Big Data Clustering: Models and Applications

Big Data Clustering: Models and Applications

By
DEWAN. F. WAHID,
B.Sc., M.S. (Mathematics); M.Sc. (Computer Science)

A Thesis Submitted to the School of Graduate Studies
in the Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy
in
Computational Science and Engineering

McMaster University
Hamilton, Ontario

Doctor of Philosophy (2023)
Computational Science & Engineering
McMaster University
Hamilton, Ontario, Canada

TITLE: Big Data Clustering: Models and Applications

AUTHOR:
Dewan. F. Wahid,
B.Sc., M.S. (Mathematics, University of Chittagong, Bangladesh);
M.Sc. (Computer Science, University of British Columbia, BC, Canada)

SUPERVISOR:
Dr. Elkafi Hassini
Professor, Operations Management, DeGroote School of Business,
McMaster University, ON, Canada

SUPERVISORY COMMITTEE CHAIR:
Dr. Kai Huang,
Associate Professor, Operations Management, DeGroote School of Business,
McMaster University, ON, Canada

SUPERVISORY COMMITTEE MEMBERS:
Dr. Wael El-Dakhakhni
Professor, Civil Engineering,
McMaster University, ON, Canada

Dr. Manish Verma
Professor, Operations Management, DeGroote School of Business,
McMaster University, ON, Canada

EXTERNAL EXAMINATION COMMITTEE MEMBER:
Dr. Stan Dimitrov
Professor, Department of Management Sciences, University of Waterloo, ON, Canada

NUMBER OF PAGES: xvi, 191

Lay Abstract

This thesis presents big data clustering frameworks that tackle application problems in different real-world scenarios. Primarily, two main approaches have been used in developing these clustering frameworks. The first approach utilizes problem-specific keywords network formulation and network (graph) clustering models with corresponding integer linear programming formulation-based heuristic algorithms, which can identify communities or clusters in big datasets. Furthermore, different procedures were followed based on related application areas to interpret and utilize identified clusters or communities. The second approach is an augmented artificial intelligence hybrid framework of unsupervised clustering and supervised classifiers with a set of minimal labelled data. All approaches have been tested with real-world data that included university researchers' publication networks and subscription-based accounting firm customers' transactions' network data. In addition, this thesis presents a cross-disciplinary taxonomy-based literature review and a bibliometric analysis for correlation clustering, a well-known network clustering problem.

Abstract

This thesis presents frameworks for data clustering on big datasets that can arise in different real-world applications. The main contributions of this thesis can be divided into the following four areas of data clustering.

Correlation clustering is a well-known problem that appears in different scientific areas with various names that identify clusters when qualitative information about objects' mutual similarities or dissimilarities is given. The first contribution of this thesis is to present a unified discussion on the cross-disciplinary taxonomy-based literature review, bibliometric analysis, literature gaps and dominant research topics related to this problem.

As the second contribution, this thesis presents the concept of a common-knowledge network and a heuristic algorithm for clustering editing to identify authors' communities in a research institution. Furthermore, several analyses, such as the dominant research topic and collaboration incident corresponding to each identified research community, are proposed in this thesis to investigate multidisciplinary research activities in research institutions.

The third contribution constitutes a framework for user-generated short-text classification based on identified line-item categories. The line-item identification phase uses the Cograph Editing-based clustering on keywords network formulated from short-texts. An integer linear programming formulation for the Cograph Editing on weighted networks and a corresponding heuristic algorithm to identify clusters in large-scale networks are also proposed. The framework has been applied to categorize invoices for a subscription-based invoicing and accounting company.

An augmented artificial intelligence (AI) hybrid fraud detection framework in the presence of minimal labelled data sets. This framework uses unsupervised clustering, a supervised classifier, red-flag prioritization, and augmented AI processes. Finally, this thesis outlines an application of this framework to identify fraudulent users in an invoicing platform.

Acknowledgements

First, I would like to thank my supervisor Professor Elkafi Hassini whose knowledge and experience inspired this thesis. Without his guidance and continuous support, I would not have been able to accomplish all that I have. Thank you, Professor.

I would like to thank my committee members, Professor Wael El-Dakhakhni and Professor Manish Verma, both of whom have provided invaluable input to my work over the years.

I also want to thank the external examination committee member Professor Stan Dimitrov sincerely for his valuable suggestions and discussions during the final thesis defence.

I was fortunate to be offered MITACS (Mathematics of Information Technology and Complex Systems) Fellowships with an industrial partner that has largely shaped the academic contributions in this thesis and equipped me with invaluable computational and industrial skills that will undoubtedly have a lasting impact on my future career development.

I want to express my gratitude to my Hamilton friends, especially Nazmul, Roqibul, Neela, Rubel & Esha, Manas & Puja, Grytan & Bayshakhi, Bashudev & Bornali, Roby & Lira, and Rabiul. They support me as my family when it is needed. I feel blessed to have such wonderful friends around me.

Finally, I am grateful to my family members; Dewan Deloar Hossain (father); Mst Farida Yeasmin (mother); Shakila Hossain (wife); and Kurratul Ayin (sister), for keeping me inside a safety net of love and care in any situation of my life.

Contents

List of Figures

List of Tables

Declaration of Academic Achievement

This dissertation, titled, **Big Data Clustering: Models and Applications**, was prepared in accordance with the guidelines set by the school of graduate studies at McMaster University for a sandwich thesis. I, Dewan. F. Wahid, declare that this thesis and works presented in it are my own with the guidance and supervision of Prof. Elkafi Hassini with further editorial comments and computational support provided by Drs Wael El-Dakhakhni and Mohamed Ezzeldin, in Chapter 3. The main chapters (Chapters 2 through 5) contain scholarly work either published or submitted for publication.

As this thesis contains materials published, submitted, or accepted for publication in journals, all steps have been taken to ensure that the necessary copyright limitations and rights have been respected.

Chapter 1

Introduction

Clustering is an unsupervised machine learning that identifies natural and meaningful partitions or clusters in a set of data based on their mutual similarity (Usama et al.; 2019). It also organizes data in such a way that rewards high intra-cluster and low inter-cluster similarity (Wunsch and Xu; 2008). Therefore, clustering aims to identify latent patterns in the input data, and it is widely used in many scientific disciplines, such as data mining, social network analysis, machine learning and pattern recognition (Zhang et al.; 2021; Best et al.; 2022). In this thesis, we designed clustering based-models for identifying clusters in different application problems.

1.1 Motivation

The works presented in this thesis are motivated by the huge investments in big data infrastructures by companies and the increasing availability of big data sets. In such environments, senior management is putting pressure on data science and analytics departments to establish healthy returns on investments in big data capabilities. The author's MITACS (Mathematics of Information Technology and Complex Systems) internship with an industrial partner has been instrumental in grounding this research in real problems and providing big data sets and computational platforms.

The thesis has a common goal of designing data clustering models that can be implemented in different empirical settings. In this regard, our research was derived from data availability in research indexing platforms and industrial partners. In addition to identifying clusters from data and developing associated algorithms, this thesis also focused on interpreting those clusters based on corresponding application areas.

1.2 Background

1.2.1 Data Clustering Definition and its Classification

This section focuses on the general definition and classification of clustering. The clustering problem can be defined formally as follows (Breaban and Luchian; 2011):

Definition 1.1 *Let* $X = \{x_1, x_2, ..., x_n\}$ *where* $x_i = \{x_{i1}, x_{i2}, ..., x_{il}\} \in \mathbb{R}^m$*, be a set of* n *data points with* l *numerical features. Also, let* $\mathcal{C} = \{C_1, C_2, ..., C_k\}$*, such that* $\bigcup_{p=1}^{k} C_p = \mathcal{X}$ *and* $C_p \cap C_q = \emptyset$*,* $\forall p, q = 1, 2, .., k$*;* $q \neq q; k \in \{1, 2, ..., card(\mathcal{C})\}$*, be a possible partition (set of clusters) of the input data. The clustering problem identifies an optimal partition* $\mathcal{C}^*$ *such that*

$$\mathcal{C}^* = \arg\max_{\mathcal{C} \in \Omega} \mathcal{F}(\mathcal{C}), \tag{1.1}$$

where Ω *is the set of all possible partitions of the input data* $\mathcal{X}$*, and* $\mathcal{F}$ *is a function that measures the quality of each partition* $\mathcal{C} \in \Omega$*.*

Generally, the quality function $\mathcal{F}$ is based on the data structure and domain of application of the clustering algorithm. Many clustering problems have been proposed in the past few decades, such as K-Means, Fuzzy C-Means, hierarchical clustering, Clustering Editing, and Correlation Clustering (Bansal et al.; 2004), targeting different data structures and applications.

1.2.2 Challenges with Big Data Clustering

Clustering algorithms are optimization problems, most often NP-hard, and they are very effective in extracting useful patterns from data (Xu and Tian; 2015). However, these algorithms come with high computational costs due to the high dimensionality and complexity associated with contemporary data (Zerhari et al.; 2015; Saeed et al.; 2020). The digital revolution in every sector in recent years is adding more challenges to this issue that urges more research for designing improved clustering algorithms for big data. In the discussion of big data, often the following question arises: 'how big is the big data?' To address this issue, Shirkhorshidi et al. (2014) presented a scale for data sizes categorization based on Hathaway and Bezdek (2006)'s study (given in Table 1.1). It is worth noting that size is not the only defining factor for big data. As discussed below, there are four other defining aspects of big data. For example, big data

can be characterized by small data that is generated from a variety of sources and has heterogeneous types, such as numeric, text and graphics.

			Big Data		
Bytes	10^6	10^8	10^{10}	10^{12}	$10^{>12}$
Size	Medium	Large	Huge	Monster	Very Large

TABLE 1.1: Hathaway and Bezdek (2006) categorization for big data (Shirkhorshidi et al.; 2014).

The challenges of clustering big data are characterized into five main components (Shirkhorshidi et al.; 2014; Sivarajah et al.; 2017; Cappa et al.; 2021):

- **Volume**: The scale of data is increasing exponentially due to modern technology. Unfortunately, clustering algorithms do not scale with big datasets (Mahdi et al.; 2021) and require more costly computing resources (Jagadish et al.; 2014).
- **Velocity**: Due to real-time tracking in every platform, the velocity of incoming real-time data is increasing drastically. Clustering algorithms must improve to deal with the high rate of dynamic data that can provide useful information in real-time (Khalilian and Mustapha; 2010).
- **Variety**: Data are generated from different sources with different features. Therefore, managing, merging, and governing unstructured and heterogeneous data is extremely challenging (Lomotey and Deters; 2013).
- **Variability**: Another challenge of managing unstructured and heterogeneous big data is inconsistent data flow, i.e., having short daily or seasonal peak/off-peak load (Lomotey and Deters; 2013).
- **Complexity**: Connecting data with correlated relationships and data linkage from different sources with different features is necessary. But unfortunately, the complexity gets out of control quickly in the case of unstructured and heterogeneous big data (Altman and Raychaudhuri; 2001).

The main research goal in this area of research is to scale up and speed up clustering algorithms without sacrificing the clustering quality. In the research literature, big data clustering algorithms can be classified into two major categories: single-machine (Alguliyev et al.; 2020) and multiple-machine clustering (Zerhari et al.; 2015). Single-machine clustering algorithms include sample and dimension-reduction-based approximate and heuristic algorithms (Djouzi and Beghdad-Bey; 2019). On the other hand,

multiple-machine clustering algorithms include parallel and MapReduce-based heuristic clustering (Chierichetti et al.; 2014; Pan et al.; 2015; Shi et al.; 2021).

Motivated by the computing infrastructure that is available to our industrial partner, and indeed most small to medium enterprises, this thesis primarily focused on designing single-machine-based heuristic clustering algorithms that can be implemented in empirical settings. In future research, the proposed algorithms can be extended to multiple-machine (parallel or MapReduce) algorithms.

1.2.3 Research Community Platform

Due to the web revolution, many communities have evolved based on common interests, such as sharing programming or technical support (e.g., Stack Overflow and GitHub) and reviewing movies (e.g., IMBD and Rotten Tomatoes) in the past few decades. In general, a research community is very similar to any other community where a group of researchers come together to share and discuss research ideas and share data and publication lists (Brandes; 2005; Clauset et al.; 2004). Like other communities, research community members also tend to query, discover and monitor relationships among their peers in the community (DeRose et al.; 2007). Examples include arXiv, ResearchGate, Social Science Research Network (SSRN), and Web of Science. Researchers subscribe to these networks for different purposes, including the need to identify the potential for collaboration across the disciplines inside or outside of their organization and analyze mutual interactions among the corresponding members inside those communities. To construct and discover mutual relationships and community-wise attributes, many network formulations, such as collaboration networks (Newman; 2001) and co-citation networks (Egghe and Rousseau; 2002), have been proposed in the research literature (Yan and Ding; 2012; Ji et al.; 2022).

1.2.4 Subscription-based Invoicing Platforms

The use of electronic invoices is becoming ubiquitous in all sectors of the industry (Cedillo et al.; 2018). The COVID-19 pandemic has reinforced this trend and helped accelerate the move to online transactions for all sizes and types of businesses. Companies generally use different invoicing platforms to generate their business invoices, send these invoices electronically to their clients, and receive payments directly to their banks (Asatiani et al.; 2019). These platforms are efficient and reliable, and they provide real-time access to business data (Christauskas and Miseviciene; 2012). Several studies indicate that small and medium-sized enterprises (SMEs) can avoid several red-tapes and mitigate

cash flow issues using invoicing platforms to manage their accounting and bookkeeping (Lee; 2016; Guerar et al.; 2020). However, maintaining an 'in-house' IT department to run an invoicing platform (system) is often very expensive for many SMEs. Following the increasing demand, many Cloud-based Accounting and Invoicing Service (CB-AIS) companies started offering subscription-based invoicing platforms to SMEs (Asatiani and Penttinen; 2015). With these subscription-based invoicing platforms, SMEs are undertaking their accounting and invoicing in the cloud instead of hiring external accountants or IT personnel (Ma et al.; 2021).

With growing popularity, subscription-based invoicing platforms are facing different practical challenges. Chapters 4 and 5 of this thesis focus on two such challenges in subscription-based invoicing platforms: categorizing invoices based on line-items and identifying fraudulent users.

1.3 Thesis Overview and Contributions

The next four chapters (Chapters 2–5) of this thesis present four publication manuscripts (published and submitted) related to data clustering literature, frameworks and applications. These frameworks were inspired by empirical problems. They were designed using real-world data and tested in real-world settings. Finally, Chapter 6 presents an overall conclusion and future research directions for this thesis. Brief overviews of contributions corresponding to Chapters 2–5 are provided below.

