investigating the locality of the algorithm. Besides the network's information, sometimes, the algorithm also expects to input certain auxiliary information, such as a threshold value. Depending on the auxiliary information, sometimes it may also impact the level of locality.

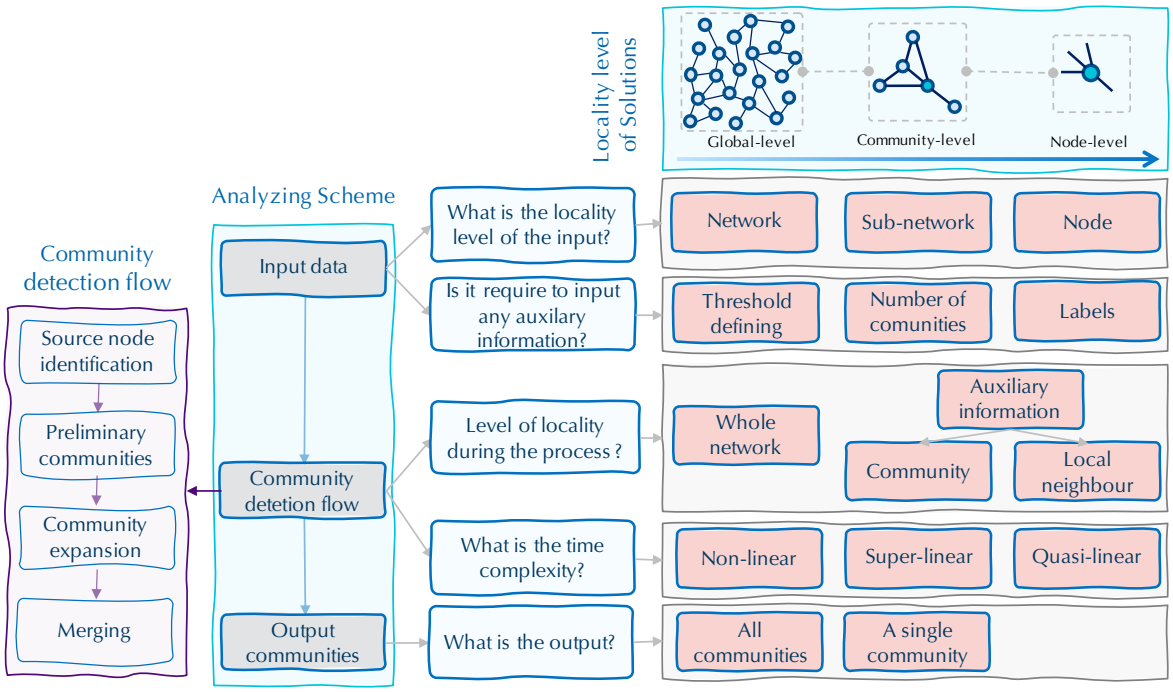

FIGURE 4.1: An overview of proposed LES model. The *analysing scheme* represents a three-level structure of community detection approaches. In the middle rectangles, the main challenges that may raise at each stage is highlighted. The pink rectangles show the existing solutions in the literature sorted from left to right based on their locality level.

4.3.2 Community Detection Flow

We assembled the core operations towards a community detection procedure in this stage. Principally, the procedure is decomposed itself into four functions as described in Fig 4.2. We noticed that when there is a discussion on community detection locality, it mainly refers to this stage. It is worth noting that not all algorithms follow the four-step model, in some instances some steps are combined (e.g., Source node identification and Preliminary communities). It is not a trivial task to decide about the locality of an approach based on this stage. A number of algorithms have included the entire network during the source node identification, however, they have increased the locality by incorporating local information while expanding the communities [61, 63, 73, 74]. Thus, we analyze this stage given the workflow described in Fig. 4.2.

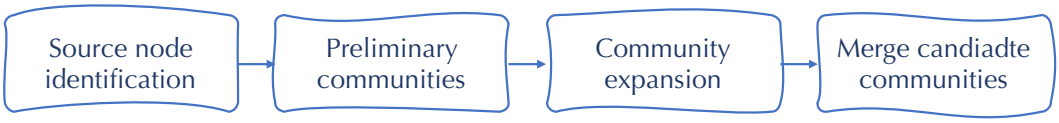

FIGURE 4.2: Community detection flow.

Source node identification.

Source node identification is one of the main steps that targets candidate nodes to be expanded later in order to shape the communities. The performance of the algorithm, however, depends highly on this step since the source nodes initiate output communities. Each contribution has introduced a slightly different approach to choose the source nodes. Besides algorithms that apply a random strategy (e.g., LPA) other tend to find the important nodes that is a good representation of its community to start

their approach from. In this step, a dedicated score is first calculated for a particular set of nodes (or the entire network) and then usually the list of scores is sorted to choose the best candidates as source nodes. We categorize the source node identification techniques into the following main classes:

- Network centrality metric [75, 76]
- Node similarity score [77, 78]
- Combination of topological measure (e.g., [74])

Table 4.1 summarized metrics used in reference for source node identification in some community detection approaches. It is noteworthy that the metrics, especially similarity scores, are not exploited only for identifying the source nodes but also as a similarity measure to quantify a node's belongingness to a community that we explore in the next subsections.

Besides the impact of the source node selection on the performance, it is also important to remind that not all of the metrics are local. Several metrics listed in Table 4.1 are required the knowledge from the entire network as the score that calculates both the degree and distance of a node from other high degree nodes. On the other, to choose source nodes, it is mostly observed that V is required to be sorted based on the chosen score (ref. Table 4.1).

TABLE 4.1: A summary of similarity scores of two nodes in the literature. ($\gamma(v)$ is the number of subgraphs with 3 edges and 3 vertices, one of which is v, $\tau(v)$ the number of triples on v, $\sigma_{st}(v)$ represents the shortest path from s to t through v s_{uv} represent a similarity score between node v and u.)

Categories	Metrics	Definition	References
Centrality	Degree	$k_v = \lvert\Gamma(v)\rvert$	[79–82]
	Clustering coeficient	$cc = \frac{\gamma(v)}{\tau(v)}$	[74]
	Betweenness centrality	$bc = \sum_{s,t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$	[79, 83]
	Edge density	$Den(G) = \frac{\lvert E\rvert}{\lvert V\rvert(\lvert V\rvert-1)/2}$	[84]
Similarity	Jaccard's Coefficient (JC)	$s_{uv} = \frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\lvert\Gamma(u)\cup\Gamma(v)\rvert}$	[61–63]
	Adamic-Adar Coefficient (AA)	$s_{uv} = \sum_{t\in\Gamma(u)\cap\Gamma(v)} \frac{1}{\log k(t)}$	[85]
	Resource Allocation (RA)	$s_{uv} = \sum_{t\in\Gamma(u)\cap\Gamma(v)} \frac{1}{k(t)}$	[77, 81, 86]
Combination	Degree - distance	$sc_i = k_v \times \sigma_v$	[73, 74]

Preliminary communities.

After detecting source nodes, the initial communities are predefined and most of the time, they are considered preliminary communities. Thus, this step may not be taken as an independent stage in the community detection flow. In many references, yet, this step is developed to extend source nodes into preliminary communities. It can be operated as merely taking the first neighbourhood of a source node as its preliminary community [62], or choosing neighbours relying on a similarity score (ref. Table 4.1) [61, 63]. The level of locality depends on the taken strategy. Per only the local neighbourhood provides a higher level of locality compared to other solutions.

Community expansion.

Several references have conducted an adequate locality level only during community expansion, regardless of previous input data. A list of local community expansion strategies developed in the literature mostly rely on community-level local information: internal connection and external connections of a

community [69]. A list of important local fitness function are provided in Table 4.2. Moreover, some approaches do not depend on such objectives. Instead, they exploit techniques such as spreading of influence (i.e., LPA) [58, 73, 74] or random walk.

Moreover, many references measure these functions (ref Table 4.2) from a community perspective such that a community calculates a respective objective to decide about adopting a new node. In other words, it is the community that determines whether to accept the joining of a new node to its community (if that node maximized the objective function) or not. By slightly changing the perspective, one can operate any of these functions on a node to decide on surrounded communities to join [24, 58, 60]. By employing this strategy, the locality level of an approach increases to the node-level.

TABLE 4.2: Summary of existence fitness functions.

Fitness Functions	Formula	Reference
Local modularity (Clauset) [87]	$R = \frac{\sum B_{ij}\sigma(i,j)}{\sum B_{ij}}$	[80]
Local modularity (Lou) [88]	$M = \frac{E_{in}}{E_{out}}$	[62, 89]
Fitness Function (Lancichinetti) [82]	$F_c = \frac{f_{in}^c}{(f_{in}^c + f_{out}^c)^\alpha}$	[63]

Merging candidate communities.

In certain instances, communities are small or sparse. Hence, identifying The merging step is not always a requirement; however, it can increase the quality of resulted communities [58, 61, 62]. The approach is accomplished by considering each community as a node and pursuing a similar approach to finding the most similar communities to merge. In some cases, it also requires a given threshold to decide on the degree of similarities between two communities [58, 61, 63]. Overall, the function requires communities to operate and its locality level can be community-level if the threshold value is not relying on the global structure of a network.

4.3.3 Output Communities

Finally, the communities are identified that can either represent a set of communities from the entire network [58, 60, 61, 74], or only a local community of a given node (subset of nodes) [56, 57] depending on the purpose behind the community detection approach. The early work is primarily motivated by finding local communities of a given node [56, 57]. Considering the underlying network, it is possible only to have meaningful local communities within a network [59] rather than global communities for the entire network.

4.4 Analyzing Existing Algorithms based on LCE

In this section we provide a review analysis of some papers regarding the scheme described in Section 4.3.

NSA.

In [61], authors have proposed an algorithm (NSA) founded on Jaccard similarity. The algorithm requires a network G and a threshold value used during the merging process of community detection flow as input. The source identification relies on the high degree nodes. Afterwards, the preliminary communities shape, adding the most similar neighbours due to the Jaccard similarity scores. The produced small communities are then merged similarly based on a given threshold on the Jaccard score between two communities this time. NSA functions on time complexity of $O(n \log(n))$. The algorithm

is global at the input level; however, it is operating locally given only nodes local neighborhood during the community detection flow. The outputs are all communities of a given network.

ECES.

The algorithm [62] is motivated by the drawback of global community detection algorithms that require the network's global information to operate. However, the model itself needs a network to process the first step of community detection flow. That is to obtain the core nodes of the network using an extended Jaccard score. The score admits local information until the second-neighbour of a node. The highest score node is then extended, including its first neighbours forming preliminary communities. Next, each community is extended by the ratio of internal links to the external ones. Finally, candidate communities satisfying the condition relying on the sum of its nodes Jaccard score are merged. The output is a set of network communities. The algorithm operates in a super-linear time complexity of $O(n \log(n))$. Similar to NSA, this approach also depend on global information to detect the source nodes, however, it enjoys a level of locality (second-neighbourhood compared to the first-neighbourhood of NSA) while extending the communities in community detection flow.

InfoNode.

In a recent approach [63], authors propose a model that concedes the increment of particular local community modularity as a condition of adopting a node. InfoNode requires both a network and a threshold value as input. Even though the source selection relies on a node degree, a local centrality metric, the approach needs to sort all nodes in regard to their degree. Therefore, this step is not local anymore. The high degree nodes are first enlarged to preliminary communities calculating the F fitness function [82], and then extended based on an internal force function defined by the authors. The growth of communities in the community detection flow is processed locally. The algorithm has a non-linear time complexity of $O(n^2)$.

DEMON.

Slightly different than the previous papers, DEMON [58] defines locality as taking each node to be responsible for joining a community. However, the algorithm requires a global network and a threshold value to process. The source nodes are chosen randomly, and an ego network for each node is identified. The local expansion of the candidate nodes is operated similarly to the LPA technique. Finally, sparse communities are merged considering the threshold value. The time complexity of the algorithm is quasi-linear reported as $O(nk^{(3-\alpha)})$. Regardless of the input, all the steps during the community detection flow have been operated in a community-level locality.

LCDA-SSN.

In another approach [60], authors developed a community detection algorithm that has increased the locality level compared to similar approaches. The proposed method is an iterative model taken only a node as input. They consider each node knowing its first neighborhood; hence, it discovers the network while operating on one node. The approach offers a self-defining source node selection giving a score to a visited node based on local structural information. The score is updated each time, as for the community cores. The community expansion is adopted from M local modularity [88]. However, it is applied to a node rather than a community. The output is all communities of a network in a quasi-linear time complexity of $O(nk)$.

4.5 Conclusion

In this chapter, we bring up new research questions in the field of community detection algorithms in complex networks, highlighting two foremost challenges: first, the absence of a standard terminology when it comes to local community detection algorithms, and second, the gap between the interpretation

of locality and community detection algorithms. We provide a Locality Exploration Scheme (LES) model based on the steps of the community detection approach and incorporate the research questions raised by the concept of locality in each step. By employing our LES, we could survey the existing techniques and strategies required for developing a community detection algorithm with an adequate locality level. Furthermore, we provide a thorough review of some of the references showing the applicability of our scheme. Our analysis can also be taken as a guideline to choose the most relevant functions while developing community detection. We show that ignoring the problem of defining locality can lead to misunderstandings, and if not addressed correctly. We plan to further extend our scheme by including evaluation metrics determined for these approaches.

Chapter 5

A Local Community Detection Algorithm with Self-Defining Source Nodes

Considering the growing size of existing networks, *local* community detection methods have gained attention in contrast to *global* methods that impose a top-down view of global network information. Current local community detection algorithms are mainly aimed to discover local communities around a given node. Besides, their performance is influenced by the quality of the source node.

In this chapter, we propose a community detection algorithm that outputs all the communities of a network benefiting from a set of *local* principles and a *self-defining* source node selection. Each node in our algorithm progressively adjusts its community label based on an even more restrictive level of locality, considering its neighbours local information solely. Our algorithm offers a computational complexity of linear order with respect to the network size. Experiments on both artificial and real networks show that our algorithm gains more over networks with weak community structures compared to networks with strong community structures. Additionally, we provide experiments to demonstrate the ability of the self-defining source node of our algorithm by implementing various source node selection methods from the literature. The results are published in the proceedings of International Conference on Complex Networks and Their Applications as "**Local community detection algorithm with self-defining source nodes**" [60]

5.1 Introduction

Complex networks exhibit modular structures, namely communities, which are directly related to important functional and topological properties in various fields. They can, for example, represent modules of proteins with similar functionality in a protein interaction network [53], or affect dynamic processes of a network such as opinion and epidemic spreading [90]. Despite the various insights and applications communities represent, they are all referred to as a densely connected set of nodes with relatively sparse links to the rest of the network. This simple definition, however, has raised great interest in discovering communities in complex networks. Numerous solutions have been proposed ever since. While most of the conventional algorithms are rooted in a top-down view obtaining the *global* information of the entire network [55, 71], others reduce the problem to a local level, by availability of a part of the network [57, 87] to find local communities of a given node(s). The existing local community detection algorithms in the literature are mostly designed to first identify a set of source nodes to initialize the community detection [62, 73, 76, 79] and then use a local community modularity to expand the communities [82, 87, 88]. The main challenges raised by these methods fall into the followings: i) the optimal result highly depends on the source node selection [73], ii) the main goal is to discover the local communities of a given set of nodes rather than all communities of a network, iii) the approaches are mostly operating in a relaxed level of locality, i.e. *local-context* the forth level of locality [68], exploiting the information of a part of the network in the community detection process, iv) even though they appreciate a level of locality while employing the algorithm, they cannot cope with any changes in the network which is mostly the case in real-world complex networks.

Taking the above-mentioned considerations into account, we propose a community detection approach that has two main properties: First, it is operating solely based on a node and its local neighbours at a time, thus, it can belong to the *local-bounded* category, introduced by Stein et al. [68], which is one level more restrictive compared to most of the state-of-the-art approaches. Secondly, it does not depend on any auxiliary process of source node selection. Instead, it is exploiting a self-defining source node that can adapt based on the local neighbourhood knowledge. Our algorithm progressively iterates over the discovered part of the network allowing each node to decide on joining one of the neighbour communities or even create a new community. We develop a community influence degree employing topological measures [25] to identify the community influence of each node to assure maintaining a hierarchical community structure centralized by high-degree nodes. We, then, perform a local modularity measure to label each node's community. This way, our algorithm addresses the challenges raised by the previous algorithms by proposing a local approach based on a self-defining source node.

Many of *local-context* community detection algorithms are founded on this assumption that the global knowledge of the network is not available, therefore, the community structure measures should be in dependant of those global properties [87] such as modularity metric Q in Girvan and Newman [91] *non-local* community detection algorithm. A variety of source (i.e., seed) selection techniques are employed by *local-context* algorithms to increase the quality of communities. Some of these methods are based on the network's centrality metrics such as degree [79], others exploit similarity metrics [76] like the Jaccard score [62], while others defined new metrics, for example, node density in [73]. With all the advantages that centrality based community detection algorithms offer, they tend to give relatively poor results in dense networks and perform better in sparse networks [53]. In the next step, benefiting from a fitness function or a local community modularity, the chosen seeds are expanded. Clauset defines a local community modularity [87] as $R = \frac{\sum B_{ij}\sigma(i,j)}{\sum B_{ij}}$. It measures the ratio of the number of links within the community (i.e., internal links) to the sum of the number of all internal and external links. Luo et al. [88] have simplified the above measure and define local modularity as $M = \frac{E_{in}}{E_{out}}$, which only divides the number of internal links of a community to the number of external links. Next, Lancichinetti et al. [82] propose a fitness function as $F_c = \frac{k_{in}^c}{(k_{in}^c + k_{out}^c)^\alpha}$, where, k_{in}^c and k_{out}^c represent the internal and external links of a community c. It requires a parameter α to control the size of the communities. While the above-mentioned algorithms are considered in the *local-context*

class of local algorithms [68], other algorithms perform with even more restricted local properties of a network categorized as *local-bounded*. These algorithms deploy entirely based on a node and the information from its neighbourhood. In an approach for wireless ad-hoc networks, Brust et al. [24] proposed an adaptive k-hop hierarchical community detection that performs using only neighbour local information. In another study [58] the authors proposed a community detection algorithm by giving the authority to nodes to vote for the community that they might belong to. Our local-bounded algorithm offers a change of mindset such that nodes are responsible to choose their community based on a self-identifying source selection. To expand the communities, we define a local modularity similar to Luo et al. [88] by engaging both internal and external links in the fraction.