1.3.1 Correlation Clustering: Cross-Disciplinary Taxonomy with Bibliometric Analysis

Correlation clustering is a multidisciplinary problem that identifies clusters when qualitative information about objects' mutual similarities or dissimilarities is given in a signed network. In this signed network, each node represents an object, and each link represents the mutual similarity or dissimilarity between two objects. The Correlation clustering problem is suitable for clustering objects when obtaining the features vector is difficult (Bonchi et al.; 2022), and has several applications in data mining (Néda et al.; 2006; Cohen and Richman; 2002), aggregating multiple clusters (Gionis et al.; 2007), and link classification (Cesa-Bianchi et al.; 2012). Over the years, the Correlation clustering problem appeared in the research literature in different scientific areas in various forms and names. Contributions in these fields have largely been islands of knowledge with little effort to link them and ensure a consistent effort of advancing Correlation clustering knowledge with no significant overlap. For example, in the social context, it was

proposed as the 'structural balance' problem Doreian and Mrvar (1996); in the signed network-based graph modification context, it was referred to as the 'clustering editing problem' Shamir et al. (2004); in the business management context it was introduced as 'multiple correlation clustering' Doyle (1992); and finally, in the data mining context it proposed as the 'correlation clustering' problem Bansal et al. (2004).

Chapter 2 of this thesis consists of the submitted text of a paper published in the *Operations Research Forum* journal, which presents a cross-disciplinary taxonomy and bibliometric analysis-based literature review on the Correlation clustering problem. The citation of this work is given below.

> Wahid, D. F., Hassini, E.(2022). A Literature Review on Correlation Clustering: Cross-Disciplinary Taxonomy with Bibliometric Analysis, *Operations Research Forum*. 3(3): 1-42. URL.

The study presented in this chapter extends a unified discussion, including a detailed discussion of mathematical formulations and solution approaches, to enhance the cross-fertilization of knowledge and mitigate gaps between disciplinary approaches to the Correlation clustering problem. In addition, a taxonomic development for several variant classes, solution procedures and applications of the Correlation clustering problem is also presented in this chapter. Furthermore, it provides a bibliometric-based analysis to investigate the collaborations, citation progressions, dominant research topics, and knowledge clusters in this area.

1.3.2 Common-Knowledge Network-based Research Community Clustering

A group of researchers who share similar knowledge or expertise and use common methodologies can be defined as a research community (Malmberg and Maskell; 2002). Identifying and analyzing knowledge-based research communities helps research institutions adopt appropriate policies for knowledge-creating investments, including utilizing the latest technology to stimulate research and developing activities (Connelly et al.; 2012; Malmberg and Maskell; 2002). Several network formulations, such as collaboration networks (Clauset et al.; 2004; Hu et al.; 2019), co-citation networks (Ding; 2011; Muñoz-Muñoz and Mirón-Valdivieso; 2017), and co-word networks based on publication keywords (Katsurai; 2017; Zhao et al.; 2018), have been proposed in the literature to capture the knowledge structure of a research institution or a scientific field.

Chapter 3 of this thesis consists of the submitted manuscript of a paper published in the *Decision Analytics Journal*, which presents the concept of a 'common-knowledge network' of authors in a research institution based on the mutual commonality of knowledge elements extracted from corresponding published articles. The citation of this published work is presented below.

> Wahid, D. F., Ezzeldin, M., Hassini, E., & El-Dakhakhni, W. W. (2022). Common-Knowledge Networks for University Strategic Research Planning. *Decision Analytics Journal*, 2, 100027. URL.

The study presented in this chapter illustrates a common-knowledge network-based model to investigate the interdisciplinary research productivity among researchers over time. This study considers publication keywords as the knowledge elements that represent authors' areas of expertise and formulate a network based on the commonality of knowledge elements (keywords) among the researchers. In addition, a heuristic algorithm for clustering editing problems on weighted networks is presented to identify research communities from the formulated common-knowledge network. Furthermore, to illustrate the synergy and interaction among researchers in identified communities, several metrics or topics, such as collaboration and publication count, dominating research topic, top influential authors, and affiliated departments, are presented in this study, corresponding to each research community.

1.3.3 Short-Text Classification Framework and Invoice Line-Item Identification

Without accurate labelling, millions of user-generated short-texts appear on different online platforms and marketplaces (Cevahir and Murakami; 2016). These short-texts generally have a limited number of words (Inches et al.; 2010), semantic properties (Sriram et al.; 2010) and contextual information (Song et al.; 2011), and consist of different noises, such as misspellings and grammatical errors (Hadar and Shmueli; 2021). Short-texts are more challenging to classify and analyze than long and well-written texts, such as news articles and textbooks.

Chapter 4 of this thesis consists of the submitted manuscript for publication, which presents a framework for classifying user-generated short and noisy texts based on the mutual co-existing keywords relationship. The submitted title of this manuscript is as below, which is under review at the time of writing.

> Wahid, D. F., Hassini, E.. User-Generated Short Text Classification using Cograph Editing-based Network Clustering with an Application in Invoice Categorization.

The study presented in this chapter outlines a process of identifying line-item categories (classifying labels) for short-text classification based on keyword (co-exiting) networks. This study proposes a Cograph Editing-based heuristic clustering algorithm using an integer linear programming formulation to identify keyword clusters from large-scale weighted networks. Furthermore, an application of the proposed framework to identify and classify invoices based on line-item categories is also presented in this study.

1.3.4 Hybrid Fraud Detection Framework for Invoicing Platforms

In recent years, invoice-related fraud incidents have been surging using various invoicing platforms (Guerar et al.; 2020). Multiple studies show that small and medium-sized enterprises (SMEs) are more vulnerable to invoice fraud and suffer disproportionate losses than large companies (Kramer; 2015). In addition to harming its users, invoice fraud damages the public reputation and creates financial dents to the service-providing company (Guerar et al.; 2020).

Chapter 5 of this thesis consists of another submitted manuscript for publication, which presents a hybrid fraud detection framework for invoicing platforms. The submitted title of this manuscript is as below, which is under review at the time of writing.

> Wahid, D. F., Hassini, E.. Hybrid Fraud Detection Framework in Invoicing Platforms using an Augmented AI Approach

The study presented in this chapter outlines a hybrid framework for identifying fraudulent users on invoicing platforms where human review is required in the final decision-making process. This framework uses a combination of unsupervised and supervised machine learning processes and a small amount of labelled data to identify fraud risk cluster(s) in the model training process. In addition, in this study, a weighted center for the fraud risk cluster based on the feature importance score is also presented and used in the red-flag prioritization and augmented AI processes. Finally, a case study of the hybrid framework in a weekly segmented structure is also discussed in this study.

1.4 Challenges of Conducting Research with Real-World Big Data

The big data generated in the real world produces new opportunities for discovering new values of any system and helps us to have a deeper understanding of the hidden structures (Shehab et al.; 2021). However, these real-world data are noisy and unstructured and often incur new challenges to organize and manage effectively (Palominos et al.; 2017). Acquiring and preprocessing big data for industries are time-consuming and computationally expensive (Zhang et al.; 2003; Luengo et al.; 2020). Many researchers have discussed the issue of getting approval and overcoming red tape to acquire data in the public sectors (Ung et al.; 2019; Hsieh et al.; 2018). From our experience, this issue of acquiring data from a fin-tech private company creates another level of complexity, such as building trust with the business partner and handling the restrictions that non-disclosure agreements could put on reporting on the research work. Other researchers, such as Samarajiva et al. (2015) and Taylor et al. (2014), briefly focused on this issue in their studies.

A recent survey (Press; 2016) shows that data scientists spend almost 80% of their time acquiring and preprocessing data (Luengo et al.; 2020). While the opportunity to do Ph.D. research in partnership with industrial partners offers deep insights into their respective fields, it also comes with several challenges. First, while more than 80% of a researcher's time is spent on data preprocessing, not much of that work is worthy of reporting in the dissertation. This could create gaps in knowledge translation and research reproducibility. The second challenge occurs in the research problem formulation when the researcher needs to balance the immediate needs for a "satisficing" solution by the industry partner with the requirement to position the research in the literature and ensure that one is filling some gaps in the literature. The third challenge is concerned with maintaining a balance between respecting the industrial partners' need for confidentiality and the expectation that research in a public institution should have open access. This matter also affects the reproducibility of scholarly works in the future. Many researchers, such as Edwards (2016) and Mebane et al. (2019), have discussed the challenge of reproducibility of industrial collaboration research and possible incentive-based approaches for industries to overcome it.

Author's Statement of Contribution

I am the author of this thesis and the first author of all works published or submitted for articles included in this thesis.

Chapter References

Alguliyev, R. M., Aliguliyev, R. M. and Sukhostat, L. V. (2020). Efficient algorithm for big data clustering on single machine, *CAAI Transactions on Intelligence Technology* **5**(1): 9–14.

Altman, R. B. and Raychaudhuri, S. (2001). Whole-genome expression analysis: challenges beyond clustering, *Current Opinion in Structural Biology* **11**(3): 340–347.

Asatiani, A., Apte, U., Penttinen, E., Rönkkö, M. and Saarinen, T. (2019). Impact of accounting process characteristics on accounting outsourcing-comparison of users and non-users of cloud-based accounting information systems, *International Journal of Accounting Information Systems* **34**: 100419.

Asatiani, A. and Penttinen, E. (2015). Managing the move to the cloud–analyzing the risks and opportunities of cloud-based accounting information systems, *Journal of Information Technology Teaching Cases* **5**(1): 27–34.

Bansal, N., Blum, A. and Chawla, S. (2004). Correlation clustering, *Machine Learning* **56**(1): 89–113.

Best, L., Foo, E. and Tian, H. (2022). Utilising k-means clustering and naive bayes for IoT anomaly detection: A hybrid approach, *Secure and Trusted Cyber Physical Systems*, pp. 177–214.

Bonchi, F., García-Soriano, D. and Gullo, F. (2022). Correlation clustering, *Synthesis Lectures on Data Mining and Knowledge Discovery* **12**(1): 1–149.

Brandes, U. (2005). *Network analysis: Methodological foundations*, Vol. 3418, Springer Science & Business Media.

Breaban, M. and Luchian, H. (2011). A unifying criterion for unsupervised clustering and feature selection, *Pattern Recognition* **44**(4): 854–865.

Cappa, F., Oriani, R., Peruffo, E. and McCarthy, I. (2021). Big data for creating and capturing value in the digitalized environment: Unpacking the effects of volume,

variety, and veracity on firm performance, *Journal of Product Innovation Management* **38**(1): 49–67.

Cedillo, P., García, A., Cárdenas, J. D. and Bermeo, A. (2018). A systematic literature review of electronic invoicing, platforms and notification systems, *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*, IEEE, pp. 150–157.

Cesa-Bianchi, N., Gentile, C., Vitale, F. and Zappella, G. (2012). A correlation clustering approach to link classification in signed networks, *Conference on Learning Theory*, JMLR Workshop and Conference Proceedings, pp. 34–1.

Cevahir, A. and Murakami, K. (2016). Large-scale multi-class and hierarchical product categorization for an e-commerce giant, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 525–535.

Chierichetti, F., Dalvi, N. and Kumar, R. (2014). Correlation clustering in mapreduce, *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 641–650.

Christauskas, C. and Miseviciene, R. (2012). Cloud–computing based accounting for small to medium sized business, *Engineering Economics* **23**(1): 14–21.

Clauset, A., Newman, M. E. and Moore, C. (2004). Finding community structure in very large networks, *Physical Review E* **70**(6): 066111.

Cohen, W. W. and Richman, J. (2002). Learning to match and cluster large high-dimensional data sets for data integration, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 475–480.

Connelly, C. E., Zweig, D., Webster, J. and Trougakos, J. P. (2012). Knowledge hiding in organizations, *Journal of Organizational Behavior* **33**(1): 64–88.

DeRose, P., Shen, W., Chen, F., Lee, Y., Burdick, D., Doan, A. and Ramakrishnan, R. (2007). Dblife: A community information management platform for the database research community, *CIDR*, pp. 169–172.

Ding, Y. (2011). Scientific collaboration and endorsement: Network analysis of coauthorship and citation networks, *Journal of Informetrics* **5**(1): 187–203.

Djouzi, K. and Beghdad-Bey, K. (2019). A review of clustering algorithms for big data, *2019 International Conference on Networking and Advanced Systems (ICNAS)*, IEEE, pp. 1–6.

Doreian, P. and Mrvar, A. (1996). A partitioning approach to structural balance, *Social Networks* **18**(2): 149–168.

Doyle, J. R. (1992). MCC—multiple correlation clustering, *International journal of man-machine studies* **37**(6): 751–765.

Edwards, A. (2016). Reproducibility: team up with industry, *Nature* **531**(7594): 299–301.

Egghe, L. and Rousseau, R. (2002). Co-citation, bibliographic coupling and a characterization of lattice citation networks, *Scientometrics* **55**(3): 349–361.

Gionis, A., Mannila, H. and Tsaparas, P. (2007). Clustering aggregation, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **1**(1): 4–es.

Guerar, M., Merlo, A., Migliardi, M., Palmieri, F. and Verderame, L. (2020). A fraud-resilient blockchain-based solution for invoice financing, *IEEE Transactions on Engineering Management* **67**(4): 1086–1098.

Hadar, Y. and Shmueli, E. (2021). Categorizing items with short and noisy descriptions using ensembled transferred embeddings, *arXiv preprint* .

Hathaway, R. J. and Bezdek, J. C. (2006). Extending fuzzy and probabilistic clustering to very large data sets, *Computational Statistics & Data Analysis* **51**(1): 215–234.

Hsieh, C.-Y., Wu, D. P. and Sung, S.-F. (2018). Registry-based stroke research in taiwan: past and future, *Epidemiology and health* **40**.

Hu, Z., Lin, A. and Willett, P. (2019). Identification of research communities in cited and uncited publications using a co-authorship network, *Scientometrics* **118**(1): 1–19.

Inches, G., Carman, M. J. and Crestani, F. (2010). Statistics of online user-generated short documents, *European Conference on Information Retrieval*, Springer, pp. 649–652.

Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R. and Shahabi, C. (2014). Big data and its technical challenges, *Communications of the ACM* **57**(7): 86–94.

Ji, P., Jin, J., Ke, Z. T. and Li, W. (2022). Co-citation and co-authorship networks of statisticians, *Journal of Business & Economic Statistics* **40**(2): 469–485.

Katsurai, M. (2017). Bursty research topic detection from scholarly data using dynamic co-word networks: A preliminary investigation, *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, IEEE, pp. 115–119.

Khalilian, M. and Mustapha, N. (2010). Data stream clustering: Challenges and issues, *arXiv preprint* .

Kramer, B. (2015). Trust, but verify: Fraud in small businesses, *Journal of Small Business and Enterprise Development* **22**(1): 4–20.

Lee, H. C. (2016). Can electronic tax invoicing improve tax compliance? A case study of the Republic of Korea's electronic tax invoicing for value-added tax, *World Bank Policy Research Working Paper* (7592).

Lomotey, R. K. and Deters, R. (2013). Rsenter: Tool for topics and terms extraction from unstructured data debris, *2013 IEEE International Congress on Big Data*, pp. 395–402.

Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S., Herrera, F., Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S. and Herrera, F. (2020). Final thoughts: From big data to smart data, *Big Data Preprocessing: Enabling Smart Data* pp. 183–186.

Ma, D., Fisher, R. and Nesbit, T. (2021). Cloud-based client accounting and small and medium accounting practices: Adoption and impact, *International Journal of Accounting Information Systems* **41**: 100513.