5.2 Preliminaries of Local Community Detection

In this section, we introduce the preliminaries and notation that are used in the rest of this chapter. We assume an undirected and unweighted network $G = (V, E)$, where V and E represent the set of nodes and the set of links, respectively. Our goal is to discover a set of all communities $C = \bigcup c_i$, such that each node $v \in V$ belongs only to one community. A *good* community is achieved if all nodes within a community are densely intra-connected, in other words, implying that the local modularity of each community is maximized. Besides, we construct a community in a hierarchical structure in such a way that nodes with a higher degree are pushed towards the center of the community whereas the lower degree nodes stay close to the border of the community. We aim to find all communities of a network by allowing each node to adjust its community label given its local neighbours, $\Gamma(v)$, and their properties at a time. We exploit a set of measures adopted from the network structure to assure that each node belongs to a community at the end of the execution time.

Definition 1. *(Community influence degree.)* Each node is influenced by its surrounding communities. To quantify this impact, we define $\lambda(v)_{c_i}$ to show the level of impact from node v with community label c_i to its neighbours, as follows:

$$\lambda(v)_{c_i} = \frac{k_v}{hl}, \tag{5.1}$$

where k_v is the degree of v (i.e. the number of nodes in $\Gamma(v)$), and hl shows the hierarchy level of v in its community. In a nutshell, hl represents the hop distance from the source node in the community. The value is 1 for source nodes, showing the first layer of the hierarchy (i.e., seed node) and increases by per hop-distance towards the border of the community. The intuition behind this measure is that a node is more likely to be in the same community as another node if the following node is closer to the source of the community and has a higher degree. Thus, we indicate the *strength* of a member in a community with a high $\lambda(v)$ value showing the high degree and low hierarchy level of that node.

Definition 2. *(Local community modularity.)* It defines the degree of a node contributing to a candidate community c_i. It is measured by the following equation:

$$\mu(v)_{c_i} = \frac{E_{in} - E_{out}}{E_{in} + E_{out}} = 2\frac{E_{in}}{E_{in} + E_{out}} - 1, \tag{5.2}$$

where E_{in} is the number of edges from node v towards the community c_i, E_{out} represents the outwards of node v. Therefore, $k_v = E_{in} + E_{out}$ is the total number of edges of v or simply the degree of node v. In other words, the local community modularity explains a membership degree for a given community. It represents the link ratio of those neighbours of v within a community minus the number of those outside the community, normalized by the degree of v. The value can vary in the range of $(-1, 1]$. It takes a negative value if it does not have any connection to the community c_i and positive if the majority of its links are toward the community.

5.3 Self-defining Local Community Detection

We design an iterative bottom-up approach allowing each node to take a decision of joining a community independently. Our algorithm discovers the whole network starting from a given node and

its local neighbours, therefore, it performs in a restricted level of locality (i.e. local-bounded). The algorithm converges when all nodes agree with their community labels. We assume a hierarchical structure for each community by encouraging high degree nodes towards the center of the community and nodes with a lower degree to the borders while maximizing the local modularity defined in Eq. 6.2. To forge a hierarchical structure, we adjust the hop-distance hl, and in the meantime, we update each node's community influence degree $\lambda(v)$ as defined in Eq. 6.1. The metric is considered as a level of attraction to encourage a node towards a community. On the other hand, to extend communities or to prevent emerging large communities we initially filter communities by measuring the local modularity from Eq. 6.2.

Algorithm 5.1 Local Community Detection Algorithm (LCDA)

Input: Node v, and $\Gamma(v)$
Output: C set of communities

1: *Initialization*:
2: $R \leftarrow v$
3: $v.hl = HL$
4: $v.cl = v$
5: *Procedure*
6: **while** stopCondition **do**
7: **for** v in R **do**
8: **if** $\deg(v) > \deg(\Gamma(v))$ **then**
9: $v.hl \leftarrow v.hl - 1$
10: $v.\lambda = \lambda(v)$
11: $v.\mu = \mu(v)$
12: $v.hl, v.cl \leftarrow Alg.\ 5.2(v)$
13: $R \leftarrow update(\Gamma(v))$
14: **return** $C \leftarrow R.cl$

Algorithm description. The general structure of the proposed local approach to detect communities of a network is described in Alg. 5.1. To extend the communities we define a set of principles that are explained in Alg. 5.2. The procedure starts by initializing the node list R (line 1), that records visited nodes and their neighbours. As a first-time-visited node in the list, the community label cl and hierarchy level hl of the node will be initialized to its node ID and a constant value HL, respectively (line 2-3). We chose HL to be 4 initially, however, it can be any value larger than 1. The next step is to adjust the node's hl value, its value will be reduced if it has the highest degree compared to its neighbours (line 6-7). Afterwards, the community influence degree $\lambda(v)$ and the local modularity $\mu(v)$ is calculated (line 9-10). To update both hl and cl of v, we input the node through some principles defined in Alg. 5.2 (line 11). Besides, the list R will be updated by the neighbours of node v. Finally, if all nodes come to an agreement such that no further changes occur, the algorithm will converge and stop. Extracting the cl of all nodes in R results in obtaining all communities of G. A set of principles is defined in Alg. 5.2 to decide the corresponding community of the node v. First, choosing the common community label (mc), the local modularity is calculated. If $v.\mu$ was positive, v takes the same label as mc. Then, v adjust its hierarchy level by taking the minimum hl of that community and increase it by one unit as its hl value. Otherwise, if $\mu(v)$ was negative or zero, then, either v itself is selected by the neighbours to be a new community, or it will temporarily follow the best candidate among its neighbourhood.

Computational complexity. The complexity of the proposed algorithm, on a network of size n, and an average degree k can be estimated as follows. The outer *while* loop repeats until the algorithm has converged. The inner *for*-loop, depends on the length of R which progressively includes all the nodes from V. Starting from one node with degree k, in the worst case, R increases as follows: $\{1, k, k^2, ..., k^m\}$, while $k^m = n$, hence, it is in the order of n and can never be more than $O(m \times n)$,

Algorithm 5.2 Local community expansion

```
1: mc = common community label
2: bc = [u in Γ(v) if u.λ is max(Γ(v).λ)]
3:
4: if (v.μ > 0) then
5:     v.hl = min(Γ(v)).hl + 1
6:     v.cl = mc
7: else if (v.μ <= 0) then
8:     if v is mc then
9:         v.hl = 1
10:        v.cl = v
11:    else
12:        v.hl = bc.hl
13:        v.cl = bc.cl
```

TABLE 5.1: Dataset of networks used for the experiments.

Real-world networks with ground-truth								
Networks	n	k_{avg}	n_{com}	Description				
Zachery's Club	34	4.59	2	Zachary's karate club				
Football	115	10.66	12	American football game				
Dolphins	62	5.13	2	Dolphin social networks				
US Politics' Books	105	8.5	3	US Politics' Books				
Synthetic networks								
Networks	n	k_{avg}	μ	t_1	t_2	c_{min}	c_{max}	n_{com}
LFR 4000	4000	25	$0.1 - 0.8$	2	1.1	40	100	63
LFR 8000	8000	25	$0.1 - 0.8$	2	1.1	60	100	103
LFR 15000	15000	25	$0.1 - 0.8$	2	1.1	40	200	82

with m as the number of iterations in the outer *while* loop until the convergence.

5.4 Experimental Analysis

In this section, we examine the performance of our algorithm with different experiments. We exploit both real-world and artificial networks that are described in Table 6.2. Following artificial networks, we generate various networks using the LFR benchmark algorithm [92]. The mixing parameter μ, identifies the density of the networks, i.e. the strength of the communities.

We first compare the results of the proposed algorithm on the networks from Table 6.2 with a set of algorithms: Louvain [20] and Fast-greedy [56], and Label Propagation Algorithm (LPA). Next, to examine the ability of self-defining source nodes of our algorithm, we implement a set of source node selection methods from the literature and combine them with our algorithm. Finally, we provide tests to validate the analytically derived low complexity of our algorithm.

5.4.1 Evaluating Quality of Communities

To measure the quality of the results obtained from networks in Table 6.2, we calculate Adjusted Mutual Information (AMI). This metric is an adjustment of the Mutual Information (MI) score to account for chance. It accounts for the fact that the MI is generally higher for two methods with a larger number of communities, regardless of whether there is actually more information shared.

We compare the results with the above-mentioned algorithms from the literature. The resulted communities of these algorithms are, then, used as a baseline to compare the performance of our

algorithm with. The results for the real-world networks are reported in Table 5.2 and for LFR benchmark networks are shown in Fig. 5.1. As shown in both results, our algorithm is comparable to

TABLE 5.2: The AMI quality metric results on the communities detected by Louvain, LPA, Fast-greedy, and our proposed algorithm (Proposed Alg.) on real-world networks. The bold values show the best results among other algorithms for each network.

Networks	Louvain	LPA	Fast-greedy	Proposed Alg.
Zachery's Club	0.46	0.48	**0.54**	0.45
Football	0.85	**0.87**	0.65	0.65
Dolphins	0.49	0.59	0.55	**0.88**
US Politics' Books	0.49	0.53	0.51	**0.56**

the other algorithms while processing entirely based on the local information and thus, benefiting from a low complexity. The algorithm gains more when the community structure of the network becomes weaker (i.e., μ is increased). The reason is that due to the locality level, our algorithm behaves greedily in a situation where the conditions to join a neighbour community are not fulfilled, by generating new communities. Hence, it ends up with different communities than the other algorithms of Louvain and LPA, and similar to Fast-greedy.

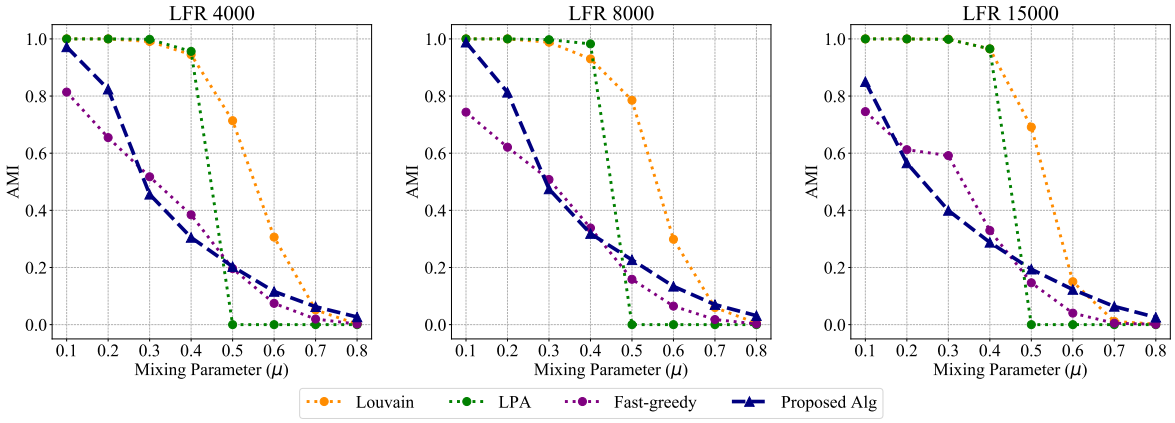

FIGURE 5.1: AMI results on the LFR benchmark networks explained in Table. 6.2.

5.4.2 Source Node Selection Analysis

Most of the existing local community detection algorithms require a source node selection before the community expansion. We implement some of the source node selection methods from the literature and develop an experiment to analyze the impact of source node selection on our algorithm. We choose different centrality and similarity scores: degree centrality [79], extended Jaccard metric [62], and node density (to find nodes with high degree, however, distant from each other) [73]. In order to be fair on choosing the best candidate nodes, we apply an outlier detection technique, Interquartile Range (IQR), to select nodes with higher scores. We then adjust the hl of these nodes to be known as the initial communities of the network and proceed as described in Alg. 5.1. We evaluate the methods on an LFR benchmark network with $n = 2000$ and report the results in Fig. 5.2. The results show that there are no differences between the proposed algorithm (Basic) and its variations by each source node selection (e.g., Basic+Degree). As shown in Fig. 5.2, our method maintains a self-identifying source node selection considering node degree.

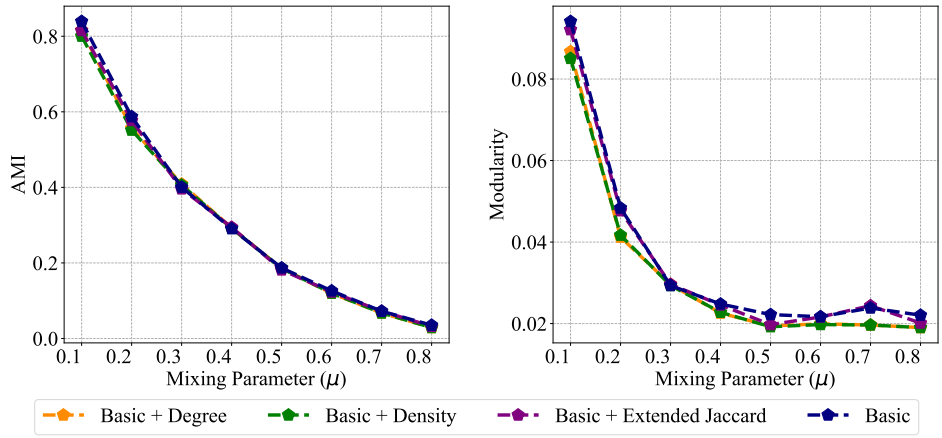

FIGURE 5.2: Employing different source node selection methods from the literature on the bases of the proposed algorithm. The methods are examined over the LFR 2000s network exploiting AMI and Modularity measures.

5.4.3 Computational Complexity Analysis

Following the experiments on the networks provided in Table. 6.2, we analyze both the number of iterations our algorithm requires to converge (the outer loop in Alg. 5.2) and the number of nodes from the list R that were qualified to the conditions thus, are forged to change adjust their properties (i.e., hl or cl) in Alg. 5.2 which are not all the nodes in R.

The overall results are shown in Fig. 5.3 and Fig. 5.4. At each level of the mixing parameter (μ), from 0.1 to 0.8, for each network size, we calculated the number of iterations that the algorithm requires until convergence. As shown in Fig. 5.3, the number of repetitions does not rely on the size of the network and is slightly influenced by μ that shows the organization of community structures. However, regardless of n, the proposed algorithm converges in the average number of 8.2 iterations. Furthermore, with regard to the inner loop of the algorithm, we calculate a ratio of the number of nodes that are entitled to modify in each iteration to the size of the network. According to Fig. 5.4, the results reveal that the number of operations in each repetition of the algorithm has never reached n. It hits 87% of n in its maximum case which has mostly occurred from 3^{rd} to 5^{th} iterations. The number of modified nodes are considerably lower than the 3^{rd} to 5^{th} iterations that substantiates the low complexity of our algorithm.

5.5 Conclusion

In this chapter, we described our proposed community detection algorithm that is benefiting from a set of local principles and a self-defining source node selection to detect communities in complex networks. We developed the algorithm exploiting community influence degree and a local community modularity that are defined in this chapter. The community influence degree of a node increases if the node has a high degree and low hierarchy level in the community that is defined based on the hop-distance from the source node. This way, we shape communities in a hierarchical structure where nodes with higher degrees are towards the center of the community. Our algorithm exploits a set of local principles allowing each node to take a decision on its community label based on its neighborhoods local information. The algorithm is designed in a more restrictive level of locality compared to the current local algorithms and offers a linear order of computational complexity. We deploy extensive experiments to analyze the performance and efficiency of our algorithm. The experiments on both real and artificial networks show that the proposed algorithm performs better in networks with weak community structures compare to the algorithms that benefit from the global information of the network. Moreover, we perform experiments to validate the ability of self-defining source node selection

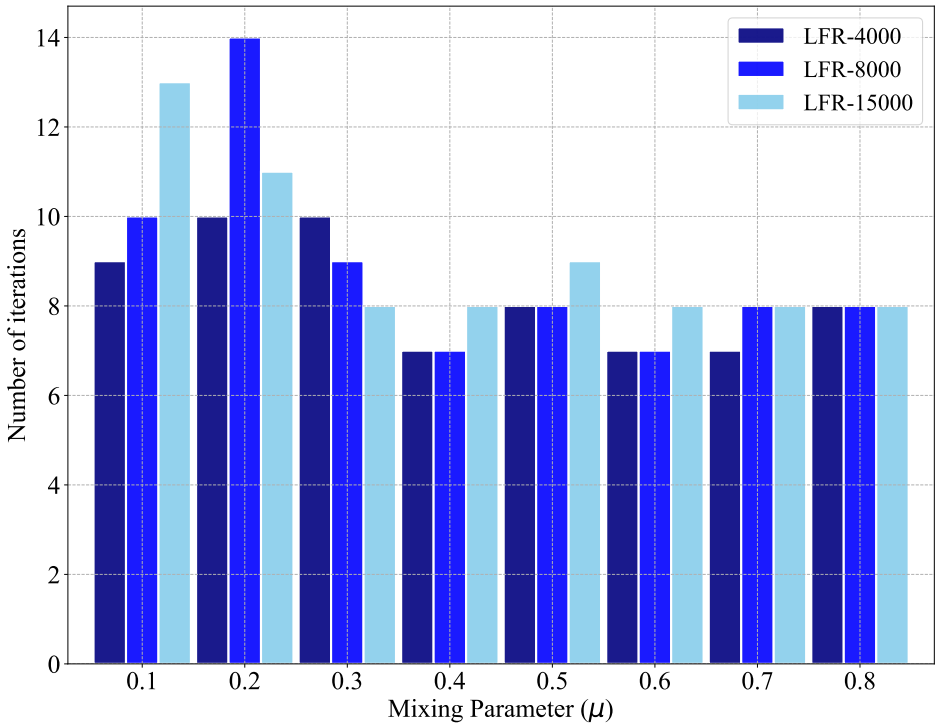

FIGURE 5.3: The results of experiments on the convergence of the algorithm on LFR networks, a bar plot showing the number of iteration.

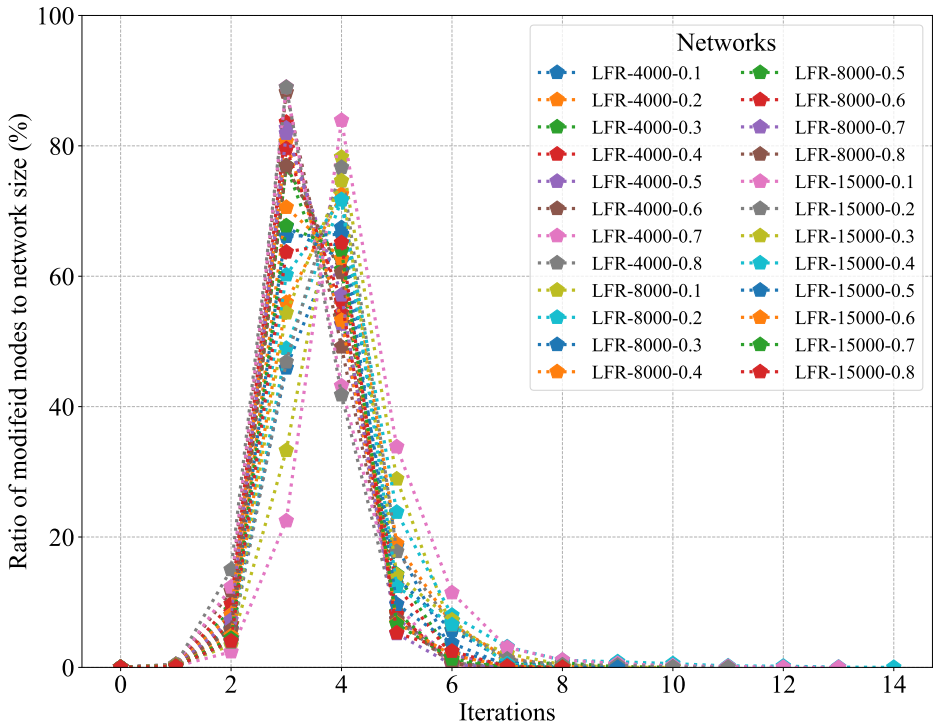

FIGURE 5.4: The results of experiments on the convergence of the algorithm on LFR networks, the percentage of the number of nodes modified per iteration.

of the our algorithm. We show that our algorithm performs independently from the source node selection methods in the literature. The experiments on the complexity of the algorithms demonstrate that, regardless of the size of the network, the algorithm converged after approximately 8 iterations, whereas, the number of nodes that are involved in the process has shown not to exceed the 87% of the whole network size. Remarkably, the locality and self-defining properties of this approach have equipped our algorithm for the future investigations on the adaptability to dynamic environments. Besides, we are planning to elaborate on the proposed approach by employing a local merging method on the output communities in order to increase the accuracy and performance of the results, while still holding the same level of the locality.

Chapter 6

Application in Biological Networks

Community detection is considered as a solution to many biological networks. In this chapter, we apply our proposed local community detection algorithm on protein-protein interaction (PPI) networks to identify protein complexes in this network. Several community detection algorithms are applied on (PPI) networks. Many of existing algorithms use global measures that operate on the entire network to identify communities. The result of using global metrics are large communities that are often not correlated with the functionality of the proteins. Moreover, PPI network analysis shows that most of the biological functions possibly lie between local neighbours in PPI networks, which are not identifiable with global metrics. Besides, the advancement of experimental techniques on PPI has motivated the generation of many Gene Ontology (GO) databases. Incorporating the functionality extracted from GO with the topological properties from the underlying PPI network yield a novel approach to identify protein complexes.

In this chapter, we exploited the capability of LCDA, our local community detection algorithm, by incorporating *functional* properties of proteins to detect protein complexes in PPI networks. We propose (LCDA-GO), that uniquely exploits information of functionality from GO combined with the network topology. LCDA-GO identifies the community of each protein based on the topological and functional knowledge acquired solely from the local neighbour proteins within the PPI network. Experimental results using the Krogan dataset demonstrate that our algorithm outperforms in most cases state-of-the-art approaches in assessment based on *Precision*, *Sensitivity*, and particularly *Composite Score*. We also deployed LCDA, the local-topology based precursor of LCDA-GO, to compare with a similar state-of-the-art approach that exclusively incorporates topological information of PPI networks for community detection. In addition to the high quality of the results, one main advantage of LCDA-GO is its low computational time complexity. The results are published in the Plos One Journal as "**From Communities to Protein Complexes: A Local Community Detection Algorithm on PPI Networks**".

6.1 Introduction

Proteins work cooperatively to accomplish biological functions. The physical interaction between proteins, known as *protein-protein interaction* (PPI), is the key for many biological functions [93], for example, the transcription of `DNA`, the translation of `mRNA`, and cell cycle [94]. Scientific progress on PPI is highly critical for applications such as protein function discovery [95], disease comprehension [96], and drug discovery [97].

To infer the physical interactions of proteins, a number of experimental techniques have been developed, such as *yeast-two-hybrid* (Y2H) [98] and *tandem affinity purification* (TAP) [99]. This resulted in the generation of several depositories and databases of experimental data on PPI (e.g., BIOGRID[1]). While these screening methods facilitate the comprehension of PPI, they have been widely criticized due to the false negative (i.e., not being able to detect interacting proteins) and false positive (i.e., identifying non-interacting proteins as interacting proteins) interaction detection. Therefore, high-throughput screening methods suffer from a considerable lack of accuracy and thus, produce an incomplete map of the interactions among the proteins [100–102].

The pairwise physical interaction of proteins within the PPI data suggests a network representation where nodes are the proteins and links are the interactions among the proteins. Exploiting network structure with network analysis tools on such data has shifted the PPI analysis to the *network* level. Besides, the existence of protein complexes justifies the high-degree clusters within the PPI network [101]. PPI networks inherit both *topological* and *functional* information [93]. The first term refers to the physical interaction describing the arrangements of the nodes in the network, and is associated with the densely connected proteins namely *communities*. The latter explains the biological function of proteins that are achieved by groups of proteins that bind each other and shape *protein complexes*. The complexes are explained by the annotations available in Gene Ontology (GO) [103,104] databases. GO provides a specific definition of protein functions and it is one of the main resources of biological information. GO provides a structured and controlled vocabulary of terms, which are subdivided into three non-overlapping ontologies: Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) [105]. We utilize GO terms to enrich PPI networks with functional properties of proteins.

It is acknowledged that in several cases, those proteins that are topologically interconnected represent similar biological processes (i.e., GO terms) [106], thereby there is an overlap between the communities of proteins and complexes. Nevertheless, the two terms are distinguished entities in PPI networks. Moreover, biological networks such as PPI networks share a common feature refereed as *local cluster connectivity* [59] that highlights the locality of the biological functions in PPI networks that are possible only between local neighbours.

Because of the correlation that exists between protein communities and complexes, detecting protein complexes from PPI networks can be translated into a community detection problem [94, 107,108]. The purpose of a community detection algorithm for PPI networks is to divide proteins into groups such that the proteins of the same group are more similar to each other rather than those in the other groups. The state-of-the-art solutions consider different objectives to divide the nodes of a given network into highly interconnected communities [53–55]. Some of these algorithms are adjusted to biological networks to tackle the protein complex detection in PPI networks [109], including C-FINDER, COACH, CLUSTERONE, MCL, CMC, MCODE, and CORE&PEEL. Even though the community detection algorithms drive optimal topological communities in PPI networks, they suffer from the particular biological nature of the network due to the disengagement of functional properties. [94,102,110,111].

The extracted interactions from experimental techniques (e.g., Y2H, TAP) are sometimes biased with incorrect inferring of existing and non-existing relationships. In other words, the available PPI networks could be incomplete and unreliable with respect to the detected nodes and links [101]. That in return will impact the results of the communities if the method depends solely on the existing topology of the network [25]. Moreover, some of the existing community detection algorithms acquire the whole network, that could be inherently incomplete, and hence results in large tangled communities of mixed or broad functionality [112] that do not explain adequately the underlying PPI network [111,113]. In addition, such algorithms perform based on the global measures that are expensive in time complexity.

[1] https://thebiogrid.org/

Encoding biological information in PPI networks could address the challenge of detecting higher quality communities of proteins with respect to their biological nature. The functionality hence could be achieved by incorporating biological information from the annotated databases (e.g., GO, DAVID). DCAFP [114], GMFTP [115], and MTGO [111] are some of the algorithms that are designed in a similar way. To tackle the next challenge regarding the reliability of the data and missing information, one possible solution could be to diminish the impact of network structure by focusing only on the local neighbours [60].

In this chapter, we propose LCDA-GO, a local community detection algorithm that combines topological and functional properties (i.e., GO terms) of PPI networks to detect associated communities that are representing protein complexes. One of the main advantages of LCDA-GO is the strong degree of locality [52] devised in the algorithm which not only reduces the dependency to the network structure but also equips the algorithm with a considerably low time complexity when compared to other state-of-the-art approaches. We compare LCDA-GO with the state-of-the-art algorithm that incorporates the topology and functionalities by exploiting GO to detect protein complexes. We also expand our experiments by providing a comparative evaluation with state-of-the-art protein complex detection approaches relying only on the topology of the network. For this experiment, we have used the LCDA algorithm [60], the local-topology based precursor of LCDA-GO.

6.2 Related Work

Many algorithms have been proposed to detect communities in PPI networks [94, 109, 116, 117]. Some of these approaches just rely on the topology of the PPI networks to detect communities, while others combine the biological functionality of the nodes to enrich the network and hence complex detection. We classify the existing community detection algorithms used for protein complexes in two categories based on the properties that an algorithm incorporates to detect the communities. We first explain community detection algorithms that perform solely on the *topology* of a network, and then, we discuss algorithms that rely on both *topology* and *functionality*.

6.2.1 Topological Approaches

One of the earliest algorithms that has been developed for PPI networks community detection is MCODE [118]. It enjoys a level of locality, by expanding a set of high-ranked nodes (i.e., source nodes) into communities. MCODE often represents very large communities and hence the number of predicted real complexes is small. The Markov Cluster algorithm (MCL) [119] is also utilized on PPI networks. The algorithm is a robust method based on a random walk to partition the network into communities. ClusterOne is a greedy approach starting from a seed node. The nodes with high cohesiveness are added or removed from the communities in an iterative process. ClusterOne is an overlapping community detection approach and it merges those groups of proteins that satisfy an overlap score.

For the comparative evaluation we used MCODE, MCL, and ClusterOne [120] to measure the performance differences of our LCDA algorithm, a version of LCDA-GO performing based on just local topological properties. Other algorithms such as COACH [121] and LCMA [122], and CFinder [123] also benefit from topology of the network to find the communities. These algorithms are discussed in [94, 109, 116].

6.2.2 Topological and Functional Approaches

Recent approaches benefit from functional enrichment of the network to accurately detect the communities of proteins in PPI networks. The main motivation of such algorithms lies in the fact that protein complexes are mostly aggregated in performing common functions. One of the earliest approaches in this category is RNSC [124]. This algorithm is initialized with a random partitioning that is optimized based on the minimum cost for node exchanging. It considers density and functional homogeneity to search for better communities. Its performance, however, depends on the initial community assignment. MTGO [111] is a recent approach that combines both topological and functionality of the PPI networks to detect the communities. Similarly to RNSC, MTGO initializes the process by a random

partitioning, and decides on rejoining the nodes into the communities if they share a common functionality and also if the new node increases the modularity of the community. The algorithm relies on two parameters *min* and *max* that control the size of the communities and impact the outcome. GMFTP [115] and DCAFP [114] are two other algorithms that are designed similarly by exploiting functionality, however, the biological nature of the networks are not directly involved in the main process and it is rather processed in advance by the network topology.

Our proposed LCDA-GO approach is similar to mentioned algorithms such that it combines both topological and functional information. However, unlike RNSC, MTGO, our proposed model does not rely on any random partitioning nor is restricted to initial input parameters. The results of LCDA-GO is compared to MTGO in Experiments and Results Section.

6.3 Local Community Detection Algorithm for Protein Complexes with Gene Ontology (LCDA-GO)

In this section, we introduce the basic notation and terminologies that will be used through the chapter. We also describe how LCDA-GO is implemented to detect communities of proteins exploiting topological and functional properties based on local conditional rules.

6.3.1 Notation and Preliminaries

We assume an undirected and unweighted network $G = (V, E)$, where V and E represent the set of nodes and the set of links, respectively. Our purpose is to divide G into set of communities, C, such that each node $v \in V$ belongs exclusively to one community c_i, and $C = \bigcup c_i$. A high quality community is a densely intra-connected (topology property) group of proteins representing lowest variation of GO terms (functional property). LCDA-GO finds communities based on both topological and functional properties in a local manner. The algorithm allows each node to adjust its community label, *cl*, given the local neighbourhoods.

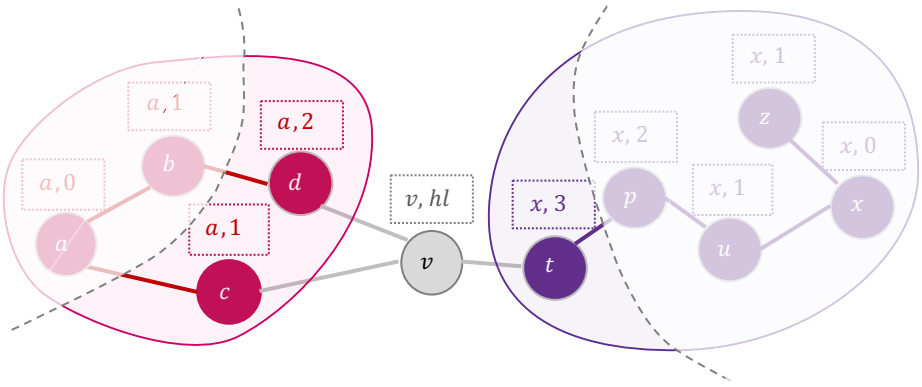

FIGURE 6.1: A snapshot of the community structures and local information that LCDA-GO is implemented on for node v. The transparent area is unknown zone that is not available during the operations. Thus, each node performs relying on the knowledge of its first neighbours. In this example, c and d are from community a and t is in community x. The community label describes the source node of the community, hence, a and x are two surrounded communities of v. The numbers attached to each node describes the hop-distance of the node from its community presenter. During the implementation, we have considered *hl* of a source node equal to 1 instead of 0.

On a given PPI network, LCDA-GO represents communities by a source node that is discovered

TABLE 6.1: Notation exploited in LCDA-GO

G	A PPI network
C	Set of solution that consists of communities of c_i such that $C = \bigcup c_i$
v	The current node
$\Gamma(v)$	Neighbours of node v
k_v	Degree of node v
cl_v	Community label of v
hl_v	Hop-distance from the community source node
g_v	GO terms of node v (i.e., functional properties)
$\lambda(v)$	Community influence degree on node v
$\mu(v)$	Local community modularity

during the algorithm. A source node is one of the high-degree nodes of the community and is connected to the nodes that have similar functional properties. The distance from the source node of a community to node v is stored in hl_v. A snapshot of LCDA-GO performance is illustrated in Fig 6.1 showing the process for node v. In this scenario, v has three neighbours $[c, d, t]$, such that node c and d belong to 'a' and t is from x (i.e., cl_c = 'a', cl_d = 'a', cl_t = 'x'). Besides, the numbers show hl of each node, that is the hop-distance from the source node of the community. According to this example node c and node d are 1 and 2 hops away from the source node of their community (i.e., a), respectively, and t is 3 hops away from its source node, x. It is worth mentioning that v does not have any other knowledge about the rest of the network as shown in the transparent zone in Fig. 6.1.

Besides the above-mentioned topological variables, cl and hl, that are consider in LCDA-GO, g is also determined to store GO terms that a protein is contributed. To access a decision on the community of node v, LCDA-GO calculates two parameters as defined in the following:

Definition 1. *(Community influence degree.)* The community influence degree of node v is calculated between v and its neighbours from community c_i as follows:

$$\lambda(v)^{u\in[\Gamma(v)\cap c_i]} = \ln(\frac{k_v}{hl_v}).|g_v \cap g_{u\in[\Gamma(v)\cap c_i]}|, \tag{6.1}$$

where $|g_v \cap g_{u\in c_i}|$ is the number of common GO functions between v and its neighbours from community c_i. The intuition behind the community influence degree is that a node is more likely to be in the same community as a neighbour node if the following node is closer to the source node of the community, has a higher degree, and shares similar functions with the neighbour node. If in a community one node has a higher community influence degree, the node could be a potential source node.

Definition 2. *(Local community modularity.)* The local community modularity for a node v is calculate for a surrounded community c_i as:

$$\begin{aligned}\mu(v)^{c_i} =& \frac{E_{in} - E_{out}}{E_{in} + E_{out}} \\ =& 2\frac{E_{in}}{E_{in} + E_{out}} - 1,\end{aligned} \tag{6.2}$$

where E_{in} is the number of links connecting node v to nodes from community c_i, and E_{out} represents the links to the other nodes. The value of local community modularity can vary in the range of $(-1, 1]$. It takes a negative value if there is no link to community c_i. The value is positive if the number of links connected to c_i surpasses the number of links to other communities. Local community modularity performs as a measure of community extension by adding v to c_i, if $\mu_v^{c_i}$ is positive.

A list of the notations used in the chapter is summarized in Table 6.1.

6.3.2 Algorithm Description

We propose an iterative bottom-up approach, LCDA-GO, allowing each node to take a decision of joining a community independently. Our algorithm starts from a node and discovers the network through each node's direct neighbours. LCDA-GO relies on a set of conditional rules to expand or generate new communities. The Local Community Expansion Rules (LCER) operate on each node based on the acquired local neighbourhood information as explained in Notation and Preliminaries Subsection. At each step of LCDA-GO nodes adjust their hop-distance (hl) value according to their distance from source nodes. If a node has a higher community influence degree and meets the conditions, it will become a source node. Thus, its hl is updated to 1. In this case, all neighbour nodes adjust their hl according to their hop-distance from the source node. LCDA-GO converges when all nodes agree with their community labels.

Algorithm 6.1 LCDA-GO

Input: Network G
Output: C set of communities

1: *Initialization*:
2: $R \leftarrow v$ from V
3: $v.hl = HL$
4: $v.cl = \texttt{v}$
5: $v.g = \texttt{GO}[v]$
6: *Procedure:*
7: **while** stopCondition **do**
8: **for** v in R **do**
9: **if** $k_v > \texttt{max}(k_{\Gamma(v)})$ **then**
10: $v.hl \leftarrow v.hl - 1$
11: $v.\lambda = \lambda(v)$
12: $v.\mu = \mu(v)$
13: LCER(v)
14: $R.\texttt{update} \leftarrow \Gamma(v)$
return $C.\texttt{update} \leftarrow cl$ from nodes of R

A pseudo code of the proposed LCDA-GO is described in Alg. 6.1 LCDA-GO. The algorithm starts by initializing the node list R (line 1), that records the visited nodes and their neighbours. The initial node is either a given node or randomly selected from the network. As a first-time-visited node in the list, the community label cl of the node is assumed as it ID, in this case, `v`, and its hop-distance hl is set to the constant value of HL (line 2-3). We chose $HL = 4$ initially, however, it can be any value larger than 1. The next step is to adjust $v.hl$: If $v.hl$ is the highest compared to v's neighbours, then it will be reduced by 1 (lines 7-8). Afterwards, $\lambda(v)$ and $\mu(v)$ is calculated (lines 10-11) and v is transmitted to Alg. 6.2 LCER (line 12) to make a decision regarding its cl. employing LCER on v, its attributes such as cl and hl will be updated consequently. Next, R expands by including neighbours of v. The processes continue such that all nodes of V is included in R and updated by LCER. Finally, if all nodes come to an agreement such that no further changes occur in community structure and each node of the network is declared in one community, the algorithm will converge. The `stopCondition` is defined as follows:

$$\texttt{stopCondition} = \begin{cases} 1, & \text{if } (R == V) \ \& \ \text{(for } v \text{ in } R, \ v.cl \text{ doesn't change)} \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

After the convergence of LCDA-GO, the set of communities is obtained by retrieving each node's cl from R.