Mahdi, M. A., Hosny, K. M. and Elhenawy, I. (2021). Scalable clustering algorithms for big data: A review, *IEEE Access* **9**: 80015–80027.

Malmberg, A. and Maskell, P. (2002). The elusive concept of localization economies: towards a knowledge-based theory of spatial clustering, *Environment and Planning A: Economy and Space* **34**(3): 429–449.

Mebane, C. A., Sumpter, J. P., Fairbrother, A., Augspurger, T. P., Canfield, T. J., Goodfellow, W. L., Guiney, P. D., LeHuray, A., Maltby, L., Mayfield, D. B. et al. (2019). Scientific integrity issues in environmental toxicology and chemistry: Improving research reproducibility, credibility, and transparency, *Integrated environmental assessment and management* **15**(3): 320–344.

Muñoz-Muñoz, A. M. and Mirón-Valdivieso, M. D. (2017). Analysis of collaboration and co-citation networks between researchers studying violence involving women., *Information Research: An International Electronic Journal* **22**(2): n2.

Néda, Z., Florian, R., Ravasz, M., Libál, A. and Györgyi, G. (2006). Phase transition in an optimal clusterization model, *Physica A: Statistical Mechanics and its Applications* **362**(2): 357–368.

Newman, M. E. (2001). Scientific collaboration networks. i. network construction and fundamental results, *Physical Review E* **64**(1): 016131.

Palominos, F., Díaz, H., Córdova, F., Cañete, L., Durí, C. et al. (2017). A solution for problems in the organization, storage and processing of large data banks of physiological variables, *International Journal of Computers Communications & Control* **12**(2): 276–290.

Pan, X., Papailiopoulos, D., Oymak, S., Recht, B., Ramchandran, K. and Jordan, M. I. (2015). Parallel correlation clustering on big graphs, *Advances in Neural Information Processing Systems* **28**.

Press, G. (2016). Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says, `https://bit.ly/3HsQpgI`. [Accessed 02-May-2023].

Saeed, M. M., Al Aghbari, Z. and Alsharidah, M. (2020). Big data clustering techniques based on Spark: A literature review, *PeerJ Computer Science* **6**: e321.

Samarajiva, R., Lokanathan, S., Madhawa, K., Kreindler, G. and Maldeniya, D. (2015). Big data to improve urban planning, *Economic and Political Weekly* pp. 42–48.

Shamir, R., Sharan, R. and Tsur, D. (2004). Cluster graph modification problems, *Discrete Applied Mathematics* **144**(1-2): 173–182.

Shehab, N., Badawy, M. and Arafat, H. (2021). Big data analytics and preprocessing, *Machine learning and big data analytics paradigms: analysis, applications and challenges* pp. 25–43.

Shi, J., Dhulipala, L., Eisenstat, D., Łącki, J. and Mirrokni, V. (2021). Scalable community detection via parallel correlation clustering, *arXiv preprint* .

Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y. and Herawan, T. (2014). Big data clustering: A review, *International conference on computational science and its applications*, Springer, pp. 707–720.

Sivarajah, U., Kamal, M. M., Irani, Z. and Weerakkody, V. (2017). Critical analysis of big data challenges and analytical methods, *Journal of business research* **70**: 263–286.

Song, Y., Wang, H., Wang, Z., Li, H. and Chen, W. (2011). Short text conceptualization using a probabilistic knowledgebase, *Twenty-second International Joint Conference on Artificial Intelligence*.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H. and Demirbas, M. (2010). Short text classification in twitter to improve information filtering, *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 841–842.

Taylor, L., Schroeder, R. and Meyer, E. (2014). Emerging practices and perspectives on big data analysis in economics: Bigger and better or more of the same?, *Big Data & Society* **1**(2): 2053951714536877.

Ung, D., Kim, J., Thrift, A. G., Cadilhac, D. A., Andrew, N. E., Sundararajan, V., Kapral, M. K., Reeves, M. and Kilkenny, M. F. (2019). Promising use of big data to increase the efficiency and comprehensiveness of stroke outcomes research, *Stroke* **50**(5): 1302–1309.

Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.-L. A., Elkhatib, Y., Hussain, A. and Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges, *IEEE Access* **7**: 65579–65615.

Wunsch, D. and Xu, R. (2008). *Clustering*, John Wiley & Sons.

Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms, *Annals of Data Science* **2**(2): 165–193.

Yan, E. and Ding, Y. (2012). Scholarly network similarities: How bibliographic coupling networks, citation networks, cocitation networks, topical networks, coauthorship networks, and coword networks relate to each other, *Journal of the American Society for Information Science and Technology* **63**(7): 1313–1326.

Zerhari, B., Lahcen, A. A. and Mouline, S. (2015). Big data clustering: Algorithms and challenges, *Proc. of Int. Conf. on Big Data, Cloud and Applications (BDCA'15)*.

Zhang, S., Zhang, C. and Yang, Q. (2003). Data preparation for data mining, *Applied artificial intelligence* **17**(5-6): 375–381.

Zhang, Y., Li, M., Wang, S., Dai, S., Luo, L., Zhu, E., Xu, H., Zhu, X., Yao, C. and Zhou, H. (2021). Gaussian mixture model clustering with incomplete data, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **17**(1s): 1–14.

Zhao, W., Mao, J. and Lu, K. (2018). Ranking themes on co-word networks: Exploring the relationships among different metrics, *Information Processing & Management* **54**(2): 203–218.

Chapter 2

A Literature Review on Correlation Clustering: Cross-Disciplinary Taxonomy with Bibliometric Analysis

The content of this chapter is the published manuscript for publication under the following citation and cited as Wahid and Hassini (2022):

Wahid, D. F., Hassini, E.(2022). A Literature Review on Correlation Clustering: Cross-Disciplinary Taxonomy with Bibliometric Analysis, *Operations Research Forum*. 3(3): 1-42. URL.

A Literature Review on Correlation Clustering: Cross-Disciplinary Taxonomy with Bibliometric Analysis

Dewan F. Wahid
School of Computational Science and Engineering
McMaster University, Hamilton, ON, Canada
Email: wahidd@mcmaste.ca

Elkafi Hassini
DeGroote School of Business
McMaster University, Hamilton, ON, Canada
Email: hassini@mcmaster.ca

Abstract

The correlation clustering problem identifies clusters in a set of objects when the qualitative information about objects' mutual similarities or dissimilarities is given in a signed network. This clustering problem has been studied in different scientific areas, including computer sciences, operations research, and social sciences. A plethora of applications, problem extensions, and solution approaches have resulted from these studies. This paper focuses on the cross-disciplinary evolution of this problem by analyzing the taxonomic and bibliometric developments during the 1992 to 2020 period. With the aim of enhancing cross-fertilization of knowledge, we present a unified discussion of the problem, including details of several mathematical formulations and solution approaches. Additionally, we analyze the literature gaps and propose some dominant research directions for possible future studies.

Keywords: correlation clustering; balance theory; structural balanced; signed network.

2.1 Introduction

Identifying partitions or clusters in a group of objects based on their mutual similarity and dissimilarity using qualitative information is a long-studied and multidisciplinary problem (Nelson; 1973; Rosch; 1977). Due to the recent adaptation of the mutual similarity/dissimilarity relation-based signed networks, our perception of real-world complex systems has increased significantly. In this network representation, any two objects are connected by a signed link representing mutual similarity (positive link)/dissimilarity (negative link). Forming communities or clustering is one of the most inherent structural properties observed in these networks. In these clusters, similar objects are grouped together. These similarities may come from the underlying complex behavioural pattern among the objects. Therefore, identifying and analyzing these communities help us understand the underlying behavioural pattern of the complex system. The *correlation clustering (CC)* problem is arguably the most natural formulation of identifying clusters in a complex system with information about mutual similarity/dissimilarity relationships (Bonchi et al.; 2022). In its simplest setting, the CC problem can be defined as in Definition 2.1 (Wirth; 2010).

Definition 2.1 *Consider $G = (V, E, s)$, a weighted signed network that represents the mutual similarity/dissimilarity relationship of a set of objects (nodes) $V = \{1, 2, ..., n\}$. Each link $(i, j) \in E$ is associated with a non-negative weight w_{ij} and is labeled by a signed function $s : E \to \{+, -\}$ to represent the mutual similarity (positive link)/dissimilarity (negative link). E^+ and E^- are the sets of all positive and negative (weighted) links in G such that $E^+ \cap E^- = \emptyset$ and $E^+ \cup E^- = E$. Let, $C[.]$ be a mapping of the objects to be clustered. That is, for any two objects $i, j \in V$, they are in the same cluster if and only if $C[i] = C[j]$. The goal is to cluster these objects so that similar objects are in the same clusters and dissimilar objects are in different clusters to the best possible extent. Therefore, the goal if the CC problem find a clustering C that minimizes*

$$\sum_{C[i]\neq C[j];(i,j)\in E^+} w_{ij} + \sum_{C[i]=C[j];(i,j)\in E^-} w_{ij}, \tag{2.1}$$

or, equivalently, maximizes

$$\sum_{C[i]=C[j];(i,j)\in E^+} w_{ij} + \sum_{C[i]\neq C[j];(i,j)\in E^-} w_{ij}. \tag{2.2}$$

Two types of cluster errors may arises in CC problem: negative-link errors, where negative links exist inside clustered groups and positive-link errors, where positive links exist between any two clustered groups. In Eq. (2.1), we minimize the sum of all positive-link and negative-link errors. Similarly, in Eq. (2.2), we maximize the sum of all positive-links inside clustered groups and negative-links between any two clustered groups. We illustrate an example of the CC problem in Fig. 2.1, where we have two clustered groups (*group 1* and *2*), and two clustering errors (minimized). The negative-link (dotted-red link) and positive-link (solid-black link) errors are created by the links (v_2, v_3) and (v_3, v_6), respectively.

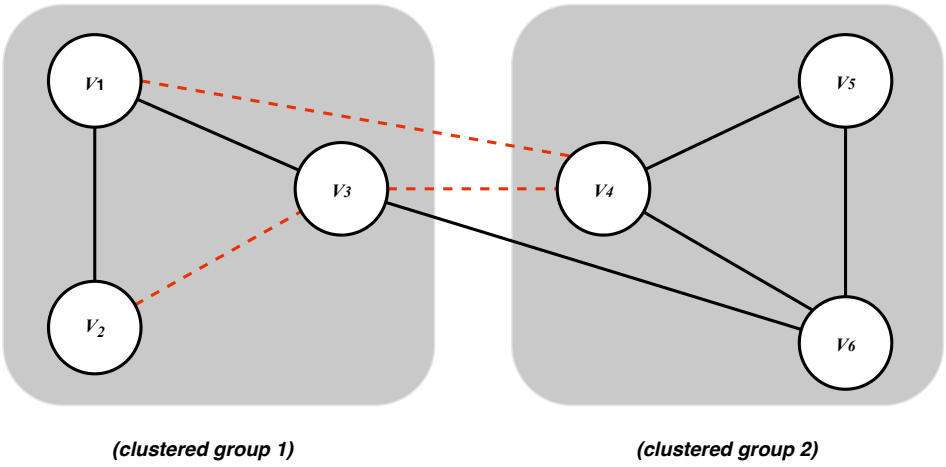


FIGURE 2.1: An example of the correlation clustering problem.

The field of clustering is a well-studied topic, and many clustering methods have been proposed in different areas. To understand where the CC problem fits in that field, we present a classification of clustering in Fig. 2.2. In general, clustering methods are unsupervised, including the CC problem. As shown in Fig. 2.2, the CC problem is a type of *Graph-based Clustering method*, a sub-class of *Partitioning-based Clustering*. This problem optimizes the quality functions Eq. (2.1) or Eq. (2.1) based on mutual similarity-dissimilarity. We also note that the CC and the well-known K-Means problems are classified as *Partitioning-based Clustering*. However, K-Means is labelled as *Distance-based Clustering*, whereas the CC problem is labelled as *Graph-based Clustering*.

The motivation for studying the CC problem comes from the broad generality that makes it applicable to a wide range of problems in different real-world scenarios. In particular, this problem is suitable for clustering objects when finding the feature vector is difficult to obtain (Bonchi et al.; 2022). In addition to identifying communities

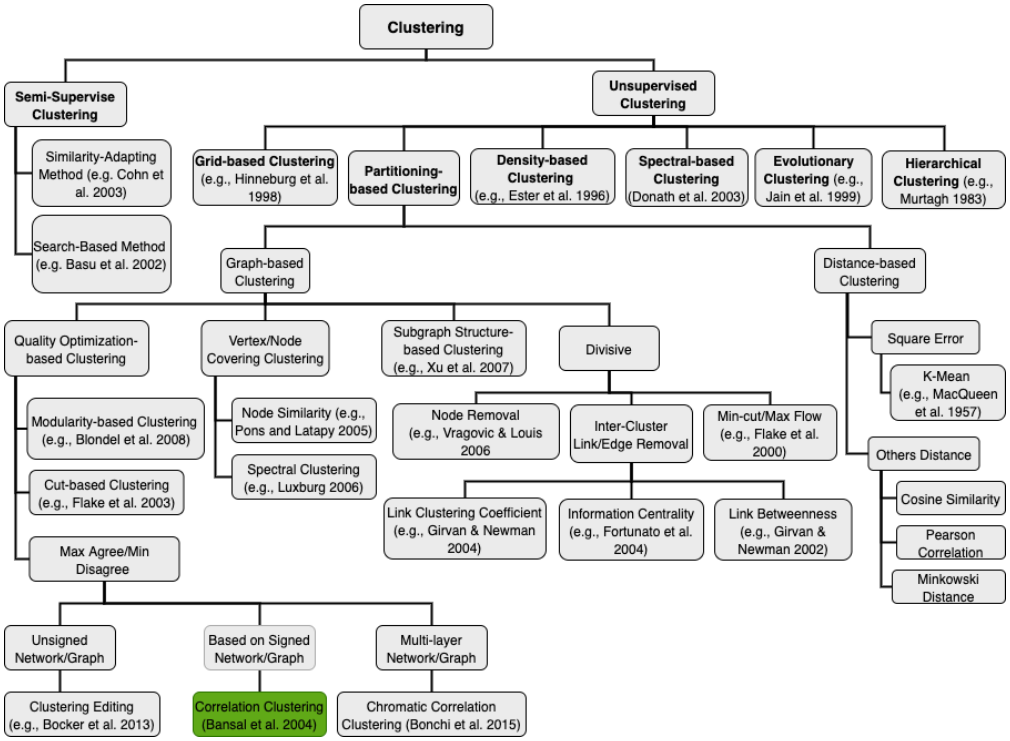


FIGURE 2.2: Overview of clustering classifications and the relative position of the CC problem. In addition to our studies on this topic, this figure is generated based on synthesized information from the following surveys: Jain et al. (1999); Grira et al. (2004); Schaeffer (2007); Kim (2009); Fortunato (2010); Nguyen et al. (2015); Pandove et al. (2018); Rokach (2009); Xu and Wunsch (2005); Bair (2013); Harenberg et al. (2014); Xu and Tian (2015); Chunaev (2020).