We defined Alg. 6.2 LCER to decide the corresponding community of v. For an input node v, it first calculates the local community modularity. Instead of computing the function for each c_i, we only consider the larger community(ies) which has the larger number of links to v. We assume that `u` is the larger community. If $\mu(v)$ is positive, v joins community `u`. Thus, the community label of v changes to `u` (line 3), and the hop-distance shift to the shortest path from v to the source node u (line 4). To measure the shortest path, we simply consider the minimum hl of the neighbours plus 1. In case $\mu(v)$ is negative or zero, one of these two scenario may occur: First, the algorithm checks for the possibility of v itself being a source node. It means that node v is selected by the neighbours as the source node, while its attributes are not updated. Hence, the attributes of v are changed to fit the condition (line 7-8). Otherwise, v changes its attributes to follow the most similar node in its neighbourhood, which is node p with highest community influence degree (line 9-10). then, either v itself is selected by the neighbours to be a new community, or it will temporarily follow the best candidate among its neighbourhoods.

Algorithm 6.2 LCER

```
1: if (μ(v) > 0) then
2:     v.hl = min(Γ(v).hl) + 1
3:     v.cl = u (μ(v) <= 0)
4:     if v.cl is u then
5:         v.hl = 1
6:         v.cl = u
7:     else
8:         v.hl = p.hl
9:         v.cl = p.cl
```

6.3.3 Computational Complexity

The complexity of the proposed algorithm is determined by two loops in the algorithms. The outer *while*-loop in Alg. 6.1 LCDA-GO - line 5 coordinates the convergence of LCDA-GO to ensure that all nodes have come to an agreement about their community assignments. The recurrence (t) of the outer loop is independent from the size of the network. Our experiments with various networks' sizes [60] shows that $8 \leq t \leq 15$. The inner *for*-loop of LCDA-GO described in 6.1 line 6, operates a set of conditional rules over each node from list R. The performance of the inner loop has the highest impact on the overall complexity of LCDA-GO.

The complexity of the inner loop on a network G of size n can be estimated as follows. The repetition of the loop changes as R is updated. The list of neighbours (i.e., R) initially starts with the neighbours of node v. Let us assume k is the average degree of G. In this case, The initial size of R, in other words, the repetition of the first loop is k ($t_1 = k$). As R progressively is extended by adding other nodes, the next loop repetitions $t_2, t_3, ..., t_m$ increases as well. To calculate the complexity, we need to sum up all recurrences of the loop: $\{t_1 = k, t_2 = k^2, ..., t_m = k^m\}$. Considering the size of the network, the final R includes all nodes of G, therefore, $t_m = k^m = n$. Then, the complexity of the series that is combining the loops is $O(t \times n)$, with t representing the iterations over the outer *while*-loop. In addition, according to our experiments [60] $t\ log(n)$, hence the average complexity of LDA-GO is $O(nlog(n))$.

The worst scenario happens when the inner-loop runs over V instead of R. In this case, each iteration performs on n nodes instead of k. The recurrence of the inner-loop is then, $\{t_1 = n, t_2 = n, ..., t_m = n\}$. However, the iterations of outer-loop remains the same since it is independent from the inner-loop. Hence, the worst case complexity stays as same as the average complexity, $O(nlog(n))$.

6.4 Experiments and Results

In this section, we first describe the PPI network dataset, GO [104] terms that are used to enrich the network, and the benchmark dataset. Next, we define the metrics and measures that we use to evaluate the performance of our algorithms, LCDA and LCDA-GO. Finally, we provide a comparative evaluation to show the performance of our algorithm compared to state-of-the-art algorithms.

6.4.1 PPI Network and Gene Ontology (GO)

To evaluate LCDA-GO and LCDA, Krogan [125] dataset is selected. It includes a set of nodes (i.e., proteins) and associated links (i.e., interactions) built on yeast *Saccharomyces Cerevisiae* data. We download the dataset from BioGrid [2] database [126]. To include the functionality we exploit Gene Ontology (GO) terms from Panther[3] database [127]. GO terms are subdivided into three categories of Molecular Function (MF), Biological Process (BP) and Cellular Component (CC). We extract the GO terms of Krogan PPI network. For evaluating the outcome, we use gold standard protein complexes CYC2008 [128] as target sets to evaluate the predicted communities resulted from LCDA-GO. The information associated with the database and datasets are described in Table 6.2.

TABLE 6.2: Datasets of networks used for the experiments.

PPI Network					
Datasets	$\|V\|$	$\|E\|$	avg. degree	# CC	$\|G_{cc}\|$
Krogan [125]	2674	7079	5.29	62	2527
PPI + MF	1014	2135	4.21	7	995
PPI + BP	1154	2502	4.33	8	1130
PPI + CC	1160	2710	4.67	10	1130
PPI + All	1523	3708	4.86	9	1498
Gene Ontology (GO)					
Database	Proteins	# MF functions	# BP functions	# CC functions	All functions
Panther [127]	2358	8	11	3	22
Benchmark					
Database	Proteins	Complexes	# ∩ Krogan	# ∩ Panther	
CYC2008 [128]	1920	408	970	813	

Krogan PPI network [125] dataset, includes 2674 proteins in total. Our analysis found 62 connected components with a giant connected component including 2527 proteins, while 42 of the components had less than 3 nodes. For the community detection, we removed all those 42 components that will not shape a community. The final PPI network includes 2590 proteins.

We generated four PPI networks from the original Krogan PPI network according to GO term categories: PPI + MF, PPI + BP, PPI + CC, PPI + ALL, such that the last network includes all the functions. We also keep the original Krogan network without annotations for further analysis. All five networks are refined by filtering the connected components with the size of less than 3 proteins.

6.4.2 Evaluation Metrics

Before presenting the evaluation results, we describe various metrics that are mostly used in the literature [94, 111, 116, 117] to assess detected complexes in PPI networks. Exploiting these metrics, we then compare the state-of-the-art algorithms with our proposed algorithm and describe them.

[2] https://thebiogrid.org/
[3] http://pantherdb.org/

Neighbour Affinity Score

To quantify the similarity of the detected complex $p = (V_p, E_p)$ with the benchmark $b = (V_b, E_b)$, we use the neighbour affinity score (AS) as defined in Eq. 6.4. This metric considers both the size of the two complexes and the common proteins in the two sets to measure the similarity between the two. In case the predicted complex is exact equal to the real complex, then AS will be equal to 1. For two complexes of p and b the affinity score is defined as follows:

$$AS(p, b) = \frac{|V_p \cap V_b|^2}{|V_p|.|V_b|} \tag{6.4}$$

where V_p is the number of proteins from the predicted complex and V_b is the number of proteins in the benchmark complex. We define a threshold θ, $AS(p, b) \geq \theta$, to control the strength of the similarity measured by AS. We consider $\theta = 0.1$ to get results from all algorithms.

Precision, Recall, and F-measure

Among the standard metrics to evaluate the predicted values based on the benchmark are $Precision$, $Recall$, and $F - measure$. However, the metrics that we have implemented in this chapter for the evaluation are slightly different than the common definition for the $Precision$, $Recall$, and F-measure and are similar to [94, 129]. We use AS as defined in Eq. 6.4 to choose a good match between the predicted and benchmark complexes. Assume that p is a predicted complex from the set of all predicted complexes P, and b is a benchmark complex from set B that includes all benchmark complexes. In this case, N_{cp} and N_{cb} are defined as follows:

$$\begin{aligned} N_{cp} =& |\{\forall p | p \in P, \exists b \in B, AS(p, b) \geq \theta\}|, \\ N_{cb} =& |\{\forall b | b \in B, \exists p \in P, AS(p, b) \geq \theta\}|. \end{aligned} \tag{6.5}$$

Based on the N_{cp} and N_{cb} values from Eq. 6.5, Precision, Recall are defined as the fraction of the matched complexes from the predicted set P, and benchmark set B respectively, according to the Eq. 6.6.

$$Precision = \frac{N_{cp}}{|P|}, \tag{6.6a}$$

$$Recall = \frac{N_{cb}}{|B|}. \tag{6.6b}$$

The harmonic average of $Precision$ and $Recall$, known as F-measure, is then calculated as follows:

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6.7}$$

We use these metrics to evaluate the overall performance of the detected complexes over the complexes within the benchmark.

Sensitivity, Positive Predicted Value, and Accuracy

Besides the metrics defined above, $Sensitivity$ (Sn) (also called $Coverage$), $Positive\ Predicted\ Value$ (PPV), and $Accuracy$ (Acc) are used to evaluate the performance and accuracy of the detected complexes [94, 101, 117]. Consider T_{ij} equal the number of common proteins between i^{th} benchmark complexes and j^{th} predicted complex. N_i is the number of proteins the i^{th} benchmark complex. Given n is the overall number of b benchmark complexes and m predicted complexes p, then Sn and PPV are defined as follows:

$$Sn = \frac{\sum_{i=1}^{n} \max_j(T_{ij})}{\sum_{i=1}^{n} N_i}, \tag{6.8a}$$

$$PPV = \frac{\sum_{j=1}^{m} \max_i(T_{ij})}{\sum_{j=1}^{m} \sum_{i=1}^{n} T_{ij}}. \tag{6.8b}$$

Larger values of Sn indicate that the community detection algorithm has well-covered the proteins in the real complexes. On the other hand, PPV highlights the probability of true positives of protein complexes in predicted communities. The accuracy of the prediction, as a summary metric, can then be defined as as the geometric average of Sn and PPV as follows:

$$Acc = \sqrt{Sn \times PPV} \tag{6.9a}$$

In addition to the above-mentioned metrics, several studies [111,120,130] rely on another measure known as *Composite Score* [131] to make a comprehensive evaluation. Therefore, as a final global performance measure, we calculate the *Composite Score* by summing up the three values of *Precision*, *Sn*, and *Acc*. This value is important to avoid the advantage of evaluation metrics to another.

6.4.3 Comparative Evaluation

We provide a set of experiments to compare the communities resulted from our algorithm with the state-of-the-art algorithms. We compared LCDA-GO and LCDA [60] with MCODE [118], MCL [119], CLUSTERONE [120], and MTGO [111]. We choose these algorithms to explore the benefits of topological and functional properties in the performance of protein complex detection methods.

TABLE 6.3: An overview of the resulted communities from each algorithm including our method on *Saccharomyces Cerevisiae* Krogan interaction datasets.

PPI + MF					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	37	244	209	65	383
N_{cb}	4	160	142	69	167
N_{cp}	2	112	117	36	154
PPI + BP					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	38	256	236	71	416
N_{cb}	3	192	170	76	202
N_{cp}	3	149	146	51	196
PPI + CC					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	51	277	237	71	425
N_{cb}	6	196	180	80	210
N_{cp}	5	158	153	54	211
PPI + All					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	52	347	142	79	548
N_{cb}	4	213	122	78	223
N_{cp}	4	178	106	52	237

Except our two algorithms, LCDA and LCDA-GO, other algorithms require setting up initial parameters such as *min size* of the community, in their software. Clearly, tuning the parameters could result in better performance, however, there is no principled way to discover the optimal values for these parameters rather than using their defined values. Table 6.3 describes a general overview of the results of employing different community detection algorithms on PPI networks. In all experiments, we benefit from the gold standard protein complexes of CYC2008 [128] as the benchmark.

To provide fair comparisons and for a detailed analysis, we have designed two experiments. In the first experiment, we only consider the communities that are detected by the algorithms only considering the topology of the network, namely, MCODE [118], MCL [119], ClusterOne [120], LCDA [60]. The second experiment is for evaluating the communities resulting from algorithms that are incorporating both topology and functionality. For this evaluation, we compared LCDA-GO with MTGO [111]. The next two subsections present the comparisons of these experiments.

TABLE 6.4: Performance comparison of the communities of the algorithms that are based on only topology on *Saccharomyces Cerevisiae* Krogan interaction datasets. θ is 0.1.

PPI + MF							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.05	0.01	0.02	0.02	**0.65**	0.11	0.19
MCL	0.45	**0.39**	**0.42**	0.26	0.60	**0.39**	1.11
ClusterOne	**0.55**	0.35	**0.42**	0.25	0.58	0.38	**1.19**
LCDA	**0.55**	0.16	0.26	**0.29**	0.33	0.31	1.16
PPI + BP							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.07	0.00	0.01	0.02	**0.68**	0.12	0.22
MCL	0.58	**0.47**	**0.52**	0.34	0.62	**0.45**	1.38
ClusterOne	0.61	0.41	0.49	0.31	0.63	0.44	1.37
LCDA	**0.72**	0.17	0.30	**0.35**	0.35	0.35	**1.41**
PPI + CC							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.10	0.01	0.02	0.03	**0.78**	0.15	0.28
MCL	0.57	**0.48**	**0.52**	0.34	0.65	**0.47**	1.39
ClusterOne	0.64	0.44	**0.52**	0.34	0.63	0.46	1.45
LCDA	**0.76**	0.20	0.31	**0.38**	0.34	0.36	**1.50**
PPI + All							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.08	0.01	0.02	0.03	**0.75**	0.15	0.26
MCL	0.51	**0.52**	**0.51**	0.39	0.63	**0.50**	1.40
ClusterOne	**0.74**	0.30	0.45	0.30	0.60	0.42	1.46
LCDA	0.66	0.20	0.30	**0.44**	0.31	0.37	**1.47**

Topological Algorithms Analysis

We compare our LCDA [60] algorithm that solely considers the topological interaction of the PPI network with other algorithms from the literature that perform in a similar manner. We select MCODE [118], MCL [119], and CLUSTERONE [120] for this comparison. We have used Cytoscape software [132] and exported the communities resulted from these methods. The input networks are extracted from Krogan dataset and divided based on GO functionalities. The assessments are described for all four

algorithms in Table 6.4 based on the metrics explained earlier in this section. As presented in the table, the performance of MCODE is considerably low compared to the other algorithms, even though we have set $\theta = 0.1$ to relax the condition for AS. MCL has overall the highest $Recall$, $Fmeasure$, and Acc, while our LCDA algorithm outperforms other algorithms with the highest $Precision$, Sn, and particularly $Composite\ Score$. The performance of ClusterOne algorithm is also high and relatively close to both MCL and and our algorithm LCDA. The $Composite\ Score$ is shown in Fig 6.2. The total height of each bar is the value of the $Composite\ Score$ and the larger scores are better. The figure describes how the three algorithms are competing for a higher performance rank and LCDA is outperform them.

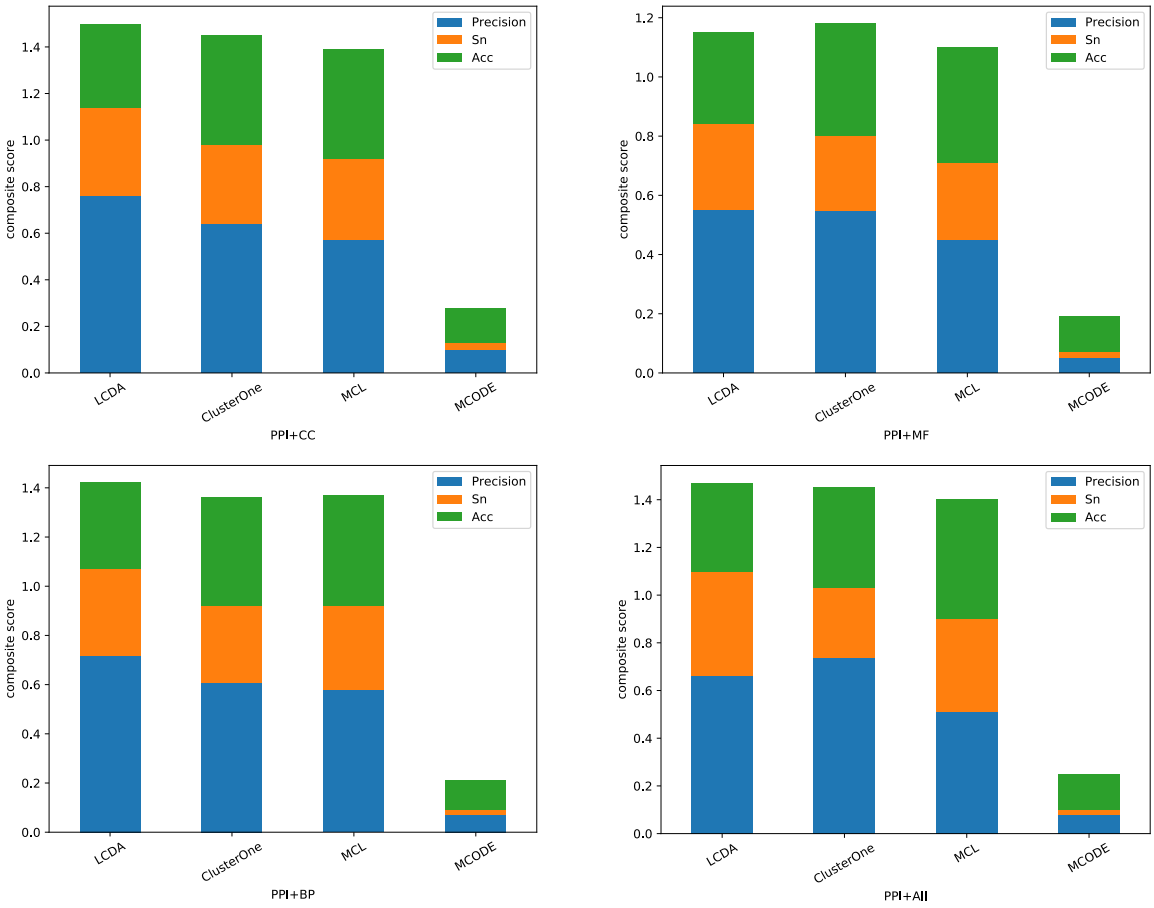

FIGURE 6.2: Composite score including $Precision$, Sn, and Acc.

Topological and Functional Algorithms Analysis

We implement and test our proposed algorithm for protein complex detection, LCDA-GO on all the networks extracted from Krogan dataset. The results are described in Table 6.5.

We choose MTGO to compare the results of LCDA-GO with since it also considers functionality as a parameter involved in the community detection and not as an in dependant process that could apply after community detection algorithm. We have exploited the MTGO software to run over the Krogan networks from Table 6.2, however, considering the large time complexity of this algorithm the final results could not converge by the time of writing this chapter. Therefore, we decided to rely on the experiments attached to their studies for this comparison. We choose only Sn, PPV, and Acc to compare the results due to the fact that they are independent from the threshold required

TABLE 6.5: Performance of LCDA-GO on *Saccharomyces Cerevisiae* from Krogan interaction datasets.

Network	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
PPI + MF	0.40	0.41	0.41	0.19	0.62	0.35	0.94
PPI + BP	0.72	0.17	0.30	0.35	0.35	0.35	1.41
PPI + CC	0.50	0.51	0.51	0.27	0.64	0.41	1.17
PPI + All	0.43	0.55	0.48	0.28	0.65	0.43	1.15

for *AS* score. The results are presented in Fig 6.3. As shown in this figure, even though MTGO has better *Sn* compared to LCDA-GO, *PPV* and *Acc* of LCDA-GO is larger. Overall, the two algorithms are competitive based on these assessments.

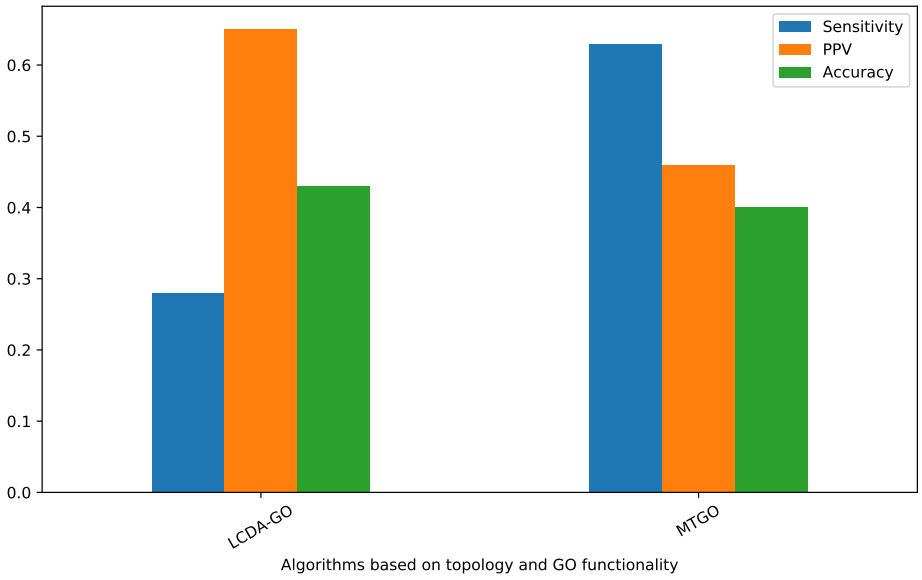

FIGURE 6.3: Comparing the results of LCDA-GO with MTGO on Krogan dataset.

Computational Complexity Analysis

Besides, the relatively close results from LCDA-GO and MTGO is the complexity of the two algorithms. Due to the locality of LCDA-GO, our algorithm enjoys from the loglinear time complexity while MTGO is a polynomial time algorithm. Our algorithm is more than 1400 times faster than MTGO when performing on Krogan dataset with 2674 nodes. The time complexity of LCDA-GO and MTGO is compared in Table 6.6.

TABLE 6.6: Complexity and run time of algorithms incorporating GO on Krogan network.

Algorithm	Time (sec)	Complexity
LCDA-GO	47.05	$O(nlog(n))$
MTGO	54000	$O(kn^3)$

6.5 Conclusion

Identifying protein complexes is an important step for biological knowledge discovery since several biological processes are accomplished in the formation of protein complexes. In this chapter, we propose a local community detection algorithm, LCDA-GO, for protein complexes by exploiting Gene Ontology (GO). LCDA-GO exploits networks' topological properties such as degree and shortest path in conjunction with protein's functional properties derived from GO databases. Our algorithm employs both topological and functional properties in local measures to perform on PPI networks in a local procedure.

We evaluate LCDA-GO and another variation of the algorithm called LCDA, the latter relying only on the topology of the network. Experimental results demonstrate their performance on real-world PPI networks from the Krogan dataset and their capabilities in finding protein complexes.

In addition, the promising performance of LCDA and LCDA-GO show the capability of our algorithms in successfully detecting protein complexes in PPI network with significantly lower time complexity than the state-of-the-art. LCDA-GO surpasses the state-of-the-art algorithms by performing on a log-linear time complexity, while recent algorithms such as MTGO run on polynomial time complexity.

One of the limitations of LCDA-GO is that it can only discover networks including one connected component. The algorithm relies on breadth-first search to discover the network, it thus could not converge if the network consists of more than one connected components. One solution to avoid this issue is to identify the connected components of the network before executing LCDA-GO and provide one node from each component as the input for the algorithm.

To extend our algorithm, we plan to evaluate LCDA-GO from functionality aspects. A GO term analysis could provide an evaluation on the significance of the functions within each community. Moreover, considering the various attributes utilized in PPI networks, we plan to analyze PPI networks from *attributed network* [133] prospect. We believe that the algorithm could expand for applications in the context of attributed networks.

Part II

Data Protection and AI Trustworthiness

Chapter 7

Privacy and Trustworthiness of AI

The huge volume, variety, and velocity of big data have empowered Machine Learning (ML) techniques and Artificial Intelligence (AI) systems. However, the vast portion of data used to train AI systems is sensitive information. Hence, any vulnerability has a potentially disastrous impact on privacy aspects and security issues. Nevertheless, the increased demands for high-quality AI from governments and companies require the utilization of big data in the systems. Several studies have highlighted the threats of big data on different platforms and the countermeasures to reduce the risks caused by attacks. The demand for AI in the market and yet the vulnerability of the data in the workflow has stimulated Standards Developing Organizations (SDOs) to set up Subcommittees (SCs) and initiate projects [10,134] with the mandate of providing standards and guidelines for big data and AI in order to help business sectors and market for a secure AI adoption. The Joint Technical Committee between the International Organization for Standardization and International Electrotechnical Commission (ISO/IEC JTC 1) [1] is a pioneer organization that is currently involved in developing standards on big data and AI.

This chapter provides a literature review on privacy and security issues of big data in AI systems. Our work departs from previous studies by discussing this issue using standards and guidelines developed by SDOs. Due to the worldwide importance of big data and AI in the market, we aim both research and standards to emphasize the opportunities where both frames can benefit from the outcomes of the other. The results are published in IEEE International Conference on Big Data (Big Data) as "**Privacy and Security of Big Data in AI Systems: a Research and Standards Perspective**" [3].

[1] `https://www.iso.org/isoiec-jtc-1.html`

7.1 Introduction

The huge volume of data generated by various sources, from connected devices to social media, termed as big data [2], is a valuable asset. The availability and widespread applications of big data [135] significantly impacts the growth of Machine Learning (ML) and Artificial Intelligence (AI) with the goals of increasing the efficiency and the accuracy of prediction and decision making and also minimizing their computational cost. Statistics depict the interest of the world market in AI systems that, only between 2018 and 2019, has increased by 154%, reached a $14.7 billion market size and will reach almost $37 billion by 2025 [136]. Stakeholders such as governments and industry sectors are attracted to benefit from AI to acquire insights from the data for customized services depend on customer's needs.

The integration of AI in various domains [5] significantly increases concerns regarding the privacy and security of data. The data that actuates AI includes various sensitive information, particularly individuals' information, including: images, speech, comments and posts on social media [6,7], financial transactions, and health record information. Feeding such data in AI systems, they become vulnerable to privacy and security attacks that are even significantly increased recently [8,9]. In a recent paper [8], the impact of adversarial attacks against AI medical systems is described such that an image of a benign melanocytic nevus is recognized as malignant with a high confidence score. A malicious attack on a face recognition system can reveal individuals images which are used to train the system [137]. By abusing a speech recognition system, an adversary can produce almost the same voice, however, transcribed the phrases [138]. Other attack techniques can cause potential safety hazards by effectively fooling the image classification system of an autonomous vehicle [139].

The demand for AI in the market and yet the vulnerability of the data in the workflow has stimulated Standards Developing Organizations (SDOs) to set up Subcommittees (SCs) and initiate projects [10,134] with the mandate of providing standards and guidelines for big data and AI in order to help business sectors and market for a secure AI adoption. SDOs develop technical standards and guidelines to address the needs and demands of particular adopters. Moreover, the standards play an important role in achieving interoperability and portability of complex ICT technologies and platforms. They can bring significant benefits to industry and consumers. The Joint Technical Committee between the International Organization for Standardization and International Electrotechnical Commission (ISO/IEC JTC 1)[2] is a pioneer organization that is currently involved in developing standards on big data and AI. The JTC 1 covers several domains concerning smart ICT and information technology including privacy, data protection and security of ICT technologies mainly under Subcommittee, ISO/IEC JTC1/SC 27 – "Information Security, Cybersecurity and Privacy Protection", and ISO/IEC JTC1/SC 42 – "Artificial Intelligence" that is dedicated to AI that is recently created and dedicated to AI and big data. Overall, the JTC 1 has already published more than 3k standards in different domains regarding smart ICT, among them are 188 for SC 27, 3 for SC 42 with 13 more standards under development for AI and big data. The other international level SDO is the International Telecommunication Union's Telecommunication Standardization Sector (ITU) that is focused on the AI in communication technologies. Furthermore, the Institute of Electrical and Electronics Engineers (IEEE) as the other international leading standard body, has also initiated projects which mostly concern the legal and ethical perspectives of AI [134]. In this chapter, our main target is the joint committee of ISO/IEC JTC 1 since it has already established a particular committee and various study and working groups in AI and big data related issues.

Different surveys in the literature have followed a particular perspective to tackle the privacy and security of machine learning and AI systems. Bae et al. [140] have considered the vulnerabilities of AI systems in the *white-box/black-box* scenarios, while Liu et al. [141] focused on learning techniques and classified attacks based on *training/testing* phases. Biggio et al. [9] proposed a four-dimensional model based on the *goal*, *knowledge*, *capability* and *attacking strategy* of the adversary. Additionally, in [142] the authors focused on the privacy and security issues of big data from another perspective based on the three main phases of big data analysis: *data preparation*, *data processing*, and *data analysis*. In an ongoing project by ISO/IEC JTC 1, the threats against the trustworthiness of AI systems are summarized and the characteristics of each has been reported [143]. We focus on the

[2] `https://www.iso.org/isoiec-jtc-1.html`

data violation threats in AI systems which are highlighted the most in the literature [9, 140, 141] and standardization [143].

7.2 Machine Learning (ML), Artificial Intelligence (AI)

In computer science, AI is associated with the accomplishments of tasks or problems by computers for which human intelligence is assumed to be required. AI is designed such that the input is the information acquired from the environment and takes actions to maximize success in achieving particular goals [144]. The most dominant way of achieving AI nowadays is by Machine Learning (ML) techniques which are build based on the concept of "without being explicitly programmed". In principle, ML consists of a set of algorithms and statistical models for computer systems to efficiently perform a particular task without relying on rule-based programming or human interaction. Developing the mathematical model is strongly dependant on the dataset, referred to as training data, which allows the program to gradually improve through the experiences and learning process from the data for predicting, detecting or making decisions [145]. A standard terminology of AI and Big data is also described in a standard document [146], an under development project from ISO/IEC JTC 1.

Machine learning techniques can be classified in different ways. In an underdevelopment standard [147] a set of ML approaches are defined as follows:

1. Supervised learning,
2. Unsupervised learning,
3. Semi-supervised learning,
4. Reinforcement learning,
5. Transfer learning.

Several techniques exist in each approach which are used based on the learning purpose and dataset. Regression, for instance is one of the well-known techniques used for prediction on labeled dataset. Clustering is another fundamental technique that is implied on unlabeled dataset for various applications such as recommendation of new options. However, clustering results shown to be highly influenced by the underlying data structure [25, 148]. Hence, a small change implied by an adversary can affect the results in the favour of the adversary [139].

7.3 Adversarial Model

We investigate privacy and security attacks of big data in AI systems that are modeled based on ML techniques. Each step in the workflow of the AI system can be the target of the specific attack(s). Hence, we use four phases in the AI overflow to identify the attacks based on the phase that an adversary penetrates to violate the system. The phases are illustrated in Fig. 7.1 on the defined AI workflow system. The first phase, *Training phase*, is the step where the trained data is fed into the ML model for the learning process. The data in this stage (labeled or unlabeled) is a significantly valuable source for the AI system that can be the aim of many attackers to violate the privacy and security [137, 149–151]. The next phase is the *Model phase* where the ML algorithm learns from the trained dataset and develops a model, which is the other valuable intellectual property of AI systems and hence is the target of various attacks [137, 152, 153]. The novel data is then fed into the trained model, named as *Apply phase*, where an adversary can penetrate the system and modify the results in his favor [139, 154, 155]. Finally, the valuable outcomes of the system, determined as the *Inference* phase, may host attacks that disclose sensitive information [156, 156, 157].

The security goals of the attacks are also investigated as the other feature. For this purpose, we consider the CIA triad [158], as the three pillars to cover the security of a system. They are summarized as follows:

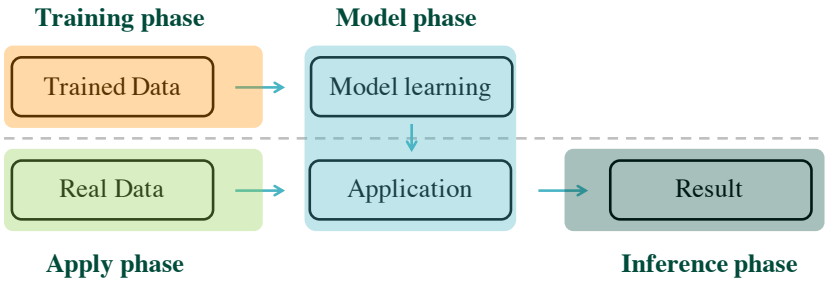

FIGURE 7.1: The workflow and different phases of AI systems developed based on ML algorithms.

- *Confidentiality* ensures the protection of sensitive information against misuse and unauthorized access. Hence, it roughly represents the privacy of a system.
- *Integrity* refers to the consistency and accuracy of data through the AI system workflow against unauthorized modification. An attack may modify the system towards misclassification, and yet does not affect the performance of the systems.
- *Availability* describes the system power to perform to achieve the expected purpose designed for the AI system with reliable outputs.

7.3.1 Standards Developing Organization (SDO)

TABLE 7.1: Identifying the phases where a particular attack penetrates the AI system.

Attack	**AI Workflow Phase**			
	Training	**Model**	**Apply**	**Inference**
Data Breach	✔	✔		✔
Bias in Data	✔			
Data Poisoning	✔			
Model extraction		✔		
Evasion			✔	

TABLE 7.2: Summary of the data privacy and security attacks in the AI workflow.

Attack	**Security Goal (CIA)**	**Attack Examples**	**Developed / Under development Standards**
Data Breach	Confidentiality	Re-identification [159] Risk of inference [160]	ISO/IEC CD 20547-4 [161], ISO/IEC PD TR 24028 [143]
Bias in Data	Integrity, Availability	Gender classification [149] Face recognition [162] Criminal legal system [163]	ISO/IEC NP TR 24027 [164], ISO/IEC PD TR 24028 [143]
Data Poisoning	Availability, Integrity	Self-driving car [151] Sentiment anlysis [165] Social media chatbot [166]	ISO/IEC PD TR 24028 [143]
Model Extraction	Confidentiality	Image recognition [152] Location data [167]	ISO/IEC PD TR 24028 [143]
Evasion	Integrity	Image classification [168] Spam emails [169] Self-driving car [139]	ISO/IEC PD TR 24028 [143]

7.4 Privacy and Security of Big Data in AI

In this section, we analyze the data privacy and security attacks concerning the defined characteristics (cf. Section 7.1). We describe the phase where the attack is imposed, the risks caused by the attack, and the real-world attack examples. Besides, an overview of the research papers and standards is conducted corresponding to each attack scenario. Table 7.1 represents, for each attack the phase(s) where a particular attack penetrates the AI system. Table 7.2 summarizes the attacks introduced in this section and lists the relevant standards where these attacks or the elements of mitigation strategies are described.

7.4.1 Data Breach

As a common privacy incident, a data breach is the disclosure of confidential or sensitive data in an unauthorized access. This type of attack has a long history [170] in privacy and security challenges of any systems and is not limited to AI. Nevertheless, AI has increased the quality of the insight gained from big data and therefore, new vulnerabilities against data and privacy breaches have raised by AI. The data breach may happen in different phases of AI workflow [171]: Training, model, and inference phases. Confidentiality which is roughly an equal to privacy is the target of the adversary providing this attack.

As an early example of data breach attacks is re-identification where attackers used another dataset from the public electoral rolls of the city of Cambridge [159] to identify medical records. Additionally, a study on mobile phone metadata revealed that unique identification of 95% of individuals from a population of 1.5 million people, requires only 4 approximate location and time data points [156]. Different methods were implied to mask the sensitive information of individuals within the datasets [170]. Nonetheless, the evolution of big data and computational techniques such as AI systems provided new opportunities to violate data privacy in the process.

7.4.2 Bias in Data

The decisions achieved by AI systems can reinforce injustice and discrimination [162] in shortening candidates list for credit approval, recruitment, and criminal legal system [163]. Even though bias is not directly recognized as the privacy and security issue of big data, it is entangled with data and thereby can significantly impact the accuracy and accountability of the results. Among different types of bias [164] identified in AI systems, we focus on those which are correlated to data: i) *Sample bias* describing an unbalanced representation of samples in training data, ii) *Algorithm bias* which refers to the systematic errors in the system, and iii) *Prejudicial bias* indicates the incorrect attitude upon an individual data. Other types such as *measurement bias* that results from poorly measuring the outcome, are out of the scope of this chapter. Bias is not a deliberate feature of AI systems, but rather the result of biases presented in the input data used to train the systems [172]. Hence, it targets the training phase and violates the *integrity* of an AI system.

Bias can target different attributes in decisions making including gender, race, age, national origin. In a project by MIT [149], known as Gender shade[3], the AI gender classification systems sold by giant technology companies (e.g., Microsoft, IBM, and Amazon) have been analyzed. The results of analysis in 2018, show a significant difference in the error rate of classifying darker-skinned female (up to 34.4%) in contrast to lighter-skinned males (0.8%). Some classification systems are considerably improved by 2019 [173] to reduce the error rate and yet the bias is not eliminated [174]. Bias is also found in a criminal legal system, Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) developed based on ML techniques to assess the sentencing and parole of convicted criminals. The purpose of COMPAS was to forecast the criminals who are most likely to re-offend [163]. However, the system has racial bias and tend to label black offenders almost twice higher risk than white offenders [175].

[3] http://gendershades.org/

7.4.3 Data Poisoning

Data poisoning [150] is one of the most widespread attacks developed based on the idea of learning with polluted data. Its disruptive effects in industrial applications have attracted experts of the standard technical committee to investigate on the countermeasures and defence techniques [143]. The attack happens by injecting adversarial training data during the learning to corrupt the model or to force a system towards producing false results [176]. Therefore, the attack works in two ways: i) a common adversarial type is to alter the boundaries of the classifier such that the model becomes useless. This way the attacker aims the availability of the system. ii) the other type, however, targets the integrity of the system by generating a *backdoor* such that the attacker can abuse the system in his favor.