Other references in Fig. 2.2 are: Cohn et al. (2003); Basu et al. (2002); Hinneburg et al. (1998); Ester et al. (1996); Donath and Hoffman (2003); Jain et al. (1999); Murtagh (1983); Xu et al. (2007); Flake et al. (2004); Blondel et al. (2008); Pons and Latapy (2005); von Luxburg (2006); Vragović and Louis (2006); Flake et al. (2000); Newman (2004); Fortunato et al. (2004); Girvan and Newman (2002); MacQueen et al. (1967); Böcker and Baumbach (2013); Bansal et al. (2004); Bonchi et al. (2015)

.

(clusters) in real-world social networks, the CC problem can be a powerful tool for data mining such as data integration (Néda et al.; 2006; Cohen and Richman; 2002; Bonchi et al.; 2022) and agnostic learning (Néda et al.; 2009), aggregating multiple clusters

(Gionis et al.; 2007), and link classification (Cesa-Bianchi et al.; 2012). To the best of our knowledge, two survey papers: Il'ev et al. (Il'ev et al.; 2016) and Pandove et al. (Pandove et al.; 2018); and a book Bonchi et al. Bonchi et al. (2022) have considered the CC problem. These studies were mostly targeted to computer science reader. To enhance cross-fertilization of knowledge, in this paper, we present a unified discussion of the problem that targets operations research readers, including details of several mathematical formulations and solution approaches, and provide taxonomies for its applications, classes, and solution procedures. In addition, to reporting on recent developments in this field, this paper aims to present a bibliometric-based analysis to investigate collaborations, citation progressions, and dominant research topics. Based on our study, we observed significant citation gaps for the research articles that appear in different scientific communities, with the exception of Doreian and Mrvar (Doreian and Mrvar; 2009), Figueiredo and Moura (Figueiredo and Moura; 2013), Levorato et al. (Levorato et al.; 2015). This review will help bridge this gap and hopefully spur more interest in the field. Finally, we also used bibliometric studies to identify knowledge clusters and related future research venues.

2.2 Early Works on CC

The CC problem can be traced back to several independent studies in different areas. In the social balance context, Heider (1946) first described this problem to explain the theory of friendly and hostile relationships balance in a social group. According to Heider's balance theory, a systematic social balance can be achieved by putting people who share similar mutual friendly attitudes in the same social group and vice-versa. Heider was also the first to use the signed graph network (a.k.a., signed network) to represent a social group in which a node represents a person and a positive (negative) link represents the mutual friendly (hostile) attitude between two persons (Fig. 2.3). Based on Heider's balance theory, Cartwright and Harary (1956) and Davis (1967) formalized the definition of a structurally-balanced signed network to be that where clustered groups have only friendly attitudes (positive links) and hostile attitudes (negative links) exist only between clustered groups (Fig. 2.4).

Doreian and Mrvar (1996) was the first to formulate the problem as an optimization model to analyze the structural balance in a signed social network. Ben-Dor et al. (1999) introduced clustering based only on node similarity measures on a complete network, in which each pair of nodes are connected by a link. They applied it to cluster gene expression patterns. Chen et al. (2001) used the same model to reconstruct the phylogenetic

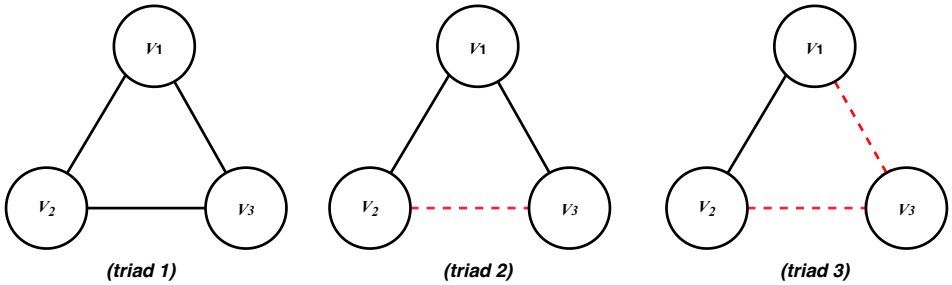


FIGURE 2.3: The network representation of Heider's balance theory using three signed networks (*triad 1*, *2*, and *3*) with mutually connected nodes (persons) v_1, v_2, and v_3. The black-solid line (positive link) represents the mutual friendly attitude between two nodes; similarly, the red-dotted line (negative link) represents the mutual hostile attitude between two nodes. Thus, according to Heider's balance theory, *triad 1* and *triad 3* are structurally balanced, and *triad 2* is imbalanced.

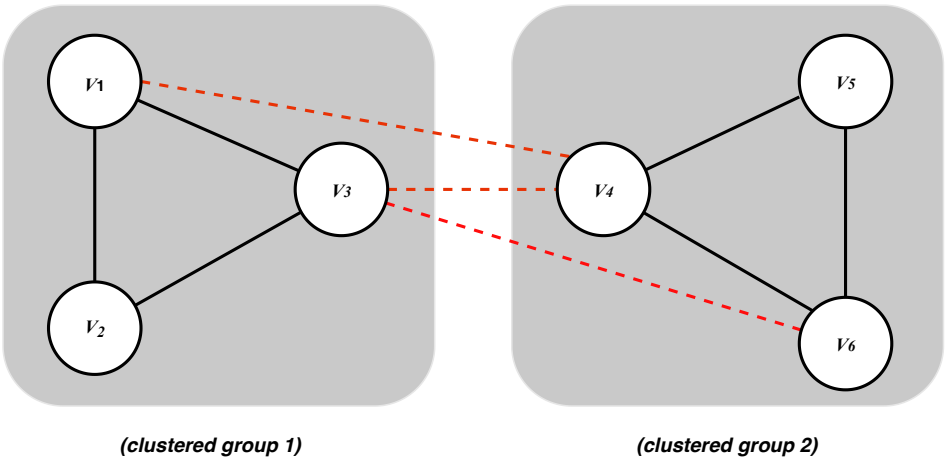


FIGURE 2.4: An example of a structurally balanced signed network with two clusters. In this state, positive links only exist inside clusters, and negative links exist only between two clusters.

tree. Later, Shamir et al. (2004) categorized this class of problems as *clustering editing* or *graph modification problems*. This clustering editing problem can be seen as a special case of Doreian and Mrvar (1996)'s node similarity and dissimilarity-based optimization problem. Doreian and Mrvar's optimization problem was unnoticed by the rest of the scientific communities for a long time.

On the other hand, in the data classification context, Zahn (1964) proposed the problem of finding an equivalence relationship that 'best approximates' a given symmetric relationship (which is equivalent to Eq. (2.1)). They also solved this problem as a special network class representing two- and three-level hierarchies. Régnier (1965) introduced a mathematical program for searching for the 'best approximate' from several given equivalence relationships. This problem is equivalent to the MaxAgree problem given in Section 2.3.1. Several studies have been done later on such equivalence relationship problems (Mirkin; 1974; Ambrosi; 1984; Barthelemy and Monjardet; 1981; Opitz and Schader; 1984; Marcotorchino and Michaud; 1981b,a).

In the business management context, Doyle (1992) was the first to introduce the concept of clustering objects in 1992 based on their mutual similarities and dissimilarities. However, in later years, we have not observed much attention to this problem in the business management field.

In 2004, the attention to this optimization problem independently started in computer science, especially in the machine learning community, when Bansal et al. (2004), inspired by applications in the area of document clustering, gave the name *Correlation Clustering (CC)* to this problem. At the same time, Shamir et al. (Shamir et al.; 2004) also independently introduced this problem. Bansal et al. (2004)'s work played a significant role in disseminating knowledge about this problem, evidenced by its leadership in citation counts.

2.3 Problem Variants and Complexity Analysis

2.3.1 CC Problem Variants

According to Definition 2.1, only positive links can exist inside a clustered group, and only negative links can exist between any two clustered groups. Based on this definition, a positive link can be considered as a *clustering agreement* if its corresponding end-nodes are in the same cluster. In contrast, it can be regarded as a *clustering disagreement* if its corresponding end-nodes are in different clusters. On the other hand, a negative link can be considered as a *clustering disagreement* if the end-nodes connected by this link are in the same cluster, whereas it can be regarded as a clustering agreement if the end-nodes are in different clusters. The following three variants of the CC problem have been used in the literature depending on the clustering agreement and disagreement.

For a given signed network in which positive and negative links, respectively, connect similar and dissimilar objects (nodes):

i *Minimum Disagreements* (MinDisAgree) *Problem*: The objective is to minimize clustering disagreements: the sum of all positive links between any two clusters and all negative links inside clusters.

ii *Maximum Agreements* (MaxAgree) *Problem*: The objective is to maximize clustering agreements: the sum of all positive links inside clusters and all negative links between any two clusters.

iii *Maximum Correlation* (MaxCorr) *Problem*: The objective is to maximize the correlation metric: the difference between the number of clustering agreements and clustering agreements.

In order to formally define the above three variants of the CC problem, let $G = (V, E, s)$ be a general weighted signed network where V and E are the node and link sets, respectively, and $|V| = n$. Each link $(i, j) \in E$ has a non-negative weight w_{ij}. Let every link (i, j) be labelled by a signed function $s : E \rightarrow \{+, -\}$, E^+ and E^- be the sets of all positive and negative (weighted) links in G such that $E^+ \cap E^- = \emptyset$ and $E^+ \cup E^- = E$. A node partition $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_k\}; k = 1, ..., m$, is a collection of k disjoint subsets (clusters) of nodes. Then the clustering disagreements due to the given partition P can be defined as follows in Eqs. (2.3) and (2.4):

$$D^+(\mathcal{P}) = \sum\{w_{ij} : (i, j) \in E^+;\ i \in \mathcal{P}_i, j \in \mathcal{P}_j, i \neq j\} \tag{2.3}$$

$$D^-(\mathcal{P}) = \sum\{w_{ij} : (i, j) \in E^-;\ i, j \in \mathcal{P}_i, i \neq j\} \tag{2.4}$$

Similarly, the clustering agreements due to the given partition $\mathcal{P}$ can be defined as follows in Eqs. (2.5) and (2.6):

$$A^+(\mathcal{P}) = \sum\{w_{ij} : (i, j) \in E^+;\ i, j \in \mathcal{P}_i, i \neq j\} \tag{2.5}$$

$$A^-(\mathcal{P}) = \sum\{w_{ij} : (i, j) \in E^-;\ i \in \mathcal{P}_i, j \in \mathcal{P}_j, i \neq j\} \tag{2.6}$$

Therefore, the total (weighted) clustering disagreements, clustering agreements, and correlation metric due to the partition $\mathcal{P}$ can be formulated, respectively, as follows:

$$f_D(\mathcal{P}) = D^+(\mathcal{P}) + D^-(\mathcal{P}) \tag{2.7}$$

$$f_A(\mathcal{P}) = A^+(\mathcal{P}) + A^-(\mathcal{P}) \tag{2.8}$$

$$f_C(\mathcal{P}) = f_A(\mathcal{P}) - f_D(\mathcal{P}) \tag{2.9}$$

According to the definitions, the quality of clustering for CC problem variants due to a given partition can be represented by either the total clustering disagreements, in Eq. (2.7), or by the total clustering agreements, in Eq. (2.8), or by the correlation metric, in Eq. (2.9). Therefore, we can formulate the MinDisAgree, MaxAgree, and MaxCorr problem variants based on the above quality functions as follows:

Problem 2.3.1 MinDisAgree
Input: A general weighted signed network $G = (V, E, s)$, where $|V| = n, s : E \to \{+, -\}$, and a function $f_D : \mathcal{P} \to \mathbb{N}_0$.
Task: Find an optimal node partition $\mathcal{P}^*$ such that, $f_D(\mathcal{P}^*) = \min_{\mathcal{P}} f_D(\mathcal{P})$

Problem 2.3.2 MaxAgree
Input: A general weighted signed network $G = (V, E, s)$, where $|V| = n, s : E \to \{+, -\}$, and a function $f_A : \mathcal{P} \to \mathbb{N}_0$.
Task: Find an optimal node partition $\mathcal{P}^*$ such that, $f_A(\mathcal{P}^*) = \max_{\mathcal{P}} f_D(\mathcal{P})$

Problem 2.3.3 MaxCorr
Input: A general weighted signed network $G = (V, E, s)$, where $|V| = n, s : E \to \{+, -\}$, and a function $f_C : \mathcal{P} \to \mathbb{N}_0$.
Task: Find an optimal node partition $\mathcal{P}^*$ such that, $f_C(\mathcal{P}^*) = \max_{\mathcal{P}}\{f_A(\mathcal{P} - f_D(\mathcal{P})\}$

Based on our literature review, we observed that almost all of the analysis and applications focus on the first two variants of correlation clustering: MinDisAgree and MaxAgree.

2.3.2 Complexity Analysis

Bansal et al. (2004) proved the NP-hardness of MinDisAgree, or equivalently MaxAgree, on unweighted signed networks (each link's weight is either $+1$ or -1) by reducing the problem to that of the Partition into Triangle GT11 (Garey and Johnson; 1979). In

this case, for a given unweighted signed network $G = (V, E, s)$, they introduced an underlying complete signed network on the same node-set (there exists either a positive or negative link between each pair of nodes) $G' = (V, E', s')$, in which the sign function labels the link $(i, j) \in E'$ as a positive link if $(i, j) \in E$; otherwise it labels it as a negative link, where $i, j \in V$. To prove the NP-hardness for weighted signed network cases, Bansal et al. (2004) introduced the reduction from the Multiway Cut problem (Avidor and Langberg; 2007) for the signed complete network. In this case, they transformed the input-weighted signed network $G = (V, E, s)$ to a complete signed network $G' = (V, E', s')$, by simply adding the link (i, j) in G' with weight $-\infty$, if $(i, j) \notin E$. Chen et al. (2001) studied the MinDisAgree problem in the context of the phylogenetic tree from the network representation of species evolutionary similarity. In this network, nodes are the species, and the link represents the evolutionary similarity between two species. The problem is to reconstruct the phylogenetic tree from a species evolutionary similarity network where the leaves of the phylogeny are labelled by nodes (i.e., species). In this network, any two nodes are connected by a link if and only if their corresponding leaves in the phylogeny are connected by a path of length at most k, where k is a predetermined proximity threshold. In this problem, the number of clustering disagreements inside a partition is equivalent to the number of dissimilarities among the species from the same phylogenetic tree root. Chen et al. (2001) also showed that it is NP-hard. Bansal et al. (2004), and Charikar et al. (2005) both showed that MinDisAgree on complete signed networks is APX-hard (i.e., it is NP-hard to approximate within some constant factor greater than 1), and it is at least as hard to approximate as the Multiway Cut problem. They also proved the existence of PTAS algorithms for the MaxAgree problem on complete singed networks. Later, Demaine et al. (2006) showed the APX-hardness on general weighted signed networks and proved that the integrality gap for MinDisAgree is $\Omega(\log n)$, where n is the number of nodes in the input network.

2.4 Taxonomy of the CC Problem and On-words

This section of the paper focuses on the taxonomic evolution of the CC problem. A graphical overview of the taxonomic evolution of this problem is given in Fig. 2.5. The different formulations of this problem (in yellow in Fig. 2.5) have already been discussed in Section 2.3.1. Solution approaches (in blue colour in Fig. 2.5) will be addressed in Section 2.4.1. CC with specific constraints (green colours in Fig. 2.5) will be discussed in Section 2.4.3. Other clustering problems (orange colours in Fig. 2.5), which are developed from the inspiration of the CC problem, will be briefly presented in Section 2.4.4. The

overview of the CC problem's applications in different areas (colours in Fig. 2.5) will be presented in Section 2.4.5. The well-known clustering problems (grey colours in Fig. 2.5, which can be reduced to the correlation clustering, will be addressed in Section 2.4.6. Finally, in Section 2.5.7, we will briefly discuss an exciting but very few-explored area of research: a comparison between well-known correlation clustering and machine learning algorithms.

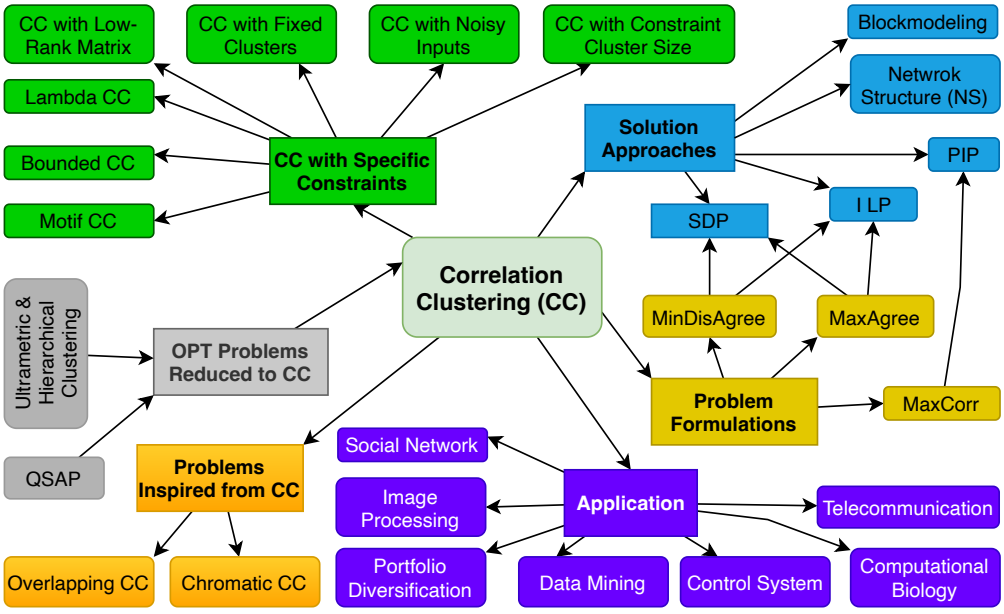


FIGURE 2.5: Taxonomy of correlation clustering. Note that the direction of the links represents the taxonomic evolution flow for this problem.

2.4.1 Solution Approaches Classification

Several classes of solution approaches appeared in the research literature. A brief overview of the solution approaches is given in Fig. 2.6. We can classify the solution approaches into four categories: Network-Structure (NS)-, Integer Linear Program (ILP)-, Semi-Definite Program (SDP)-, and Block Modelling- (BM) based techniques. Brief discussions on these solution approaches are given in the following subsections.

Network Structure (NS)-based Approach

The CC problem identifies communities in signed networks. Therefore, using the network's structural properties (e.g., node's neighbours, random walk, path, etc.) to design a solution algorithm is a very well-studied approach. For example, Ailon et al. (2008)

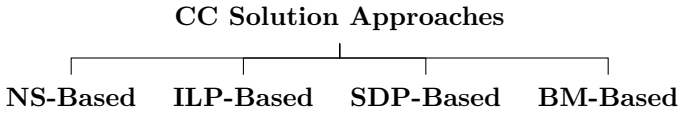


FIGURE 2.6: Solution approaches classification for the correlation clustering problem.

used random pivots and nearest neighbours of nodes to identify the best candidates for aggregating in a cluster. This study categorizes all the similar solution methods as the network structure-based approach.

Integer Linear Program (ILP)-based Approaches

The integer linear program-based solution methods for the CC problem's MINDISAGREE and MAXAGREE variants appeared in several studies (e.g., Charikar et al. (2005); Demaine et al. (2006); Demaine and Immorlica (2003); Wirth (2005)). To formally define the ILP formulations for MINDISAGREE (Problem 2.3.1) and MAXAGREE (Problem 2.3.2), for a given signed weighted network $G = (V, E, s)$ and $|V| = n$, consider a set of $\binom{n}{2}$ binary decision variables $\{x_{ij} : 1 \leq i < j <\leq n\}$ to represent the mutual cluster association of each distinct pair of nodes $i, j \in V$. For a given node partition P, we can define the binary variable x_{ij} as follows:

$$x_{ij} = \begin{cases} 0; & \text{if } i \text{ and } j \text{ are in the same cluster,} \\ 1; & \text{otherwise.} \end{cases} \tag{2.10}$$

Using this definition, $1-x_{ij} = 1$ if i and j are within a common cluster; and $1-x_{ij} = 0$ otherwise. Note that if $(i, j) \in E$, then the non-negative weight associated with the link (i, j) is w_{ij}; otherwise, it is 0. Thus, from Eq. (2.7), the total number of clustering disagreements due to the partition $\mathcal{P}$ is:

$$f_D(\mathcal{P}) = \sum_{(i,j)\in E^+} w_{ij}x_{ij} + \sum_{(i,j)\in E^-} w_{ij}(1 - x_{ij}). \tag{2.11}$$

Similarly, from Eq. (2.8), the total number of clustering disagreements due to the partition P is:

$$f_A(\mathcal{P}) = \sum_{(i,j)\in E^+} w_{ij}(1 - x_{ij}) + \sum_{(i,j)\in E^-} w_{ij}x_{ij}. \tag{2.12}$$

By using Eq. (2.11), the ILP formulation for MinDisAgree (Problem 2.1) can be written as follows:

$$\begin{aligned}
\min_{\mathcal{P}} \quad & \sum_{(i,j)\in E^+} w_{ij}x_{ij} + \sum_{(i,j)\in E^-} w_{ij}(1 - x_{ij}); \quad \forall i,j \in V && (2.13)\\
\text{s. t.:} \quad & x_{ij} + x_{jk} \geq x_{ik}; \quad \forall i,j,k \in V, && (2.14)\\
& x_{ij} = x_{ji}; \quad \forall i,j \in V, && (2.15)\\
& x_{ij} \in \{0,1\}; \quad \forall i,j \in V. && (2.16)
\end{aligned}$$

The inequality constraint, in Eq. (2.14), enforces that for any distinct nodes $i, j, k \in V$, if i and j are in a common cluster, then k is also in that cluster. This constraint is referred to as the triangle inequality constraint. The constraint, in Eq. (2.15), represents the undirected link property of the input network.

Similarly, the ILP formulation for MaxAgree (Problem 2.3.2) by using Eq. (2.12) can be written as follows:

$$\begin{aligned}
\max_{\mathcal{P}} \quad & \sum_{(i,j)\in E^+} w_{ij}(1 - x_{ij}) + \sum_{(i,j)\in E^-} w_{ij}x_{ij}; \quad \forall i,j \in V && (2.17)\\
\text{s. t.:} \quad & x_{ij} + x_{jk} \geq x_{ik}; \quad \forall i,j,k \in V, && (2.18)\\
& x_{ij} = x_{ji}; \quad \forall i,j \in V, && (2.19)\\
& x_{ij} \in \{0,1\}; \quad \forall i,j \in V. && (2.20)
\end{aligned}$$

As we mentioned above, CC is NP-hard; the exact problem based on the ILP formulation is hard to solve. Therefore, the ILP formulation-based solution methods depend on the relaxation of the integer constraints to generate an approximation or heuristic solution to the problem (Pandove et al.; 2018). The relaxed ILP versions for the MinDisAgree and MaxAgree can be formulated by replacing integer constraints in Eqs. (2.16) and (2.20) with the following constraint in Eq. (2.21). The ILP-based solution methodologies use different techniques, such as *region growing* (Demaine et al.;

2006) and *ultrametric distance* (Wahid; 2017), to obtain a solution from the relaxed solution of the ILPs.

$$0 \leq x_{ij} \leq 1. \tag{2.21}$$

The complexity of the MaxAgree is PTAS (Bansal et al.; 2004) and MinDisAgree is a 2.06-approximation (Chawla et al.; 2015) for unweighted signed networks. For general weighted signed network, the complexity is 0.7666-factor approximation for MaxAgree (Charikar and Wirth; 2004; Swamy; 2004) and $\mathcal{O}(\log n)$ for MinDisAgree (Charikar et al.; 2005; Demaine et al.; 2006). A brief overview of the results related to these two formulations can be seen in Table 2.2.

Semi-Definite Program (SDP)-based Approaches

Swamy (2004) and Wirth (2005) independently introduced the SDP relaxation-based solution method for the MaxAgree (Problem 2.3.2) version of the correlation clustering problem. Later, this relaxed SDP formulation was used in several solution algorithms and applications of correlation clustering studies (e.g., Abdelnasser et al. (2014); Ahn et al. (2015); Charikar et al. (2005); Giotis and Guruswami (2006); Mathieu and Schudy (2010); Zaw et al. (2017)). The SDP for MaxAgree addresses the scalability issue of the correlation problem by reducing the number of variables (Ahn et al.; 2015; Samal et al.; 2018). To formulate the SDP, for a given node partition solution $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_k\}; k = 1, ..., m$, on the general signed weighted network $G = (V, E)$, set a distinct basis associated with each cluster in $\mathcal{P}$. For every node i in a cluster, we set the unit vector v_i to be that basis vector. Therefore, the clustering agreements for a given partition $\mathcal{P}$ can be expressed as the dot product of the unit vectors associated with the nodes. Set $v_i.v_j = 1$, if i and j are within a common cluster; and $v_i.v_j = 0$ otherwise. Using this notation, the SDP relaxation for the MaxAgree can be written as follows:

$$\max_{\mathcal{P}} \quad \sum_{(i,j)\in E^+} w_{ij}(v_i.v_j) + \sum_{(i,j)\in E^-} w_{ij}(1 - v_i.v_j); \quad \forall i, j \in V \tag{2.22}$$

$$\text{s. t.:} \quad v_i.v_i = 1; \quad \forall i \in V, \tag{2.23}$$

$$v_i.v_j \geq 0; \quad \forall i, j \in V. \tag{2.24}$$

Ahn et al. (Ahn et al.; 2015) presented a polynomial-time, $\mathcal{O}(n.polylog(n))$-space approximation algorithm for the SDP relaxation. A brief overview on the results related to this formulation is in Table 2.2.

Polynomial Integer Program (PIP)-based Approaches

Bonizzoni et al. (Bonizzoni et al.; 2008) proposed a Polynomial Integer Problem (PIP) formulation for the MaxCorr (Problem 2.3.3) and proved the existence of a PTAS for this problem. To formulate the PIP for the MaxCorr, consider for a signed weighted network $G(V, E, s)$ with $|V| = n$, w_{ij}^{+} and w_{ij}^{-} are the positive and negative link weights corresponding to links $(i, j) \in E^{+}$ and $(i, j) \in E^{-}$, respectively. For a given node partition $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_k\}$, is a collection of k disjoint subsets (clusters) of nodes, set the value of the binary variable $x_{ij} = 1$, if and only if, the nodes i and j are in the same cluster; otherwise $x_{ij} = 0$. Consequently, define $\sum_t x_{it}x_{jt} = 1$, if and only if, i, j are in the same cluster; otherwise $\sum_t x_{it}x_{jt} = 0$. Therefore, the PIP formulation of the MaxCorr can be written as follows:

$$\max_{\mathcal{P}} \quad \sum_{(i,j)}, \left(w_{ij}^{+} \sum_{t} x_{it}x_{jt} + w_{ij}^{-} \left(1 - \sum_{t} x_{it}x_{jt} \right) \right) \tag{2.25}$$

$$\text{s. t.:} \quad \sum_{t} x_{it} = 1; \quad \text{for } 1 \leq i \leq n, , \tag{2.26}$$

$$x_{it} \in \{0, 1\}. \tag{2.27}$$

Blockmodeling (BM)-based Approach

The matrix-based blockmodeling-was first introduced by Lorrain and White (1971). Later, Batagelj (1997) formulated and analyzed the optimization version of this problem. Doreian and Krackhardt (2001) and first used the matrix-based Blockmodeling method to analyze the structural balance in signed social networks in 2001. Later in 2009, Doreian and Mrvar (Doreian and Mrvar; 2009) used this method to solve the MinDisAgree (Problem 2.3.1) version of the CC problem. Though BM is a flexible method for analyzing signed networks, in practice, this method is only feasible and efficient for small dense networks due to the limitation of handling large-scale matrices in the computational process (De Nooy et al.; 2018). BM method first appeared in the social sciences journals, and most of its applications remained within that area (e.g., Doreian (2008); Drummond et al. (2013); Figueiredo and Frota (2014)).

BM method uses matrices as the computational and result visualization tools. This method partitions the matrix representation of the given network into the initial matrix-partition of positive and negative blocks (clusters). For a given matrix-block partition, positive links in the positive block (grey-coloured) and negative links in the negative block (white-coloured) can be specified as clustering disagreements. The method moves the cells between blocks as a local optimization procedure. For a given network (Fig. 2.7), the matrix representation and initial partition of positive and negative blocks and the output of the Blockmodeling method are presented in Table 2.1.

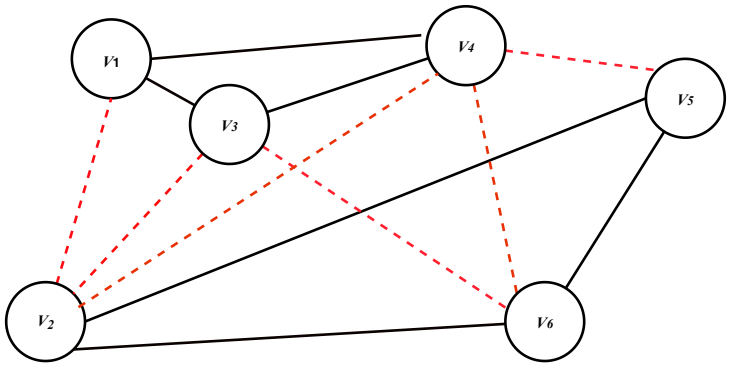


FIGURE 2.7: A signed network with nodes: $v_1, ..., v_6$. Positive links (black lines): (v_1, v_3), (v_1, v_3), (v_3, v_4), (v_2, v_5), (v_2, v_6), and (v_5, v_6). Negative links (red dotted lines): (v_1, v_2), (v_2, v_3), (v_2, v_4), (v_3, v_6), (v_4, v_5), and (v_4, v_6).

	v_1	v_2	v_3	v_4	v_5	v_6
v_1		-1	1	1		
v_2	-1		-1		1	1
v_3	1	-1	1			-1
v_4	1	-1	1		-1	-1
v_5		1		-1		1
v_6		1	-1	-1	1	

	v_1	v_3	v_4	v_2	v_5	v_6
v_1		1	1	-1		
v_3	1		1	-1		-1
v_4	1	1		-1	-1	-1
v_2	-1	-1	-1		1	1
v_5			-1	1		1
v_6		-1	-1	1	1	

TABLE 2.1: (a) The matrix representation of the above-signed network (in Fig. 2.7) with initial random blocks (clusters) separated by the different colours.; (b) Balanced positive and negative blocks. In this problem, the clustering error is zero. Positive and negative blocks are coloured as grey and white, respectively.