In a particular study on injecting poisoned samples to a deep learning model, it is shown that only 50 polluted samples are enough to achieve a 90% attack success rate in the system [177] while the accuracy remains almost the same. Early examples of data poisoning attacks are the worm signature generation [178], and spam filtering [179]. In another real world scenario of classifying the street signs in the U.S., a backdoor attack lead to the misclassification of the stop sign as the speed limit sign [151]. In social media, the data poisoning attack on Microsoft's chatbot, Tay, created a bot who made offensive and racist statements [166]. The bot was shut down only 16 hours after its launch. Sentiment analysis [165], malware clustering and detection [180–182] are the other target domains of this attack.

7.4.4 Model Extraction

The trained model is a valuable intellectual property in ML systems due to i) the big data source that is been used to train the model, and ii) the parameters (e.g., weights, coefficients) which generated for the model based on its function (e.g., classification) [143, 183]. The adversary's aim from the model extraction might be to infer record(s) that is used to train the model, thus, violates the confidentiality of the system. Based on how sensitive the trained data is (e.g., medical record), the attack can cause a significant privacy breach by disclosing sensitive information [153]. A reverse-engineering of ML model can happen by observing the input and output pairs [184] or by sending queries and analyzing the responses [183], where Tramer et al. prove that sending only hundreds of queries is sufficient enough to clone the same system with almost 100% accuracy.

Many ML techniques (e.g., logistic regression, linear classifier, support vector machine, and neural network) [185, 186] are shown to be vulnerable to this type of attack [183] and yet the proposed defense mechanisms are not sufficient enough to protect the privacy and security of data. In a study by Fredrikson et al. [137, 152], the authors report that having access to a face recognition model, they reproduce almost 80% of an individual's image from the training dataset. In a similar yet more successful attack on face recognition [187], attackers infer samples with a 100% success rate. Other examples of membership inference attack is also observed in location data disclosure [167], machine translation and video captioning [153], and medical diagnosis [187].

7.4.5 Evasion

Evasion is a popular common attack in which the attacker's aim is to evade detection by fooling the systems towards misclassification [155]. It happens in the apply phase of the AI workflow, where the real data is implied on the trained model. The well-known example of evasion attacks is the adversarial samples [143]. They are malicious samples that are designed adding a few chosen bytes to the original sample [154]. Even though adversarial samples and poisoning data might look similar, they function differently. Considering a classifier, a data poisoning attack alters the classification boundary, however, adversarial samples modify the input samples to be classified in the wrong category. Hence, both lead to misclassification by targeting a different phase of AI workflow.

Adversarial samples are popular in comprising computer vision. In an experiment on autonomous vehicles [139], a couple of minor changes on the stop sign caused the learning model to misclassify the sign with a speed limit 45 sign. Even though for a human eyes such modifications does not affect the understanding of the street sign.

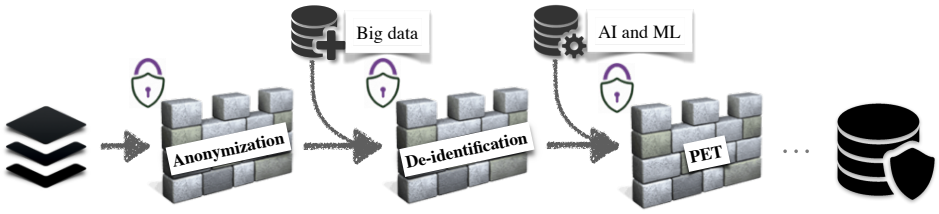

FIGURE 7.2: A overview of the evolution of defense techniques for AI and big data analysis.

7.5 Countermeasures and Privacy-preserving Solutions

This section describes an overview of the countermeasures and defense mechanisms of each particular attack mentioned in Section 7.4.

7.5.1 Data Breach

The data protection and privacy techniques evolved during the time based on the growth of big data and the complexity of data analysis techniques. The purpose of these mechanisms is to ensure the confidentiality of data used for data analysis. Overall, the privacy-preserving techniques of big data can be categorized in three classes: Anonymization, De-identification, and Privacy-enhancing Techniques (PET). The privacy concerns of data is not a recent issue, started from data analysis on medial datasets in 1998, when the researcher find out that the anonymization is not sufficient solely to protect data privacy [170]. The sensitive data disclosure reports [156, 159] represent the deficiency of anonymization, where replacing clear identifier was enough solely to ensure the privacy and security of the data. Hence, the second level of mechanisms developed by *k-anonymity* [170] family, including *l-diversity* and *t-closeness* [142]. These techniques are suitable to mask sensitive information such as location-based data [188] to guarantee that the identity of records is not distinguishable in a dataset. The emergence of AI and ML techniques along with the increased complexity of big data, the conventional de-identification methods become obsolete [157]. Hence, PET was developed for privacy-preserving data analysis in various domains such as e-health [189, 190]. Fig. 7.2 describes these techniques according to the evolution of privacy-preserving techniques.

The next generation of the privacy-preserving approach is focused on the concept of *sending the code to the data*. The OPen ALgorithms (OPAL) project [191] has combined different mechanisms such as access-control protocols, aggregation schemes and develop a platform that allows third-parties (e.g., researchers) to submit algorithms that will be trained on data. The privacy of individuals, however, is guaranteed while data is being analyzed. Furthermore, Google's DeepMind has also developed a *verifiable data audit* which ensures that any interaction with health records data is recorded and accessible to mitigate the risk of foul play.

7.5.2 Bias in Data

To identify different types of bias several metrics are introduced in the literature [172] including difference in means, difference in residuals, equal opportunity, disparate impact, and normalized mutual information. Moreover, benefiting the metrics, approaches to mitigate AI bias are developed such as optimized prepossessing, reject option classification, learning fair representations, and adversarial debiasing [192]. Besides, a set of toolboxes are designed which are accumulated the identification metrics along with the mitigation approaches together as a framework for different ML algorithms. The purpose is to diagnose and remove AI biases if exists in the system. The available toolboxes are Lime, FairML [193], Google What-If and IBM Bias Assessment Toolkit [194] which is mostly used for face detection systems.

7.5.3 Data Poisoning

The feasibility of data poisoning attacks on ML algorithms such as Support Vector Machine (SVM) classifier is studied [195]. One common approach to detect the poisoned data is to identify the outlier (i.e., anomaly detection) since the injected data is expected to follow a different data distribution. Paudice et al. [196] developed their defense model against data poisoning based on anomaly detection. However, poisoned samples can evade anomaly detection if the adversary knows the data distribution. Hence, advanced techniques are required to defeat the attack. In [197], a method is proposed to perturb the incoming input and observe the randomness of the outcome. A low variance in the predicted classes represents malicious samples. Nelson et al. [198] proposed a technique to recognize and remove the poisoned data in the training dataset by separating the new joined input and calculate the accuracy of the model on them.

7.5.4 Model Extraction

Juuti et al. [199] proposed a method to detect model extraction attack by analyzing the distribution of consecutive API queries and compare it with benign behavior. One possible defense technique against model extraction is by training multiple models using different partitions of training data to each model. The techniques are proposed by Papernot et al. known as PATE [200]. Another approach to protect the learning model is to limit the information regarding the probability score of the model and degrade the success rate by misleading the adversary [201].

7.5.5 Evasion

Adversarial samples, as the most common evasion attacks, leas to misclassification only by small perturbations in the original inputs. Hence, a potential defense mechanism is to ensure that a small modification in the input cannot change the result significantly. Adversarial training is based on this technique to train the model based on the adversarial samples, however, with true labels such that it can avoid the noise [202]. In a similar approach by Deepfool [168] the ideas is to compute the perturbations which fool the classifier and thus quantify the robustness of the classifier. In another approach, the goal is to detect the adversarial samples from the original ones and therefore remove them from the dataset [203].

7.6 Conclusion

The huge volume, variety, and velocity of big data have empowered Machine Learning (ML) techniques and Artificial Intelligence (AI) systems. As privacy and security threats evolve, so too will the technology need to adapt – as well as the rules and regulations that govern the use of such technologies. The two perspectives of the research outcomes and standards development are considered in this study. We focus on challenges and threats of big data in the AI workflow by providing a review of the recent research literature, standard documents, and ongoing projects on this topic. Several projects are initiated by SDOs to investigate different aspects of big data privacy aspects and security issues. Even though most of the standards mentioned in this study are ongoing projects, they are expected to be published in the near future. One of the advantages standards can bring into research is a more coherent terminology, which is defined once and used later in subsequent projects. In contrast, researchers often use different terminologies for the same or similar concepts. Besides, according to the rapid growth of AI, developed road maps in standards can provide insights according to the demands and requirements of the market. Hence, it may provide opportunities for new research activities to address line with market needs.

Chapter 8

AI Conformity Assessment

AI has been deployed into many applications where trust is imperative. It is shown that how AI is capable to be integrated in our lives and we see both the promises and the harms of the AI-based technologies. The primary need to work on a certification program is paramount to allow these technologies operate with proper transparency and control. One main requirement towards developing an efficient guideline and standard is to invite different stakeholders to get involved and participate in the process. The purpose of this simulation is to develop a useful framework where different stakeholders can participate in the development of a certain standard and guideline by providing feedback based on their own benefits and use-case.

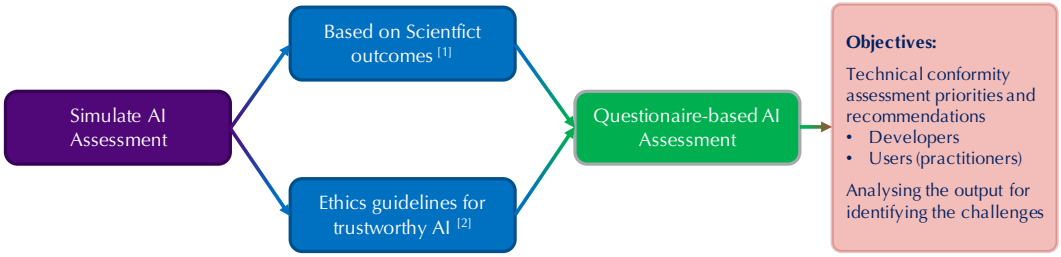

FIGURE 8.1: An overview on simulation of the AI Assessment.

8.1 AI Conformity Assessment

ISO certificates are good examples to explain what a standard certificate is. An ISO certificate (e.g., ISO 9001, ISO 27001) certifies that a system, process, or service has met all the standardization and quality assurance requirements. ISO certificates exist in various industry areas, including Smart ICT domains such as quality management systems and information security management systems. When an organization bills itself a particular ISO certification, it means that that organization met the requirement designed under that particular ISO certification. Hence, the organization follows a verified level of quality principles, which in turn brings many business benefits. With all ongoing projects in ISO for developing AI standards (ISO/IEC 42001 – AI management system), there has not been a project regarding the certification for AI systems broadly utilized by different stakeholders. The need for guidelines and principles for safe, transparent, and trustworthy AI systems should be a priority for multiple standardization development organizations. IEEE has taken the first step towards this matter by developing the Ethics Certification Program for Autonomous and Intelligent Systems (ECPAIS) for the purpose of developing critical certification criteria for responsible innovation and delivery of autonomous and intelligent systems (A/IS).

AI has been deployed into many applications where trust is imperative. The primary need to work on a certification program is paramount to allow these technologies operate with proper transparency and control. One main requirement towards developing an efficient guideline and standard is to invite different stakeholders to get involved and participate in the process. The purpose of this simulation is to develop a useful framework where different stakeholders can participate in the development of standards. Figure 8.1 shows a summary of the objectives of this simulation.

8.2 The Structure and Organization

The proposed assessment consists of four clauses to assure a minimum value of ethics, trust, and privacy in an AI system. The clauses are defined based on HLEGAI report [1] as follows:

8.2.1 Robustness and Safety

A crucial component of achieving Trustworthy AI is technical robustness, which is closely linked to the prevention of harm. Technical robustness requires that AI systems be developed with a preventative approach to risks and how they reliably behave as intended while minimizing unintentional and unexpected harm and preventing unacceptable harm. This should also apply to potential changes in their operating environment or the presence of other agents (human and artificial) that may interact with the system in an adversarial manner. Besides, the physical and mental integrity of humans should be ensured.

[1]High-Level Expert Group on Artificial Intelligence (HLEGAI), Ethics Guidelines for Trustworthy AI, (2019)

8.2.2 Explicability

Explicability is crucial for building and maintaining users' trust in AI systems. This means that processes need to be transparent, the capabilities and purpose of AI systems openly communicated, and decisions – to the extent possible – explainable to those directly and indirectly affected. Without such information, a decision cannot be duly contested. An explanation as to why a model has generated a particular output or decision (and what combination of input factors contributed to that) is not always possible. These cases are referred to as 'black-box' algorithms and require special attention. In those circumstances, other explicability measures (e.g., traceability, auditability, and transparent communication on system capabilities) may be required, provided that the system as a whole respects fundamental rights. The degree to which explicability is needed is highly dependent on the context and the severity of the consequences if that output is erroneous or otherwise inaccurate.

8.2.3 Non-Discrimination and Fairness

The development, deployment, and use of AI systems must be fair. While we acknowledge that there are many different interpretations of fairness, we believe that fairness has both a substantive and a procedural dimension. The substantive dimension implies a commitment to: ensuring equal and just distribution of both benefits and costs, and ensuring that individuals and groups are free from unfair bias, discrimination and stigmatisation.If unfair biases can be avoided, AI systems could even increasesocietal fairness. Equal opportunity in terms of access to education, goods, services and technology should also be fostered. Moreover, the use of AI systems should never lead to people being deceived or unjustifiably impaired in their freedom of choice.

8.2.4 Privacy and Data Governance

AI systems must guarantee privacy and data protection throughout a system's entire life-cycle. This includes the information initially provided by the user, as well as the information generated about the user over the course of their interaction with the system (e.g. outputs that the AI system generated for specific users or how users responded to particular recommendations). Digital records of human behaviour may allow AI systems to infer not only individuals' preferences, but also their sexual orientation, age, gender, religious or political views. To allow individuals to trust the data gathering process, it must be ensured that data collected about them will not be used to unlawfully or unfairly discriminate against them. Fig. 8.2 shows the structure of the simulated AI Certificate.

8.3 Methodology

The simulation is designed in a questionnaire format. The educational purpose of the framework is for the stakeholders to gain knowledge on the requirements and how to implement them. The primary goal is to involve them in an interactive way to develop and improve a practical certificate. This simulation's inquiry can provide valuable information for a gap analysis to identify possible new items as Luxembourg national contribution to support the potential AI Certificate and standards.

We implement the AI conformity assessment by asking two particular roles to get involved in this simulation, each from a different perspective:

- AI developer
- Internal examiner

The AI developer's primary concern is privacy and trustworthiness of the system being developed. The AI developer has a technical background on AI. We asked them to prioritize each question considering their AI developments by scoring (considering 1 for the most important one and 5 as the least important) questions of each section. This represents the priorities of the developer with respect to the developed AI systems. Finally, we also asked for in each the related factor(s) that is missing from the list and yet important based on the AI domain and system of use.

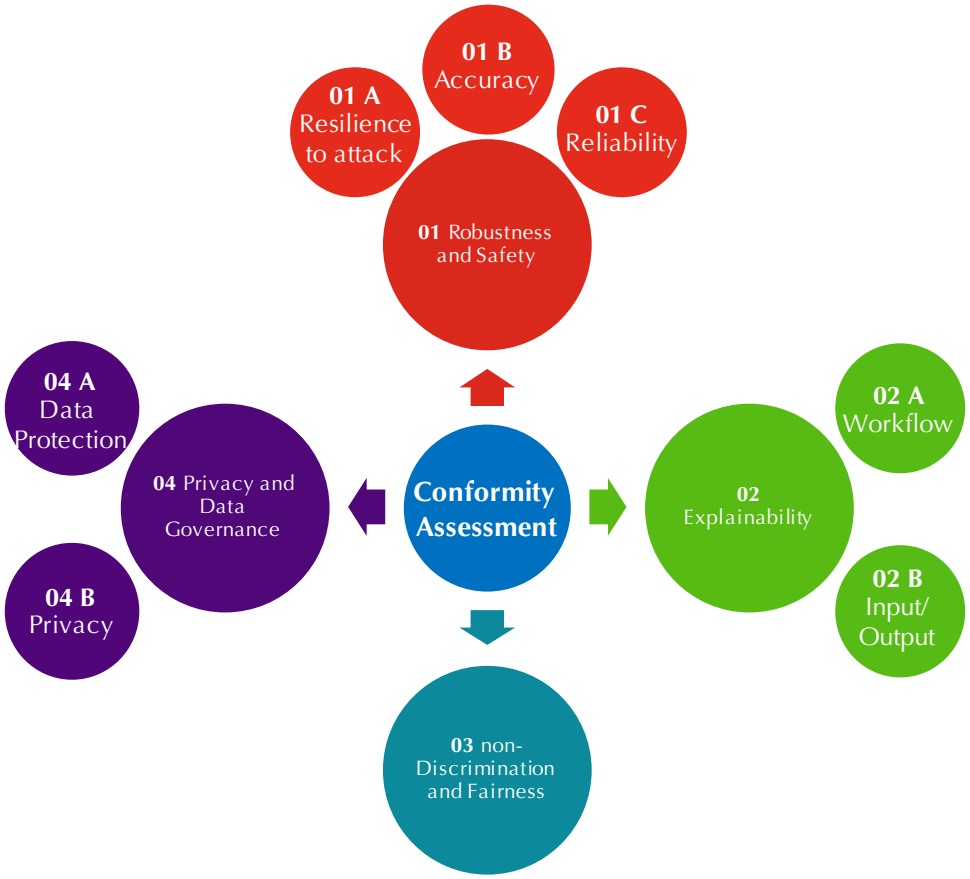

FIGURE 8.2: The main structure of the Simulated AI conformity assessment.

The internal examiner's focus is data privacy and conformity of the AI systems that is using individuals data and creating models based on this data. In this case, we asked them to fill in the response column of the questionnaire by replying the questions (21 Y/N questions). The questions are organized in four clauses of the AI Conformity Assessment shown in Figure 1.

8.4 Implementation and Results

We provide a dry-run test with our team at SnT. A group of 10 scientist and engineers participated in this simulation. Each person was involved in AI application research/development and was asked to answer the questionnaire considering the application. In general, 5 different applications of AI were considered in this test including fintech, generating heuristics, UAVs, constraint solver, and medical imaging. The results of the first dry-run of the conformity assessment is shown in Fig. 8.3.

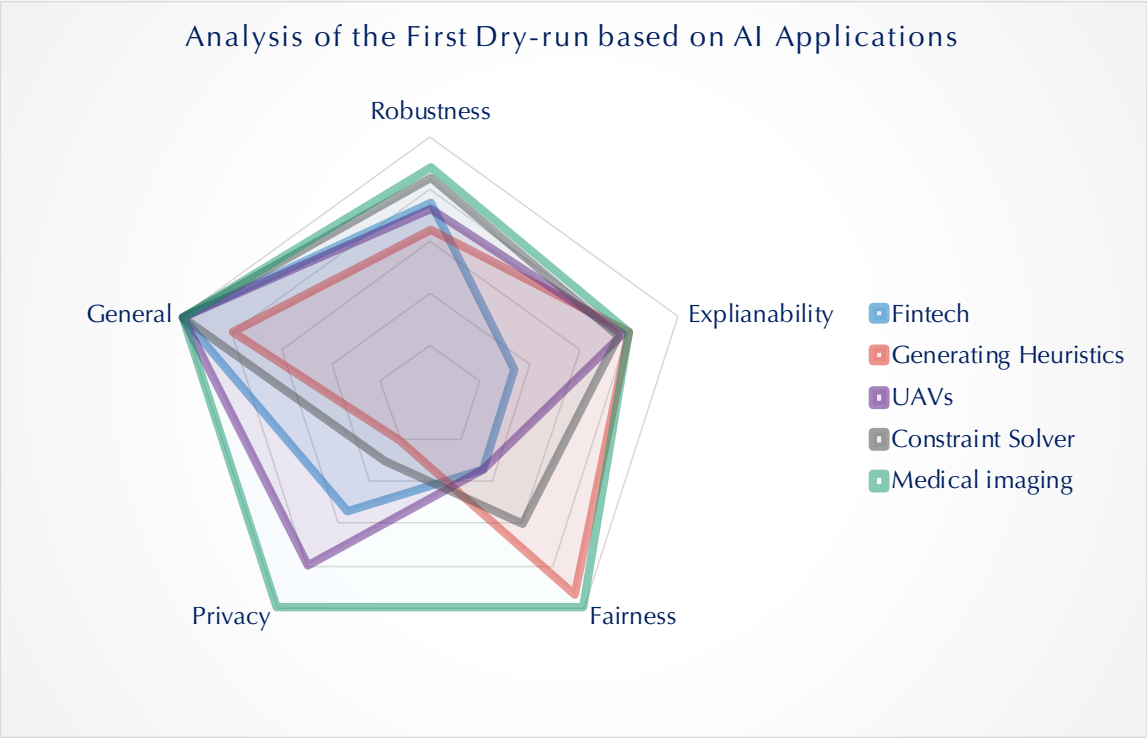

FIGURE 8.3: The results of the first dry-test of the AI Conformity Assessment at University of Luxembourg.

Part III

Conclusion Remarks

Chapter 9

Concluding Remarks and Future Work

This dissertation tackled several challenges in AI and Complex Networks. In summary, we defined and developed local network analysis algorithms, as well as described the challenges of AI trustworthiness from a unique perspective by combining standards and research. Accordingly, the main conclusions drawn from the work carried out in this thesis, as well as the potential extensions and applications of the thesis findings, are described in the following sections.

9.1 Main Conclusions

The first Part of the thesis is devoted to complex network analysis, and Chapters 2-6 are the contributions we made on this topic.

First, we elaborated on the network construction from relational data and designed an approach in Chapter 2 to describe potential network topologies driven from a real-world relational dataset of research collaboration. We defined a Linkage Threshold (LT) parameter that is formulated over the number of collaborators in teams and their contribution percentage. As results, we presented different network layers each presenting an organizational perspective from the collaboration data. We performed network analysis with metrics such as clustering coefficient, closeness and betweenness centrality, and illustrated their impact on the different network layers. The network analysis of the generated network layers reveals different behavior in each layer. The metrics results are then used as an important input to visualize the generated graphs. We conclude that the LT has a significant impact on network properties and should be chosen cautiously.

In Chapter 3 3, we revisit the network construction challenge by first improving our method to incorporate more data features and then developing network analysis to analyze the impact of the constructed layers on community detection results. Our findings show that the quality of detected communities can differ significantly when increasing the links and connections of the network.

We discovered a gap in local approaches to community detection as a result of our research on community detection. In Chapter 4 4, we emphasized the lack of a concrete taxonomy for local community detection algorithms, in contrast to the numerous studies and taxonomies for global community detection algorithms. We proposed a locality exploration scheme (LES) as a solution and investigated the concept of locality at each stage of the existing community detection algorithms. We demonstrated the applicability of our scheme by reviewing some algorithms based on our proposed scheme. Our findings can be used to guide the selection of the most relevant functions when developing community detection algorithms and deploying them on networks.

In Chapter 5 5, we proposed a new local community detection algorithm (LCDA) to address the drawbacks of traditional global algorithms as well as the limitations of previous local algorithms. LCDA was created using a set of local principles to emphasize the algorithm's locality. LCDA relies solely on neighborhood local information to identify all network communities. The algorithm is designed with a more restricted level of locality than current local algorithms and has a logarithmic order of computational time complexity. Extensive experiments were carried out to evaluate the performance and efficiency of our algorithm both on real and artificial networks. The outcomes show that LCDA performs better in networks with weak community structures than algorithms that benefit from the network's global information.

We demonstrate an application of LCDA on biological networks in Chapter 6 6 by extending LCDA to a new variation that includes network functionality as well. In this regard, we created LCDA-GO, which combines network topology and functionality, and applied it to protein-protein interaction (PPI) networks to detect protein communities that collaborate in the cell. Our LCDA-GO algorithm identifies the community of each protein based solely on topological and functional knowledge (GO) acquired from the PPI network's local neighbor proteins. Experiment results on the yeast PPI network demonstrated that our algorithm outperforms state-of-the-art approaches in assessment based on Precision, Sensitivity, and, in particular, Composite Score in the majority of cases. Aside from the high quality of the results, one major advantage of LCDA-GO is its low computational time complexity when compared to previous algorithms of a similar type.

The second Part of the thesis, Chapter 7-8, describes the contributions made for AI trustworthiness including standards in AI and big data.

In Chapter 7 7, we defined an AI workflow and discussed the threats and attacks that aim to undermine AI's trustworthiness, robustness, and privacy. We analyzed several threats by defining the threat's target and security goal. To formalize our survey, we consider both standardization and research resources when developing the analysis. We introduce the most recent projects initiated by the International Standardization Organization (ISO) by the time and connect them to existing AI challenges and threats that could be used for further gap analysis.

Finally, in Chapter 8 8, we developed a questionnaire-based AI Conformity Assessment to bridge

the gap between the most recent research findings and the concerns about AI regulation and certification. We proposed a pilot simulation to assess the level of awareness of people working with AI, as well as to examine AI in different environments to identify gaps. We conducted a dry-run test with our SnT team and reported the results.

9.2 Future Work

The work presented in this thesis has the potential to be expanded in a variety of ways. We propose some possible extensions to the current work in this section.

1. Some challenges need to be further addressed in the community detection algorithm as listed below:
   - Firstly, re-build the LCDA algorithm on objective oriented or optimization model
   - Secondly, providing the solid proof on the properties of the algorithm (e.g., termination)
2. Investigating on the dynamicity aspects of the algorithm by employing it on temporal networks.
3. The privacy-preserving of the algorithm is also yet to be discussed. The locality base of LCDA initiates privacy preserving features on the algorithm. However, additional tests and analysis are required to be employed.
4. Regarding the AI Assessment, the framework could be further extended by converting existing measures and algorithms to an AI trustworthy tool that can both upskilling the employees and also to asses AI in different domains.

Bibliography

[1] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 2009.

[2] D. Laney, "3d data management: Controlling data volume, velocity and variety," *META group research note*, vol. 6, no. 70, p. 1, 2001.

[3] S. Dilmaghani, M. R. Brust, G. Danoy, N. Cassagnes, J. Pecero, and P. Bouvry, "Privacy and security of big data in ai systems: a research and standards perspective," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 5737–5743.

[4] A.-L. Barabási, "Network science book," *Network Science*, vol. 625, 2014.

[5] A. K. Keith Kirkpatrick, "Artificial intelligence use cases," Tractica, Tech. Rep., 2018.

[6] J. Chen, A. R. Kiremire, M. R. Brust, and V. V. Phoha, "Modeling online social network users' profile attribute disclosure behavior from a game theoretic perspective," *Computer Communications*, vol. 49, 2014.

[7] J. Chen, M. R. Brust, A. R. Kiremire, and V. V. Phoha, "Modeling privacy settings of an online social network from a game-theoretical perspective," in *IEEE Int. Conference on Collaborative Computing: Networking, Applications and Worksharing*, Oct 2013, pp. 213–220.

[8] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019.

[9] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 4, pp. 984–996, 2013.

[10] CEN-CENELEC. (2019) Artificial Intelligence, Blockchain and Distributed Ledger Technologies. [Online]. Available: https://www.cencenelec.eu/standards/Topics/ArtificialIntelligence\/Pages/default.aspx

[11] S. Shirinivas, S. Vetrivel, and N. Elango, "Applications of graph theory in computer science an overview," *International Journal of Engineering Science and Technology*, vol. 2, no. 9, pp. 4610–4621, 2010.

[12] D. S. Bassett, P. Zurn, and J. I. Gold, "On the nature and use of models in network neuroscience," *Nature Reviews Neuroscience*, p. 1, 2018.

[13] A. Karduni, A. Kermanshah, and S. Derrible, "A protocol to convert spatial polyline data to network formats and applications to world urban road networks," *Scientific data*, vol. 3, p. 160046, 2016.

[14] S. Hong, B. C. Coutinho, A. Dey, A.-L. Barabási, M. Vogelsberger, L. Hernquist, and K. Gebhardt, "Discriminating topology in galaxy distributions using network analysis," *Monthly Notices of the Royal Astronomical Society*, vol. 459, no. 3, pp. 2690–2700, 2016.

[15] S. P. Fraiberger, R. Sinatra, M. Resch, C. Riedl, and A.-L. Barabási, "Quantifying reputation and success in art," *Science*, 2018.

[16] C. T. Butts, "Revisiting the foundations of network analysis," *science*, vol. 325, no. 5939, pp. 414–416, 2009.

[17] M. E. Newman, "Scientific collaboration networks. i. network construction and fundamental results," *Physical review E*, vol. 64, no. 1, p. 016131, 2001.

[18] S. Dilmaghani, A. Piyatumrong, G. Danoy, P. Bouvry, and M. R. Brust, "Innovation networks from inter-organizational research collaborations," in *Heuristics for Optimization and Learning*. Springer, 2021, pp. 361–375.

[19] J. G. Davis, J. K. Panford, and J. B. Hayfron-Acquah, "Big and connected data analysis with graph and relational databases using collaborative filtering technique," *Int. Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 12, 2017.

[20] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics*.

[21] J. Yang and X.-D. Zhang, "Predicting missing links in complex networks based on common neighbors and distance," *Scientific reports*, vol. 6, p. 38208, 2016.

[22] L. Pan, T. Zhou, L. Lü, and C.-K. Hu, "Predicting missing links and identifying spurious links via likelihood analysis," *Scientific reports*, vol. 6, p. 22955, 2016.

[23] Z. Sha, Y. Huang, J. S. Fu, M. Wang, Y. Fu, N. Contractor, and W. Chen, "A network-based approach to modeling and predicting product coconsideration relations," *Complexity*, vol. 2018, 2018.

[24] M. R. Brust, H. Frey, and S. Rothkugel, "Adaptive multi-hop clustering in mobile networks," in *Proc. of the 4th Int. Conf. on mobile technology, applications*, 2007.

[25] S. Dilmaghani, M. R. Brust, A. Piyatumrong, G. Danoy, and P. Bouvry, "Link definition ameliorating community detection in collaboration networks," *Frontiers in Big Data*, vol. 2, p. 22, 2019.

[26] G. Casiraghi, V. Nanumyan, I. Scholtes, and F. Schweitzer, "From relational data to graphs: Inferring significant links using generalized hypergeometric ensembles," in *Int. Conf. on Social Informatics*. Springer, 2017, pp. 111–120.

[27] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 981–990.

[28] A. Schein, J. Paisley, D. M. Blei, and H. Wallach, "Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts," in *ACM Int. Conf. on Knowledge Discovery and Data Mining*, 2015.

[29] M. Akbas, M. Brust, and D. Turgut, "Social network generation and role determination based on smartphone data," in *IEEE International Conference on Computer Communications (INFOCOM) Student Workshop*, 2012.

[30] M. I. Akbas, M. R. Brust, C. H. C. Ribeiro, and D. Turgut, "Deployment and mobility for animal social life monitoring based on preferential attachment," in *IEEE Conference on Local Computer Networks*, Oct 2011, pp. 484–491.

[31] ——, "fapebook - animal social life monitoring with wireless sensor and actor networks," in *IEEE Global Telecommunications Conference - GLOBECOM*, Dec 2011, pp. 1–5.

[32] M. I. Akbas, M. R. Brust, D. Turgut, and C. H. Ribeiro, "A preferential attachment model for primate social networks," *Computer Networks*, vol. 76, pp. 207 – 226, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614003983

[33] X. Ouvrard, J.-M. L. Goff, and S. Marchand-Maillet, "Networks of collaborations: Hypergraph modeling and visualisation," *arXiv preprint arXiv:1707.00115*, 2017.

[34] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.

[35] U. Brandes, "A faster algorithm for betweenness centrality," *J. of math. sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[36] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, and J. Kertesz, "Generalizations of the clustering coefficient to weighted complex networks," *Physical Review E*, vol. 75, no. 2, p. 027105, 2007.

[37] J. Huang, Z. Zhuang, J. Li, and C. L. Giles, "Collaboration over time: characterizing and modeling network evolution," in *Proc. of the inter. conf. on web search and data mining*. ACM, 2008, pp. 107–116.

[38] M. Jamali and H. Abolhassani, "Different aspects of social network analysis," in *2006 IEEE/WIC/ACM Inter. Conf. on Web Intelligence (WI'06)*. IEEE, 2006, pp. 66–72.

[39] J. C. Long, F. C. Cunningham, P. Carswell, and J. Braithwaite, "Patterns of collaboration in complex networks," *BMC Health Services Research*, vol. 14, no. 1, p. 225, 2014.

[40] S. Bloemheuvel, M. Atzmueller, M. Postma, M. Atzmueller, and W. Duivesteijn, "Evolution of contacts and communities in social interaction networks of face-to-face proximity," 2018.

[41] T. Chakraborty, N. Ganguly, and A. Mukherjee, "An author is known by the context she keeps: significance of network motifs in scientific collaborations," *Social Network Analysis and Mining*, vol. 5, no. 1, p. 16, 2015.

[42] G. Pan, W. Zhang, Z. Wu, and S. Li, "Online community detection for large complex networks," *PloS one*, vol. 9, no. 7, p. e102799, 2014.

[43] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 3, pp. 115–135, 2016.

[44] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Citeseer, Tech. Rep., 2002.

[45] M. E. Newman, "Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality," *Physical review E*, vol. 64, no. 1, p. 016132, 2001.

[46] A. Sharma and S. D. Bhavani, "A network formation model for collaboration networks," in *International Conference on Distributed Computing and Internet Technology*. Springer, 2019, pp. 279–294.

[47] I. Scholtes, "When is a network a network?: Multi-order graphical model selection in pathways and temporal networks," in *Proc. of the ACM SIGKDD*. ACM, 2017, pp. 1037–1046.

[48] K. Faust, "7. very local structure in social networks," *Sociological Methodology*, vol. 37, no. 1, pp. 209–256, 2007.

[49] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, p. 440, 1998.

[50] M. E. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical review E*, vol. 64, no. 2, p. 026118, 2001.

[51] M. Newman, "Network structure from rich but noisy data," *Nature Physics*, vol. 14, no. 6, p. 542, 2018.

[52] S. Dilmaghani, M. R. Brust, G. Danoy, and P. Bouvry, "Community detection in complex networks: A survey on local approaches," in *Intelligent Information and Database Systems*. Cham: Springer International Publishing, 2021, pp. 757–767.

[53] M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks," *Notices of the AMS*, vol. 56, no. 9, 2009.

[54] S. E. Schaeffer, "Graph clustering," *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007.

[55] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, 2010.

[56] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, 2004.

[57] J. P. Bagrow and E. M. Bollt, "Local method for detecting communities," *Physical Review E*, vol. 72, no. 4, 2005.

[58] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "Demon: a local-first discovery method for overlapping communities," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, 2012.

[59] S. Muff, F. Rao, and A. Caflisch, "Local modularity measure for network clusterizations," *Physical Review E*, vol. 72, no. 5, p. 056107, 2005.

[60] S. Dilmaghani, M. R. Brust, G. Danoy, and P. Bouvry, "Local community detection algorithm with self-defining source nodes," in *International Conference on Complex Networks and Their Applications*. Springer, 2020, pp. 200–210.

[61] J. Cheng, X. Su, H. Yang, L. Li, J. Zhang, S. Zhao, and X. Chen, "Neighbor similarity based agglomerative method for community detection in networks," *Complexity*, vol. 2019, 2019.

[62] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, 2018.

[63] K. Guo, L. He, Y. Chen, W. Guo, and J. Zheng, "A local community detection algorithm based on internal force between nodes," *Applied Intelligence*, vol. 50, no. 2, 2020.

[64] Y. Li, K. He, D. Bindel, and J. E. Hopcroft, "Uncovering the small community structure in large networks: A local spectral approach," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 658–668.

[65] N. Gulbahce and S. Lehmann, "The art of community detection," *BioEssays*, vol. 30, no. 10, pp. 934–938, 2008.

[66] D. Angluin, "Local and global properties in networks of processors," in *Proceedings of the twelfth annual ACM symposium on Theory of computing*, 1980, pp. 82–93.

[67] M. R. Brust and S. Rothkugel, "A taxonomic approach to topology control in ad hoc and wireless networks," in *International Conference on Networking (ICN'07)*, 2007.

[68] M. Stein, M. Fischer, I. Schweizer, and M. Mühlhäuser, "A classification of locality in network research," *ACM Computing Surveys (CSUR)*, vol. 50, no. 4, 2017.

[69] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *Acm computing surveys (csur)*, vol. 45, no. 4, pp. 1–35, 2013.

[70] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova, "Community detection in large-scale networks: a survey and empirical evaluation," *Wiley Reviews: Computational Statistics*, vol. 6, no. 6, 2014.

[71] Z. Yang, R. Algesheimer, and C. J. Tessone, "A comparative analysis of community detection algorithms on artificial networks," *Scientific reports*, vol. 6, 2016.

[72] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 4, no. 5, pp. 512–546, 2011.

[73] Y. Chen, P. Zhao, P. Li, K. Zhang, and J. Zhang, "Finding communities by their centers," *Scientific reports*, vol. 6, 2016.