2.4.2 Algorithm Classifications

As we mentioned above, the CC problem is NP-hard; therefore, solving this problem in an exact manner is challenging. Over the years, several solution algorithms have been proposed to solve this problem. We can categorize the proposed solution algorithms into three main groups: approximation, heuristic, and parallel algorithms. An overview of the classification of these solution algorithms is given in Fig. 2.8. Brief discussions on these algorithm categorizes are presented in Subsections 2.4.2 to 2.4.2.

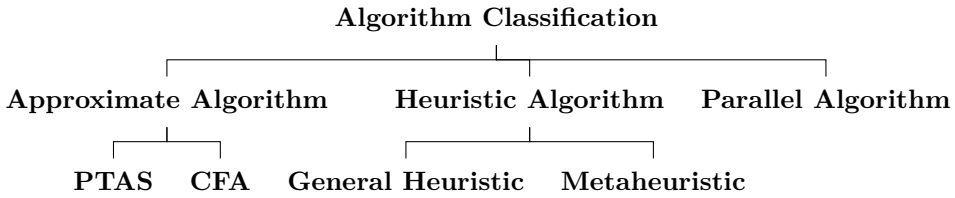


FIGURE 2.8: Solution algorithms classifications for the CC problem.

Approximation Algorithms

Approximation algorithms efficiently provide approximate solutions for the hard optimization problem with provable guarantees (Williamson and Shmoys; 2011). We classify the approximation algorithms for the CC problem into two subcategories: Constant Factor Approximation (CFA) and Polynomial-Time Approximation Scheme (PTAS). Table 2.2 lists a summary of the approximation-based solution algorithms for the correlation clustering problem. In this table, we can notice that about half of the approximation algorithms followed NS-based solution approaches, and equivalently, about half of these algorithms are designed for ± 1 weighted (a.k.a., unweighted) signed networks. Also, we can observe that most of the studies that design approximation algorithms focus on MINDISAGREE and MAXAGREE problem variants, and only two studies on the MAXCORR problem.

Heuristic Algorithms

Heuristic algorithms are generally designed to overcome the complexity limitation of the approximation algorithms and find a good-enough solution for the optimization problem. Several heuristic algorithms proposed in the literature for the CC problem. In order to keep this paper concise, we present a brief overview of these heuristic algorithms in Table 2.3. In this table, we can notice that all the heuristic's algorithms, except the

Reference	Problem Variant	Approach	Signed Network Type	Approximation	Complexity	Key Techniques and Points
Ben-Dor et al. (1999)	MaxCorr	NS	weighted, complete	PTAS	$\mathcal{O}((n^2 \log n)^c)$	Add or remove element to the cluster based on the affinity metric.
Bansal et al. (2004)	MinDisAgree	NS	± weighted	CFA	3-factor	Multicut partitioning triangle
	MaxAgree	NS	± weighted, complete	PTAS	-	
Swamy (2004)	MaxAgree	SDP	weighted	CFA	0.7666-factor	Random-hyperplae rounding procedure on the SDP relaxation
Charikar and Wirth (2004)	MaxCorr	SDP	weighted	PTAS	$\Omega(n)$	Rounding on the SDP relaxation
Charikar et al. (2005)	MinDisAgree	ILP	weighted, complete	CFA	4-factor	Region growing on LP relaxation
		ILP	weighted	PTAS		
	MaxAgree	SDP	weighted	CFA	0.7664-factor	
Demaine et al. (2006)	MinDisAgree	ILP	weighted	PTAS		Region growing on LP relaxation
Giotis and Guruswami (2006)	MinDisAgree	NS	±1 weighted	PTAS		The number of clusters k is fixed.
	MaxAgree	NS	±1 weighted	CFA	0.878-factor	
Coleman et al. (2008)	MaxCorr	NS	±1 weighted, complete	PTAS	2-factor	Local search and for 2 clusters (fixed)
Bonizzoni et al. (2008)	MaxCorr	PIP	±1 weighted	PTAS	2-factor	Max ratio between two scores (MinDisAgree and MaxAgree) is at most a constant.
Ailon et al. (2008)	MaxAgree	NS	±1 weighted	CFA	3-factor	Called KwikCluster, Random pivot & aggregation of positive neighbours.
Ailon et al. (2012)	MinDisAgree	NS	weighted	CFA	3-factor	Random pivot-based greedy procedure
Chawla et al. (2015)	MinDisAgree	ILP	±1 weighted, complete	CFA	2.06-factor	Link random pivot on LP relaxation
		ILP	± weighted, k-partite	CFA	3-factor	
			weighted, complete	CFA	1.5-factor	
Ahn et al. (2015)	MaxAgree	ILP, SDP	weighted	PTAS	$\Omega(n.polylog(n))$	Maintains an oracle in the data stream that uses a ball to cut links to create.
Fukunaga (2019)	MinDisAgree	ILP	weighted	PTAS	$\mathcal{O}(k \log n)$	Random pivoting and region growing on LP relaxation
Cambus et al. (2021)	MinDisAgree	NS	weighted, complete	PTAS	$\mathcal{O}(\log \Delta \log \log n)$	Parallel and randomize greedily

TABLE 2.2: Overview of the CC's approximation algorithms. Note that, NS-Network Structure, ILP-Integer Linear Program, SDP-Semi Definite Program, PIP-Polynomial Integer Program, CFA-Constant Factor Approximation, PTAS-Polynomial-Time Approximation Scheme.

Heuristic CC, use the NS-based solution approach; all of these identify communities in general weighted signed network. Note that, although the CC problem is defined on signed networks, many heuristic algorithms (e.g., Pivot Algorithm (Chierichetti et al.; 2014) and KwikCluster (Pan et al.; 2015)) are designed based on the heuristic clustering algorithm on unsigned networks. The interested reader can consult recent surveys on heuristic algorithms for unsigned networks, e.g., Chunaev (2020); Javed et al. (2018).

Parallel Algorithms

Due to the enormous growth of data and computational limitations on a single processor, designing efficient parallel algorithms for clustering can significantly improve the computational efficiency. Over the years, several parallel algorithms for CC were proposed in the literature. We present a brief overview of these parallel algorithms in Table 2.4. In this table, we can notice that almost half of the parallel algorithms are parallel-based approximation algorithms. In addition, a novel approach of solving the CC problem using parallel Bender's decomposition in which each node is associated with a Benders' subproblem is presented in (Keuper et al.; 2019).

2.4.3 CC with Specific Constraints

Several variants of the CC problem with additional constraints (e.g., cluster size, cluster numbers) were proposed over the years to identify communities in different types of data. We briefly discuss these variants of the correlation clustering problem with specific constraints in the following subsections.

CC with Fixed Clusters

The number of clusters in the CC problem is not fixed, and it arises naturally in the clustering process. However, several studies considered this problem with a fixed number of clusters. Giotis and Guruswami (2006) studied the CC problems with a fixed k number of clusters and proved that it is also NP-hard for $k \geq 2$. They referred MinDisAgree and MaxAgree versions of the CC problem with a fixed k number of clusters as MinDisAgree[k] and MaxAgree[k], respectively. They also provided PTAS algorithm with complexity $\mathcal{O}(\sqrt{\log n})$) for the MinDisAgree[$k$], and a 0.878-factor approximation for MinDisAgree[k] problems.

Similar approaches of CC with a fixed number of clusters can also be seen in Coleman et al. (2008), Ailon et al. (2012), and Ji et al. (2020). All of these papers focused on the case of two clusters. They presented local search-based 2- and 3-factor approximations

Algorithm Name & Reference	**Problem Variant**	**Approach**	**Signed Network Type**	**Max Net. Size** ($\|V\| = n$, $\|E\| = m$)	**Key Techniques and Points.**
CAST (Ben-Dor et al.; 1999)	MaxCorr	NS	general weighted, complete	n = 1500	Add/remove greedily nodes to a cluster based on an affinity metric.
COPAC (Achtert et al.; 2007)	MaxAgree	NS	general weighted	n=4610	Predict the best neighbours for clustering by using local search.
Genetic CC (Zhang et al.; 2008)	MaxCorr	NS	general weighted	n=800	Metaheuristic (Genetic) algorithm-based CC and uses fitness testing for a candidate solution.
GRASP (Drummond et al.; 2013)	MaxAgree	NS	general weighted	n=1000, m=859	Greedy randomized adaptive search-based metaheuristic algorithm
Restoring Wang and Li (2013)	MaxAgree	NS	general (un)weighted	n=1295, m=837865	Iteratively chooses two connected vertices and restores their neighbourhood.
RM and RMM Lingas et al. (2014)	MaxAgree	NS	unweighted	n=4000, m=24,522	Put all node into singleton clusters; if #of + link $>$ #of − link in $C_i \times C_j$; then merge.
ILS Metaheuristic Levorato et al. (2015, 2017)	MinDisAgree	NS	general weighted	n=100000	Iterated local search-based metaheuristic algorithm
Heuristic CC Wahid (2017)	MinDisAgree	ILP	general weighted	n=16356	Using ultrametric distance and rounding on the solution of ILP relaxation
Metric-Constrained Optimization Veldt et al. (2019)	MinDisAgree	ILP	general weighted	n=3068,m=119161	Developed a generalized metric-constrained linear and quadratic algorithm by proving CC is equivalent to ML metric-nearness

TABLE 2.3: Overview of the heuristic algorithms for the CC problem.

Algorithm Name & Reference Journal	**Problem Variant**	**Approach**	**Signed Network Type**	**Max Net. Size** ($\|V\| = n$, $\|E\| = m$)	**Complexity**	**Key Techniques and Points**
Parallel GRASP (Drummond et al.; 2013)	MaxAgree	NS	general weighted	n=1000, m=859	–	Greedy randomized adaptive search procedure with parallel schema
ParallelPivot Algorithm (Chierichetti et al.; 2014)	MaxAgree	NS	±1 weighted	n=41M; m=2.5B	3-factor apx runs $\mathcal{O}\left(\frac{1}{\epsilon}\log n \log \Delta\right)$	Random pivot and greedy based agglomerative approach
C4 Algorithm (Pan et al.; 2015)	MaxAgree	NS	±1 weighted	n=118,142,155; m=1,019,903,190	3-factor apx runs $\mathcal{O}\left(\frac{1}{\epsilon}\log n \log \Delta\right)$ rounds	Random pivot and greedy based agglomerative approach
ClusterWild Algorithm (Pan et al.; 2015)	MaxAgree	NS	±1 weighted	n=118,142,155; m=1,019,903,190	$3+\epsilon$-factor runs $\mathcal{O}\left(\frac{1}{\epsilon}\log n \log \Delta\right)$ rounds	Random pivot and greedy based agglomerative approach
CC in Data Stream (Ahn et al.; 2015)	MaxAgree; MinDisAgree	NS	±1 weighted	- -	PTAS runs $\mathcal{O}(n.poly(\log n)$ rounds	Dynamic network stream model with single & multi-pass streaming
ILS Parallel Metaheuristic (Levorato et al.; 2017)	MinDisAgree	NS	general weighted	n=10000	- -	Parallel iterated local search
(Aszalcós and Bakó; 2017)	MaxAgree	NS	general weighted	n=20000	- -	Parallel contraction and divided--conquer approach
Parallel Benders (Keuper et al.; 2019)	MinDisAgree	NS	general weighted	m=100000	–	Used Benders decomposition and cutting plane optimization
Massively Parallel Clustering (Cambus et al.; 2021)	MinDisAgree	NS	general weighted, complete	- -	3-approximation $\mathcal{O}(\log \Delta.poly(\log \log n))$ runs rounds	Parallel and randomize greedy based approach for MIS;

TABLE 2.4: Overview of parallel solution algorithms for the CC problem.

algorithms for MinDisAgree[k] and MaxAgree[k] on the general weighted signed network. Ailon et al. (Ailon et al.; 2018) presented a semi-supervised framework for approximate clustering (SSAC) for MinDisAgree[k] and MaxAgree[k] problems in which a learner answers a *same-cluster* query: *"given any two vertices, do they belong to the same optimal cluster?"* They showed that a $(1+\epsilon)$-approximation algorithm exists for any small ϵ with a polynomial running time in the input parameters k and $1/\epsilon$. Instead of fixing the node partition, Klein et al. (2015) proposed a version of correlation clustering with a fixed link partition and provided a 2-factor approximation solution.

CC-with Noisy Input

As we mentioned above, CC is an APX-hard problem. To avoid the APX-hardness of this problem, Mathieu and Schudy (2010) proposed a semi-random variant of CC in which the input network comes from a noisy system. According to their model, it starts with an arbitrary partition $\mathcal{P}$ where each pair of nodes is perturbed independently with probability p. Then, the updated partition $\mathcal{P}$ is generated by switching every perturbed pair of nodes between the clusters. This perturbation is controlled by an adversary process (i.e., decides whether to switch them). They provided a PTAS algorithm based on the SDP formulation for CC (given in Section 2.4.1). This PTAS produces $1 + \mathcal{O}\left(n - \frac{1}{6}\right)$ times the cost of the optimal clustering, if $p \leq \frac{1}{2} - n^{-\frac{1}{3}}$, where n is the number of nodes and p is the probability of link perturbation. Later, similar attempts of studying CC with noisy inputs can be seen in Rebagliati et al. (2013) with stochastic-based link labelling and in Makarychev et al. (2015) with noisy partial inputs. The CC with a noisy input model is helpful since, in practice, due to the complexity and expenses related to the process of collecting nodes' pairwise qualitative information (e.g., similarity or dissimilarity), the underlying network can be relatively sparse. This additional level of assumption (nodes perturbation with probability p) becomes useful for dividing data when no prior knowledge of the number of clusters exists.

CC with Constrained Cluster Size

Puleo and Milenkovic (2015) presented a CC version with additional constraints on the cluster size and extended this problem for more generalized weighted networks. They also provided an approximation algorithm with the ratio of $5 - 1/\tau$, where $\tau \in [1, \infty)$. To formulate this problem, they assumed a soft constraint to ensure that each cluster at most $K + 1$ nodes exist. They provided the following ILP formulation of this model:

$$\min_{x,y} \quad \left[\sum_{(i,j)\in E^+} w_{ij}x_{ij} + \sum_{(i,j)\in E^-} w_{ij}(1-x_{ij})\right] + \sum_{v\in V}\mu_v y_v, \tag{2.28}$$

$$\text{s. t.:} \quad x_{ij} + x_{jk} \geq x_{ik}; \quad \forall i,j,k \in V, \text{ and } i \neq j \neq k, \tag{2.29}$$

$$\sum_{i\neq j}(1-x_{ij}) \leq K + y_i; \quad \forall i \in V, \tag{2.30}$$

$$x_{ij} \in \{0,1\}; \quad \forall (i,j) \in E, \tag{2.31}$$

$$y_i \geq 0; \quad \forall i \in V. \tag{2.32}$$

The value of the binary variable $x_{ij} = 1$ indicates that the end nodes $i, j \in V$ are in different clusters; otherwise, $x_{ij} = 0$. The parameter μ_v is the penalty when more than $(K+1)$ nodes exist in a cluster. The constraint, in Eq. (2.30), represents the new restriction on the cluster size together with the penalty term $\sum_{v\in V}\mu_v y_v$ in Eq. (2.28) (objective function). The additional variable y_v represents the number of nodes by which the cluster containing v exceeds the size bound.