[74] X. Wang, G. Liu, J. Li, and J. P. Nees, "Locating structural centers: A density-based clustering method for community detection," *PloS one*, vol. 12, no. 1, 2017.

[75] J. M. Hernández and P. Van Mieghem, "Classification of graph metrics," *Delft University of Technology: Mekelweg, The Netherlands*, pp. 1–20, 2011.

[76] S. Li, J. Huang, Z. Zhang, J. Liu, T. Huang, and H. Chen, "Similarity-based future common neighbors model for link prediction in complex networks," *Scientific reports*, vol. 8, no. 1, 2018.

[77] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[78] X. Wang and G. Sukthankar, "Link prediction in heterogeneous collaboration networks," in *Social network analysis-community detection and evolution*. Springer, 2014, pp. 165–192.

[79] C. H. Comin and L. da Fontoura Costa, "Identifying the starting point of a spreading process in complex networks," *Physical Review E*, vol. 84, no. 5, 2011.

[80] Q. Chen and T.-T. Wu, "A method for local community detection by finding maximal-degree nodes," in *2010 International Conference on Machine Learning and Cybernetics*, vol. 1. IEEE, 2010, pp. 8–13.

[81] R. Shang, W. Zhang, L. Jiao, R. Stolkin, and Y. Xue, "A community integration strategy based on an improved modularity density increment for large-scale networks," *Physica A: Statistical Mechanics and its Applications*, vol. 469, pp. 471–485, 2017.

[82] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *Journal of physics*, vol. 11, no. 3, 2009.

[83] H. Mahyar, R. Hasheminezhad, E. Ghalebi, A. Nazemian, R. Grosu, A. Movaghar, and H. R. Rabiee, "Identifying central nodes for information flow in social networks using compressive sensing," *Social Network Analysis and Mining*, vol. 8, no. 1, p. 33, 2018.

[84] Z. Lin, X. Zheng, N. Xin, and D. Chen, "Ck-lpa: Efficient community detection algorithm based on label propagation with community kernel," *Physica A: Statistical Mechanics and its Applications*, vol. 416, pp. 386–399, 2014.

[85] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[86] Y. Pan, D.-H. Li, J.-G. Liu, and J.-Z. Liang, "Detecting community structure in complex networks via node similarity," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 14, pp. 2849–2857, 2010.

[87] A. Clauset, "Finding local community structure in networks," *Physical review E*, vol. 72, no. 2, 2005.

[88] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," in *2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI'06)*, 2006.

[89] Q. Chen, T.-T. Wu, and M. Fang, "Detecting local community structures in complex networks based on local degree central nodes," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 3, pp. 529–537, 2013.

[90] C. Stegehuis, R. Van Der Hofstad, and J. S. Van Leeuwaarden, "Epidemic spreading on complex networks with community structures," *Scientific reports*, vol. 6, no. 1, 2016.

[91] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, 2002.

[92] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, 2008.

[93] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761, pp. C47–C52, 1999.

[94] X. Li, M. Wu, C.-K. Kwoh, and S.-K. Ng, "Computational approaches for detecting protein complexes from protein interaction networks: a survey," *BMC genomics*, vol. 11, no. 1, pp. 1–19, 2010.

[95] M. Li, X. Wu, J. Wang, and Y. Pan, "Towards the identification of protein complexes and functional modules by integrating ppi network and gene expression data," *BMC bioinformatics*, vol. 13, no. 1, pp. 1–15, 2012.

[96] N. Safari-Alighiarloo, M. Taghizadeh, M. Rezaei-Tavirani, B. Goliaei, and A. A. Peyvandi, "Protein-protein interaction networks (ppi) and complex diseases," *Gastroenterology and Hepatology from bed to bench*, vol. 7, no. 1, p. 17, 2014.

[97] S. Mujawar, R. Mishra, S. Pawar, D. Gatherer, and C. Lahiri, "Delineating the plausible molecular vaccine candidates and drug targets of multidrug-resistant acinetobacter baumannii," *Frontiers in cellular and infection microbiology*, vol. 9, p. 203, 2019.

[98] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences*, vol. 98, no. 8, pp. 4569–4574, 2001.

[99] O. Puig, F. Caspary, G. Rigaut, B. Rutz, E. Bouveret, E. Bragado-Nilsson, M. Wilm, and B. Séraphin, "The tandem affinity purification (tap) method: a general procedure of protein complex purification," *Methods*, vol. 24, no. 3, pp. 218–229, 2001.

[100] A.-C. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A.-M. Michon, C.-M. Cruciat *et al.*, "Functional organization of the yeast proteome by systematic analysis of protein complexes," *Nature*, vol. 415, no. 6868, pp. 141–147, 2002.

[101] J. Ji, A. Zhang, C. Liu, X. Quan, and Z. Liu, "Survey: Functional module detection from protein-protein interaction networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 261–277, 2012.

[102] S. Srihari and H. W. Leong, "A survey of computational methods for protein complex prediction from protein interaction networks," *Journal of bioinformatics and computational biology*, vol. 11, no. 02, p. 1230002, 2013.

[103] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[104] G. O. Consortium, "The gene ontology project in 2008," *Nucleic acids research*, vol. 36, no. suppl_1, pp. D440–D444, 2008.

[105] M. Milano, "Gene prioritization tools," 2019.

[106] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, "Network medicine: a network-based approach to human disease," *Nature reviews genetics*, vol. 12, no. 1, pp. 56–68, 2011.

[107] V. Spirin and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *Proceedings of the national Academy of sciences*, vol. 100, no. 21, pp. 12 123–12 128, 2003.

[108] X.-L. Li, C.-S. Foo, and S.-K. Ng, "Discovering protein complexes in dense reliable neighborhoods of protein interaction networks," in *Computational Systems Bioinformatics: (Volume 6)*. World Scientific, 2007, pp. 157–168.

[109] M. Pellegrini, "Community detection in biological networks," 2019.

[110] S. S. Bhowmick and B. S. Seah, "Clustering and summarizing protein-protein interaction networks: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 638–658, 2015.

[111] D. Vella, S. Marini, F. Vitali, D. Di Silvestre, G. Mauri, and R. Bellazzi, "Mtgo: Ppi network analysis via topological and functional module identification," *Scientific reports*, vol. 8, no. 1, pp. 1–13, 2018.

[112] S. Rahiminejad, M. R. Maurya, and S. Subramaniam, "Topological and functional comparison of community detection algorithms in biological networks," *BMC bioinformatics*, vol. 20, no. 1, pp. 1–25, 2019.

[113] R. Guimera and L. A. N. Amaral, "Cartography of complex networks: modules and universal roles," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 02, p. P02001, 2005.

[114] L. Hu and K. C. Chan, "A density-based clustering approach for identifying overlapping protein complexes with functional preferences," *BMC bioinformatics*, vol. 16, no. 1, pp. 1–16, 2015.

[115] X.-F. Zhang, D.-Q. Dai, L. Ou-Yang, and H. Yan, "Detecting overlapping protein complexes based on a generative model with functional and topological properties," *BMC bioinformatics*, vol. 15, no. 1, pp. 1–15, 2014.

[116] M. Wu, X. Li, and C.-K. Kwoh, "Algorithms for detecting protein complexes in ppi networks: an evaluation study," in *Proceedings of third IAPR international conference on pattern recognition in bioinformatics (PRIB 2008)*, 2008, pp. 15–17.

[117] S. Brohee and J. Van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks," *BMC bioinformatics*, vol. 7, no. 1, pp. 1–19, 2006.

[118] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC bioinformatics*, vol. 4, no. 1, pp. 1–27, 2003.

[119] S. vanDongen, "A cluster algorithm for graphs," *Information Systems [INS]*, no. R 0010, 2000.

[120] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein-protein interaction networks," *Nature methods*, vol. 9, no. 5, p. 471, 2012.

[121] M. Wu, X. Li, C.-K. Kwoh, and S.-K. Ng, "A core-attachment based method to detect protein complexes in ppi networks," *BMC bioinformatics*, vol. 10, no. 1, pp. 1–16, 2009.

[122] X.-L. Li, C.-S. Foo, S.-H. Tan, and S.-K. Ng, "Interaction graph mining for protein complexes using local clique merging," *Genome Informatics*, vol. 16, no. 2, pp. 260–269, 2005.

[123] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[124] A. D. King, N. Pržulj, and I. Jurisica, "Protein complex prediction via cost-based clustering," *Bioinformatics*, vol. 20, no. 17, pp. 3013–3020, 2004.

[125] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis *et al.*, "Global landscape of protein complexes in the yeast saccharomyces cerevisiae," *Nature*, vol. 440, no. 7084, pp. 637–643, 2006.

[126] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "Biogrid: a general repository for interaction datasets," *Nucleic acids research*, vol. 34, no. suppl_1, pp. D535–D539, 2006.

[127] P. D. Thomas, A. Kejariwal, M. J. Campbell, H. Mi, K. Diemer, N. Guo, I. Ladunga, B. Ulitsky-Lazareva, A. Muruganujan, S. Rabkin *et al.*, "Panther: a browsable database of gene products organized by biological function, using curated protein family and subfamily classification," *Nucleic acids research*, vol. 31, no. 1, pp. 334–341, 2003.

[128] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak, "Up-to-date catalogues of yeast protein complexes," *Nucleic acids research*, vol. 37, no. 3, pp. 825–831, 2009.

[129] G. Geva and R. Sharan, "Identification of protein complexes from co-immunoprecipitation data," *Bioinformatics*, vol. 27, no. 1, pp. 111–117, 2011.

[130] Q. Dai, M. Guo, Y. Guo, X. Liu, Y. Liu, and Z. Teng, "A least square method based model for identifying protein complexes in protein-protein interaction network," *BioMed research international*, vol. 2014, 2014.

[131] U. Maulik, A. Mukhopadhyay, M. Bhattacharyya, L. Kaderali, B. Brors, S. Bandyopadhyay, and R. Eils, "Mining quasi-bicliques from hiv-1-human protein interaction network: a multiobjective biclustering approach," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 10, no. 2, pp. 423–435, 2012.

[132] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[133] R. Interdonato, M. Atzmueller, S. Gaito, R. Kanawati, C. Largeron, and A. Sala, "Feature-rich networks: going beyond complex network topologies," *Applied Network Science*, vol. 4, no. 1, pp. 1–13, 2019.

[134] P. Cihon, "Standards for ai governance: International standards to enable global coordination in ai research & development," 2019.

[135] "ISO/IEC TR 20547-2: Information technology – Big data reference architecture – Part 2: Use cases and derived requirements," International Organization for Standardization, Geneva, CH, Standard, 2018.

[136] K. Aditya and C. Wheelock, "Artificial intelligence market forecasts," Tractia, Tech. Rep., 2016. [Online]. Available: https://www.tractica.com/wp-content/uploads/2016/08/MD-AIMF-3Q16-Executive-Summary.pdf/

[137] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1322–1333.

[138] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[139] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[140] H. Bae, J. Jang, D. Jung, H. Jang, H. Ha, and S. Yoon, "Security and privacy issues in deep learning," *arXiv preprint arXiv:1807.11655*, 2018.

[141] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE access*, vol. 6, pp. 12 103–12 117, 2018.

[142] N. Samir Labib, C. Liu, S. Dilmaghani, M. R. Brust, G. Danoy, and P. Bouvry, "White paper: Data protection and privacy in smart ict-scientific research and technical standardization," ILNAS ANEC G.I.E/University of Luxembourg, Tech. Rep., 2018.

[143] "ISO/IEC PDTR 24028: Information technology – Artificial Intelligence (AI) – Overview of trustworthiness in Artificial Intelligence," International Organization for Standardization, Geneva, CH, Standard.

[144] D. L. Poole, A. K. Mackworth, and R. Goebel, *Computational intelligence: a logical approach*. Oxford University Press New York, 1998, vol. 1.

[145] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[146] "ISO/IEC WD 22989: Artificial intelligence – Concepts and terminology," International Organization for Standardization, Geneva, CH, Standard.

[147] "ISO/IEC WD 23053: Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)," International Organization for Standardization, Geneva, CH, Standard.

[148] A. M. Fiscarelli, M. R. Brust, G. Danoy, and P. Bouvry, "A memory-based label propagation algorithm for community detection," in *International Conference on Complex Networks and their Applications*. Springer, 2018, pp. 171–182.

[149] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proc. In Conf. on fairness, accountability and transparency*, 2018.

[150] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. D. Tygar, "Antidote: understanding and defending against poisoning of anomaly detectors," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 2009, pp. 1–14.

[151] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[152] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, 2014, pp. 17–32.

[153] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[154] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 533–537.

[155] J. Zhang and X. Jiang, "Adversarial examples: Opportunities and challenges," *arXiv preprint arXiv:1809.04790*, 2018.

[156] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific reports*, vol. 3, p. 1376, 2013.

[157] Y.-A. de Montjoye *et al.*, "Response to comment on "unique in the shopping mall: On the reidentifiability of credit card metadata"," *Science*, vol. 351, no. 6279, pp. 1274–1274, 2016.

[158] J. Andress, *The basics of information security: understanding the fundamentals of InfoSec in theory and practice*. Syngress, 2014.

[159] L. Sweeney, "Simple demographics often identify people uniquely," *Health (San Francisco)*, vol. 671, pp. 1–34, 2000.

[160] E. Jahani, P. Sundsøy, J. Bjelland, L. Bengtsson, Y.-A. de Montjoye *et al.*, "Improving official statistics in emerging markets using machine learning and mobile phone data," *EPJ Data Science*, vol. 6, no. 1, p. 3, 2017.

[161] "ISO/IEC CD 20547-4: Information technology – Big data reference architecture – Part 4: Security and Privacy," International Organization for Standardization, Geneva, CH, Standard.

[162] M. Wall. (2019) Biased and wrong? Facial recognition tech in the dock. [Online]. Available: https://www.bbc.com/news/business-48842750

[163] S. X. Zhang, R. E. Roberts, and D. Farabee, "An analysis of prisoner reentry and parole risk using compas and traditional criminal history measures," *Crime & Delinquency*, vol. 60, no. 2, pp. 167–192, 2014.

[164] "ISO/IEC NP TR 24027: Information technology – Artificial Intelligence (AI) – Bias in AI systems and AI aided decision making," International Organization for Standardization, Geneva, CH, Standard.

[165] A. Newell, R. Potharaju, L. Xiang, and C. Nita-Rotaru, "On the practicality of integrity attacks on document-level sentiment analysis," in *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*. ACM, 2014, pp. 83–93.

[166] J. Vincent. (2016) Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day. [Online]. Available: https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist

[167] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," *CoRR*, vol. abs/1708.06145, 2017.

[168] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[169] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[170] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," technical report, SRI International, Tech. Rep., 1998.

[171] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, 2019.

[172] J. H. Hinnefeld, P. Cooman, N. Mammo, and R. Deese, "Evaluating fairness metrics in the presence of dataset bias," *arXiv preprint arXiv:1809.09245*, 2018.

[173] I. D. Raji and J. Buolamwini, "Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products," in *AAAI/ACM Conf. on AI Ethics and Society*, vol. 1, 2019.

[174] A. Amini, A. Soleimany, W. Schwarting, S. Bhatia, and D. Rus, "Uncovering and mitigating algorithmic bias through learned latent structure," 2019.

[175] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, "How we analyzed the compas recidivism algorithm," *ProPublica (5 2016)*, vol. 9, 2016.

[176] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *Advances in neural information processing systems*, 2017, pp. 3517–3529.

[177] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[178] J. Newsome, B. Karp, and D. Song, "Paragraph: Thwarting signature learning by training maliciously," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2006, pp. 81–105.

[179] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. A. Sutton, J. D. Tygar, and K. Xia, "Exploiting machine learning to subvert your spam filter." *LEET*, vol. 8, pp. 1–9, 2008.

[180] B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto, and F. Roli, "Poisoning behavioral malware clustering," in *Proceedings of the 2014 workshop on artificial intelligent and security workshop*. ACM, 2014, pp. 27–36.

[181] S. Misra, M. Tan, M. Rezazad, M. R. Brust, and N.-M. Cheung, "Early detection of crossfire attacks using deep learning," *arXiv preprint arXiv:1801.00235*, 2017.

[182] M. Rezazad, M. R. Brust, M. Akbari, P. Bouvry, and N.-M. Cheung, "Detecting target-area link-flooding ddos attacks using traffic analysis and supervised learning," in *Future of Information and Communication Conference*. Springer, 2018, pp. 180–202.

[183] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 601–618.

[184] S. J. Oh, M. Augustin, B. Schiele, and M. Fritz, "Towards reverse-engineering black-box neural networks," *arXiv preprint arXiv:1711.01768*, 2017.

[185] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 36–52.

[186] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05, 2005.

[187] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: Membership inference attacks against generative models," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 1, pp. 133–152, 2019.

[188] F.-J. Wu, M. R. Brust, Y.-A. Chen, and T. Luo, "The privacy exposure problem in mobile location-based services," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–7.

[189] F. K. Dankar and K. El Emam, "The application of differential privacy to health data," in *Proceedings of the 2012 Joint EDBT/ICDT Workshops*. ACM, 2012, pp. 158–166.

[190] S. E. Dilmaghani, "A privacy-preserving solution for storage and processing of personal health records against brute-force attacks," Ph.D. dissertation, Bilkent University, 2017.

[191] OPAL. (2017) Open Algorithms. [Online]. Available: http://www.opalproject.org/

[192] F. Kamiran, A. Karim, and X. Zhang, "Decision theory for discrimination-aware classification," in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 924–929.

[193] J. A. Adebayo *et al.*, "Fairml: Toolbox for diagnosing bias in predictive modeling," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[194] R. K. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic *et al.*, "Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," *arXiv preprint arXiv:1810.01943*, 2018.

[195] B. Biggio, I. Corona, B. Nelson, B. I. Rubinstein, D. Maiorca, G. Fumera, G. Giacinto, and F. Roli, "Security evaluation of support vector machines in adversarial environments," in *Support Vector Machines Applications*. Springer, 2014, pp. 105–153.

[196] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. C. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," *arXiv preprint arXiv:1802.03041*, 2018.

[197] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," *arXiv preprint arXiv:1902.06531*, 2019.

[198] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. Sutton, J. Tygar, and K. Xia, "Misleading learners: Co-opting your spam filter," in *Machine learning in cyber trust*. Springer, 2009, pp. 17–51.

[199] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "Prada: protecting against dnn model stealing attacks," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 512–527.

[200] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with pate," *arXiv preprint arXiv:1802.08908*, 2018.

[201] T. Lee, B. Edwards, I. Molloy, and D. Su, "Defending against machine learning model stealing attacks using deceptive perturbations," *arXiv preprint arXiv:1806.00054*, 2018.

[202] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[203] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 135–147.