CC with Low-Rank Matrices

Since many real-world data are inherently low-dimensional, Veldt et al. (2017) presented a CC variant when the input network can be transformed into a low-rank matrix. To transform the CC problem in matrix formulation, consider, for a given weighted signed network $G(V,E,s)$, matrix $\mathbf{A}$ such that $A_{ij} = w_{ij}^+ - w_{ij}^-$, where w_{ij}^+ and w_{ij}^- are the weights corresponding to the positive and negative links. Then the matrix-based ILP formulation for the MaxAgree problem can be written as:

$$\min_{x,y} \quad -\sum_{i<j} A_{ij}x_{ij} \tag{2.33}$$

$$\text{s. t.:} \quad x_{ij} + x_{jk} \geq x_{ik}, \tag{2.34}$$

$$x_{ij} \in \{0,1\}; \quad \forall (i,j) \in E. \tag{2.35}$$

Binary variables are as defined as in Eq. (2.10). In Eq. (2.33), the inequality condition $(i < j)$ on $\mathbf{A}$ row-column indices (node indices in the signed network G) allows this formulation to avoid the equality condition given in Eq. (2.19). Consider an indicator vector $\mathbf{x}_i \in \{\mathbf{e}_1, ..., \mathbf{e}_n\}$ for each node i, where $\mathbf{e}_j$ is the j-th standard basis vector in $\mathbb{R}^n$.

This basis indicator vector for node i implies the cluster association of this node. By using the substitution $x_{ij} = 1 - \mathbf{x}_i^T \mathbf{x}_j$ in the above ILP formulation and by dropping the constant term in the objective, the matrix-based CC problem becomes:

$$\min_{x,y} \quad -\sum_{i<j} A_{ij} \mathbf{x}_i^T \mathbf{x}_j i^T \tag{2.36}$$

$$\text{s. t.:} \quad \mathbf{x}_i \in \{\mathbf{e}_i, ..., \mathbf{e}_n\}; \quad \forall i = 1, ..., n. \tag{2.37}$$

CC can be solved in polynomial time when a positive semi-definite low-rank matrix represents the node similarity (Veldt et al.; 2017). But the problem remains NP-hard if the underlying matrix has at least one negative eigenvalue.

Lambda Correlation Clustering

Lambda Correlation Clustering (LambdaCC), proposed by Veldt et al. (2017) and Gleich et al. (2018), is a special case of the CC problem with a condition for weighted links to identify communities in ± 1 weighted signed networks. In LambdaCC, $(w_{ij}^+, w_{ij}^-) \in (1-\lambda, 0), (0, \lambda)$, where $\lambda \in (0, 1)$ is a user-chosen parameter, and w_{ij}^+ and w_{ij}^- are the positive and negative link weights, respectively. By using this condition for the link weight, the objective function for the LambdCC is given by:

$$\min_{\mathcal{P}} \quad \sum_{(i,j)\in E^+} (1 - \lambda w_{ij}) x_{ij} + \sum_{(i,j)\in E^-} \lambda w_{ij} (1 - x_{ij}); \quad \forall i, j \in V. \tag{2.38}$$

The λ parameter allows tuning the impact of positive and negative links in the clustering process. LambdaCC's constraints are the same as CC's constraints. This generalized clustering framework can also be used to detect communities in unsigned unweighted networks by treating the non-link, i.e. $(i, j) \notin E$, weighted as λ (Gleich et al.; 2018). Veldt et al. (Veldt et al.; 2017) showed that LambdaCC is NP-hard and provided 3 and 2-factor approximations based on the LP rounding for the special cases $\lambda > 1/2$, and $\lambda > |E|/((1 + |E|))$, respectively.

Motif Correlation Clustering

Li et al. (2017) presented a higher-order generalization of CC called Motif Correlation

Clustering (MotifCC) to identify communities in unweighted completed networks based on higher-order motif patterns shared among nodes. An LP relaxation formulation for MotifCC was also presented in Li et al. (2017). Similar attempts of MotifCC on unweighted completed networks can be seen in Veldt et al. (2017) and Gleich et al. (2018). Recently, Hua et al. (2021) proposed a MotifCC based on the star structure motif. They also constructed a new ILP formulation based on cycle inequalities to perform the local search with final clustering results. In MotifCC, the clustering disagreement defines the cost of placing ε number of similar nodes into two clusters. In contrast, in CC, clustering disagreement counts the cost of two similar nodes (positive link) between two clusters.

To define the MotifCC, for a given complete unweighted network $G(V, E)$ with n nodes, let E_k denote the set of all k-tuples of nodes in G. For each $\varepsilon \in E_k, w_\varepsilon^+$ and w_ε^- are positive and negative weights associated with this tuple ε. The cost of placing at least one pair of nodes from ε in different clusters is w_ε^+ and the cost of placing all nodes from ε in one cluster is w_ε^-. For a give node partition $\mathcal{P} = \mathcal{P}_1, ..., \mathcal{P}_k$, the ILP formulation for can be expressed as follows in Eqs. (2.39) – (2.44):

$$\min_{x,y} \quad \sum_{\varepsilon \in E} w_\varepsilon^+ x_\varepsilon + w_\varepsilon^-(1 - x_\varepsilon), \tag{2.39}$$

$$\text{s. t.:} \quad x_{ij} + x_{jk} \geq x_{ik}; \quad \forall i, j, k \in V, \tag{2.40}$$

$$x_{ij} \in \{0, 1\}; \quad \forall i < j, \tag{2.41}$$

$$x_i \geq x_\varepsilon; \quad \forall i, j \in \varepsilon, \tag{2.42}$$

$$(k-1)x_\varepsilon \leq \sum_{i,j \in \varepsilon} x_{ij}; \quad \forall \varepsilon \in E_k, \tag{2.43}$$

$$x_\varepsilon \in \{0, 1\}; \quad \forall \varepsilon \in E_k. \tag{2.44}$$

where $x_{ij} = 1$ if i, j are in separated clusters. In Eq. (2.42), the inequality constraint ensures that if any two nodes $i, j \in \varepsilon$ are separated, then the k-tuple is split (i.e., $x_\varepsilon = 1$). Similarly, the constraints given in Eq. (2.43), ensure that if all pairs of nodes in ε are in the same cluster, then $x_\varepsilon = 0$.

Li et al. (2017) proved that MotifCC is NP-complete and provided a 9-approximation algorithm for this problem when the positive and negative weights associated with each tuple $\varepsilon \in E_k$ (of size 2 or 3) satisfy the following condition:

$$w_{\varepsilon}^{+} + w_{\varepsilon}^{-} = 1. \tag{2.45}$$

Gleich et al. (2018) gave a $4(k-1)$-approximation, and Fukunaga (2019) provided a $\mathcal{O}(k \log n)$ approximation for the general weighted network by using an LP rounding approach.

Bounded Correlation Clustering

Geerts and Ndindi (2016) proposed another variant of CC, called *Bounded Correlation Clustering (BCC)*, with additional constraints for link insertion and deletion in the clustering process. The link insertion process can be interpreted as putting negative links inside a cluster. Similarly, the link deletion process can be interpreted as placing positive links between clusters. Therefore, in BCC, there exists an acceptable bound for link insertion or deletion. Any link beyond this bound will be denied being deleted or inserted. Geerts and Ndindi (2016) showed that the BCC problem is NP-hard and then proposed an LP relaxation-based approximation algorithm.

2.4.4 Clustering Problems Inspired from the CC Problem

Several clustering problems have been designed based on the CC framework over the years due to its paradigm flexibility and applicability. Below we describe two such clustering problems in the following subsections.

Overlapping Correlation Clustering

Bonchi et al. (2013) proposed a clustering framework, inspired by CC, called the *Overlapping Correlation Clustering (OCC)*, in which the partitioned objects may overlap between clusters. They also proved that OCC is NP-hard and provided an approximation solution. Later, Andrade et al. (2014) presented an evolutionary algorithm, and Chagas et al. (2019) proposed a hybrid heuristic algorithm for the OCC problem.

Chromatic Clustering

Bonchi et al. (2015) proposed, a novel clustering framework called Chromatic Clustering to generalize the CC problem. It used different colours to include more than two types of qualitative link relations (rather than just the positive and negative). Bonchi et al. (2015) also established the NP-hardness of this problem. This clustering framework also appeared in several studies as multiplex or multi-layer clustering such as: Hmimida

and Kanawati (2015); Huang and Wang (2016); Lancichinetti and Fortunato (2012); Mondragon et al. (2018).

2.4.5 Optimization Problems Reduced to CC

The CC problem is a class of optimization problems that can be reduced from other well-known problems. This section focuses on those well-known optimization problems from which CC can be derived as a special case.

Ultrametric-based Hierarchical Clustering

Hierarchical Clustering (HC) (Murtagh; 1983) creates a hierarchy of clusters based on the similarity measure (or distance) of objects. Ultrametric-based Hierarchical Clustering (UHC) uses the ultrametric distance function in the process of clustering (Roy and Pokutta; 2017). A distance function $u : V \times V \rightarrow \mathbb{R}_0^+$ can be defined as ultrametric if: (Wahid; 2017)

$$\max\{u_{ij}, u_{jk} \geq u_{ik}\}; \text{ for all } i, j, k \in V, \tag{2.46}$$

where u_{ij} is the similarity distance between elements i and j. Let U be the distance matrix that satisfies Eq. (2.46). We can call U the *Ultrametric Matrix*. Therefore, for an input similarity distance matrix $\mathbf{X}$ with $x : V \times V \rightarrow \mathbb{R}_0^+$, the UHC can be defined as follows:

Problem 2.4.1 UltrametricHC
***Input*:** A distance matrix $\mathbf{X}$ with $x : V \times V \rightarrow \mathbb{R}_0^+$.
***Task*:** Find a Ultrametric-matrix U with $u : V \times V \rightarrow \mathbb{R}_0^+$ in the minimum distance (cost).

By imposing the binary condition on the distance function as $u : V \times V \rightarrow \{0, 1\}$, the optimality condition, in Eq. (2.46), can be redefined as follows:

$$u_{ij} + u_{jk} \geq u_{ik}. \tag{2.47}$$

We can call this new distance function the *0-1-ultrametric* (Roy and Pokutta; 2017; Wahid; 2017). Eq. (2.46) is equivalent to the triangle inequality constraint, in Eq. (2.14),

in the ILP formulation of the CC problem. Therefore, we can define a distance matrix U as the *0-1-ultrametric-matrix* if each element in U satisfies the inequality condition in Eq. (2.47) (Wahid; 2017). By using this $0-1$-ultrametric-matrix, we can redefine the MinDisAgree (Problem 2.3.1) as the following $0-1$ *Ultrametric-based Hierarchical Clustering* problem:

Problem 2.4.2 0-1-UltrametricHC
***Input*:** A distance matrix $\mathbf{X}$ with $x : V \times V \rightarrow [0, 1]$.
***Task*:** Find a 0-1-Ultrametric-matrix U with $u : V \times V \rightarrow \{0, 1\}$ in the minimum distance (cost).

Here, $\mathbf{X}$ is the matrix representation for the input network. The output matrix in (Problem 2.4.2) represents a set of disjoint clusters.

Quadratic Semi-Assignment Problem

QSAP is a well-known optimization problem that arises in many practical applications. Bonizzoni et al. (2008) presented a polynomial integer problem (PIP) formulation for MaxCorr (Problem 2.3.3) (briefly discussed in Section 2.4.1). The PIP formulation for this problem, given in Eqs. (2.25)-(2.27) is, in fact, a modified version of QSAP.

2.4.6 Applications

The applications of the CC problem appear in different scientific areas, such as for identifying communities in social networks (Doreian and Mrvar; 1996) and gene expression pattern recognition Ben-Dor et al. (1999). In Table 2.5, based on our extensive search in the literature, we present a list of references for different applications of the CC problem from the literature. Except for three studies (Bhattacharya and De; 2008; Sumi and Neda; 2008; Wei et al.; 2018); all other studies occurred in the last decade, which is a testimony of a growing interest in the application of CC. We also observed that most applications are in social networks and image processing.

2.5 Bibliometric Analysis

Bibliometric analysis is a descriptive analytics tool that helps researchers to investigate and visualize a large number of academic publications from the research literature. This analysis includes using networks to gain insights into authors' interactions (Mimno et al.;

Application Area	**Number**	**References Journals**
Community Identification in Social Network	13	Akorli et al. (2019); Bakó (2018); Barik et al. (2018); Belyaeva et al. (2017); Bessonov et al. (2013); Bhattacharya and De (2008, 2010); Joglekar (2014); Krasowski et al. (2017); Sumi and Neda (2008); Vassy et al. (2017); Wei et al. (2018); Zhang et al. (2014)
Image Processing	11	Alush and Goldberger (2012); Firman et al. (2013); Kappes et al. (2016); Kim et al. (2012); López-Sastre et al. (2011); Marra et al. (2016, 2017); Mehta et al. (2016); Solera and Calderara (2013); Yarkony et al. (2012); Zhu and Cao (2011)
Data Mining	5	Aszalós and Mihálydeák (2015); Slaoui et al. (2018); dong Xie et al. (2015); Zhao, Xiong, Yu, Zhang and Zhao (2016); Zhao, Xiong, Zhang, Yue and Yang (2016)
Computational Biology	3	Albin et al. (2011); Papenhausen et al. (2013); Wang et al. (2013)
Telecommunication	3	Abdelnasser et al. (2014); Maatouk et al. (2018); Nga et al. (2016)
Portfolio Diversification	3	Galagedera (2013);Isogai (2014); Zhan et al. (2015)
Control Systems	2	Albin et al. (2011); Wang et al. (2013)

TABLE 2.5: Application of the correlation clustering in different scientific areas.

2006), understand the chronological evolution of research interests (Morris et al.; 2002), and cluster the studies' research topics (Mogee; 1991). It also helps the researchers ascertain the existing research gaps and the hidden patterns of knowledge structure in the areas of study and identify possible future research paths (Daim et al.; 2006). Recently it used to assess the potential of interdisciplinary research in higher education journals (Wahid et al.; 2022). Such an analysis is becoming almost necessary, especially when we want to see how fast the literature grows (Hu et al.; 2020). In the following subsections, we present different aspects of the bibliometric analysis for the literature CC.

2.5.1 Initial Data Collection and Refinement

The publication data was collected from Web-of-Science (WoS), an online-based citation indexing website. In the first step of the data collection, we collected journal articles covering the topic of correlation clustering. In the research literature, correlation and clustering are very popular keywords and are overly used in different disciplines to explain various topics that are not necessarily related to our intended topic. To avoid publications that are not related to our topic, we used the keyword "correlation clustering" to search in the WoS. After the initial search, we collected 230 articles. We recognized that some publications are unrelated to the CC problem by inspecting the collected publications. Some of these eliminated publications are related to other uses of "correlation clustering" such as in the area of mathematics, where "correlation clustering" refers to generalized subspace clustering in the data space (Zimek; 2009). To identify these non-relevant studies, we decided to investigate each article individually to decide whether it is related to our investigated topic. Finally, this investigating and removing process yielded 158 articles.

We have already mentioned that the CC was first proposed by Doreian and Mrvar (1996) in the social science area under a different name and as a criterion to analyze the structural balance in signed networks. But the attention to this problem was started in 2004, right after the publications of the paper by Bansal et al. (2004) in the area of machine learning. In the following years, we observed a clear gap in the mutual citation among the articles from these two fields. We noticed that articles published on CC in social science mainly cite Doreian and Mrvar (1996). In contrast, computer science articles on this topic cited Bansal et al. (2004). In our observation, we founded that only a few articles crossed this citation gap over the years. Therefore, to include the articles from both scientific areas (social and computer sciences), we decided to collect all of the articles corresponding to both articles' citations from the WoS database. This

Steps	**Data Sets**		
	#of articles by keyword-search "correlation clustering."	#of articles that cited Doreian and Mrvar (1996)	#of articles cite Bansal et al. (2004)
Initial Search (Sept 21, 2020)	230	150	412
Refining: Delete irrelevant and duplicate articles	158	107	309

TABLE 2.6: Number of articles in the initial data collection and the refining steps.

decision resulted in 150 from Doreian and Mrvar (1996)'s citations and 412 articles from Bansal et al. (2004)'s citations. These three sources of collected articles were further investigated to eliminate any duplicate studies. As shown in Table 2.6, this resulted in 158, 107, and 309 studies in each of the collected data sets, respectively, for a total of 574 studies.

2.5.2 Data Analysis Tools

We used an open-source R-package called Bibliometrix (Aria and Cuccurullo; 2017) for the bibliometric data analysis. This package provides a complete set of tools for conducting comprehensive bibliometric research and scientific mappings such as influential authors, collaborations, and article citation analyses. For network visualization and community detection, we used Gephi (Bastian et al.; 2009) since it has the flexibility and ability to perform further clustering analysis on large-scale network data.

2.5.3 Overview Data Statistics

Fig. 2.9 presents the yearly distribution of publications from 1992, when Doyle (1992) published his work on multiple correlation clustering in social networks, to September 2020. From this figure, though the CC problem was introduced in the early 1990s, it only gained significant interest in the literature after 2004 when Bansal et al. (2004) independently introduced this problem in the computer science community.

To get an idea about the research communities that focused on the CC problem, we present the top 20 journals that published articles on this topic in Table 2.7. In total, 133 articles were published in 20 sources, representing about 26% of the total number of reviewed works. This table can give us a notion about the familiarity of the

SN	Source Names	Articles Numbers	Scientific Communities
1	Social Network	21	SS
2	Physica A-Statistical Mechanics and Its Application	11	CS, BS, SS
3	IEEE Transactions on Pattern Analysis and Machine Intelligence	10	CS
4	Theoretical Computer Science	10	CS
5	Algorithmica	8	CS (OR)
6	IEEE Transactions on Knowledge and Data Engineering	8	CS
7	Journal of Combinatorial Optimization	8	CS (OR)
8	Journal of Computer and System Sciences	6	CS
9	Journal of Mathematical Sociology	6	CS, SS
10	Discrete Applied Mathematics	5	CS (OR)
11	Journal of Machine Learning Research	5	CS
12	IEEE Conference on Computer Vision and Pattern Recognition	4	CS
13	IEEE International Conference on Computer Vision	4	CS
14	Artificial Intelligence	4	CS
15	Bioinformatics	4	BS
16	Information Processing Letters	4	CS
17	Pattern Recognition	4	CS
18	Physical Review E	4	CS, BS, SS
19	Scientific Reports	4	CS, BS, SS
20	IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)	3	CS
Total		133	

TABLE 2.7: Top 20 journal and conference proceeding with the corresponding number of published articles on CC over the years (1999-2020). Note that, CS (OR) represents the field of operations research but in the wider picture it belongs to the computer science field.

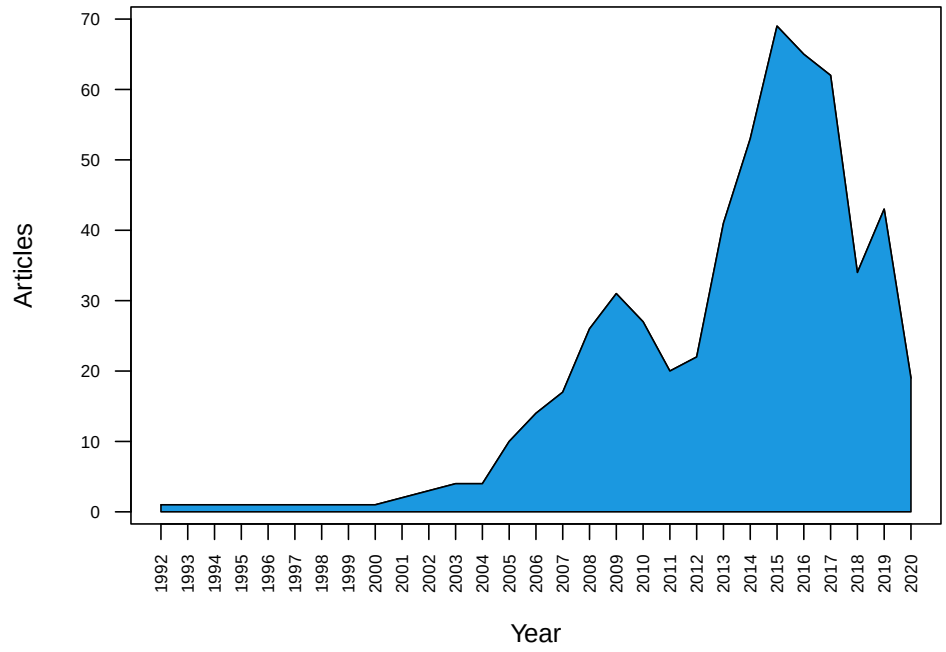


FIGURE 2.9: Yearly distribution of the published articles over the study period

CC problem in different scientific communities. The publication sources (journals and conference proceedings) can be classified into three broader scientific fields: *Computer Science (CS)*, *Social Science (SS)*, *Biological Science (BS)*. We can observe from Table 2.7 that the CC problem received the most attention from the computer and social science communities. This problem received the subsequent most attention in biological science since we can see four articles published in Bioinformatics, a Biological Science journal. However, some may argue that Bioinformatics journal overlaps between the Computer Science and Biological Science communities. Furthermore, some journals in this list overlap among all the above three scientific communities. We can also notice that the optimization problem is significantly underrepresented in the Operations Research (OR) community. For example, eight articles published in the Journal of Combinatorial Optimization may relate to the OR. Moreover, in the broader sense, the articles published in Algorithmica (8 articles) and Discrete Applied Mathematics (5 articles) may also be classified in the field of OR.

Author's Name	**PageRank Score**	**Discipline Associationn**
Andres B.	0.00545	Information Science
Guo J.	0.00389	Computer Science
Chen J.	0.00359	Computer Science
Li Y.	0.00343	Information Science
Ailon N.	0.00342	Computer Science
Doreian P.	0.00300	Sociology
Hamprecht F.A.	0.00294	Computer Science
Li J.	0.00293	Computer Science
Komusiewicz C.	0.00278	Information Science
Liu X.	0.00273	Computer Science
Liu J.	0.00262	Computer Science
Niedermeier R.	0.00261	Information Science
Koethe U.	0.00251	Computer Science
Aszalos L.	0.00242	Information Science
Mrvar A.	0.00238	Social Science
Hou J.	0.00229	Bioinformatics
Wu J.	0.00225	Mathematics
Zhang L.	0.00220	Economy
Figueiredo R.	0.00219	Information Science
Steinley D.	0.00217	Psychology

TABLE 2.8: Top 20 influential authors according to their corresponding PageRank scores in the author's collaboration network.

2.5.4 Authors Collaboration Clusters

The collaboration tendency of an author shows a positive correlation with the article citations (Hsu and Huang; 2011) and the corresponding author's ranking (Abramo et al.; 2019). Each node represents an author in a collaboration network, and each weighted edge represents the number of articles co-authored by two authors. We formulated a collaboration network of 990 nodes and 1891 edges from our collected publication data. PageRank, introduced by Brin and Page (Brin and Page; 1998), is a metric to evaluate the priority of a node in a network. We applied the PageRank analysis on the collaboration network and presented the top 20 authors list according to their corresponding PageRank scores in Table 2.8. We also investigated the institutional discipline association corresponding to these authors shown in Table 2.8. We noticed that the majority of these authors are associated with computer science-related disciplines (e.g., computer science and information). Only three authors (Doreian P., Mrvar A., and Steinley D.) belong to social science-related disciplines (e.g., sociology and psychology). In this list, there exists only one author from each of the following disciplines: mathematics (Wu J.), bioinformatics (Hou J.), and Economy (Zhang L.). This observation supports our previous argument that CC gets comparatively more attention in the computer science community.

To get insight into the influential authors' community structures, we use the modularity algorithm, proposed in Blondel et al. (2008) and implemented in Gephi, on the authors' collaboration network. Fig. 2.10 exhibits the ten largest authors' communities in the collaboration network. In combination, there are 306 nodes (i.e., approximately 30% of the whole network) in the ten largest communities. The node size in this figure is scaled with the corresponding weighted degree. Table 2.9 presents the top 3 authors, according to the weighted degree, from each of the ten cluster communities and their published articles' research focus. We observe in this table that, in our collected data, the research focuses on the published articles by the authors from 10 identified clusters extended from algorithm analysis to application in different sectors such as community detection in social and biological networks and image segmentation.

2.5.5 Direct Citation Analysis

Direct citation network, introduced in Garfield (2004), is a historical network map of most relevant articles' direct citations drawn from a bibliometric collection. It helps to trace the historical development of research-based innovations and the evolution of

Cluster-ID & Top 3 Authors (according to the degree)	**Published Articles Research Focuses**
Community # 1: Guo J., Komusiewicz C., Niedermeier R.	Optimization algorithm and complexity analysis.
Community # 2: Liu X., Li J., Zhang Y.	Evolutionary algorithms and community detection in social and biological networks.
Community # 3: Li Y., Chen J., Wu J.	Chromatic correlation clustering.
Community # 4: Ailon N., Charikar M., Wirth A.	Integer and semidefinite programming formulation and solution algorithms.
Community # 5: Andres B, Hamprecht FA, Koethe U	Application in image segmentation
Community # 6: Gong M., Cai Q., Chen Z.	Application in wireless sensors and complex networks.
Community # 7: Kindler G., Strauss K., Makarychev K.	Solution algorithms and application in large-scale biological networks.
Community # 8: Zhang C., Zhao Q., Yarkony J.	Solution algorithms and application in complex and biological networks.
Community # 9: Chawla S., Bansal N., Blum A.	Approximate and heuristic Algorithm for correlation clustering.
Community # 10: Rudroff F., Oliveira A. P., Dimopoulous S.	Application in biological networks.

TABLE 2.9: Top 3 authors from 10 largest identified clusters communities from the collaboration network and their corresponding articles research focuses.

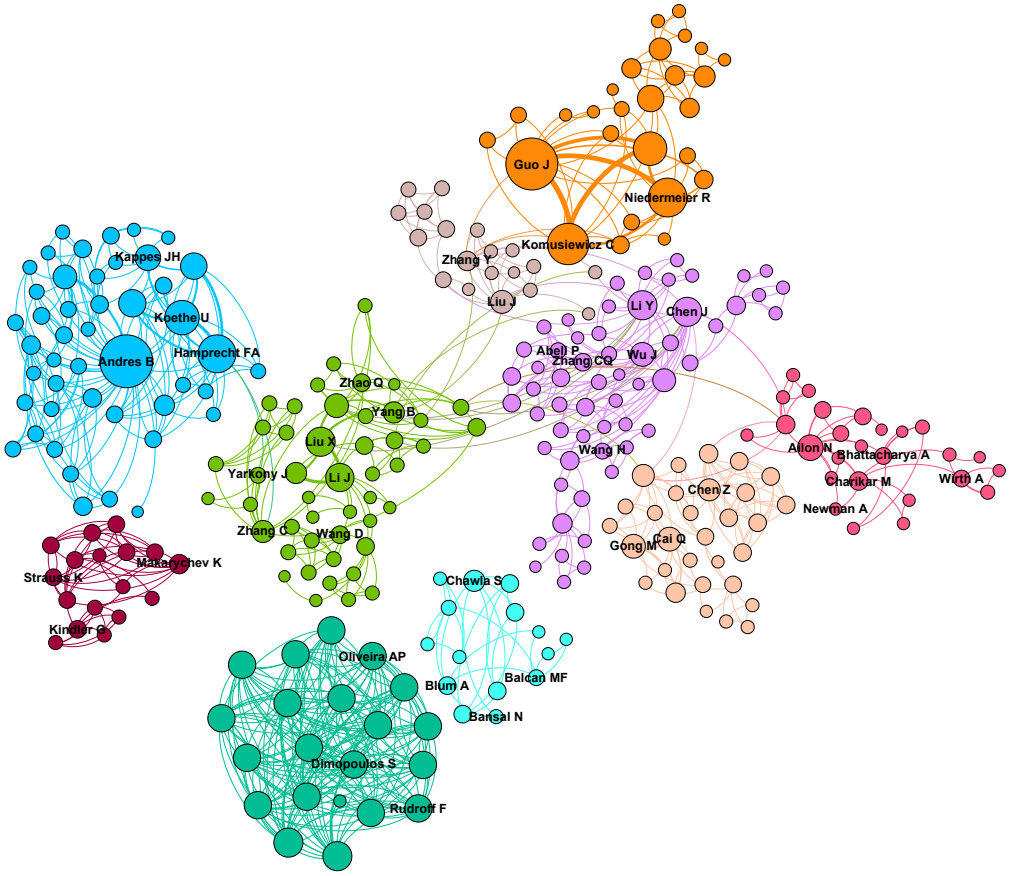


FIGURE 2.10: Top 10 author's clusters communities in the authors' collaboration network. The node size is scaled with the weighted degree.

coherent knowledge structures. It also provides a more factual taxonomy of a scientific field from socio-cognitive and historical perspectives compared to the co-citation or bibliometric coupling analyses (Klavans and Boyack; 2017). In this network, each node represents an article. A directed edge from a source node A to a target node B implies that article B is cited by article A. We present a direct citation network for the top 30 articles, according to their degree, in Fig. 2.11. In this figure, we can observe that due to the direct citations, two separate research-based knowledge clusters have been growing in two different scientific communities: social and computer sciences. The articles in the upper knowledge cluster cite Doreian and Mrvar (2009) and can be associated with the

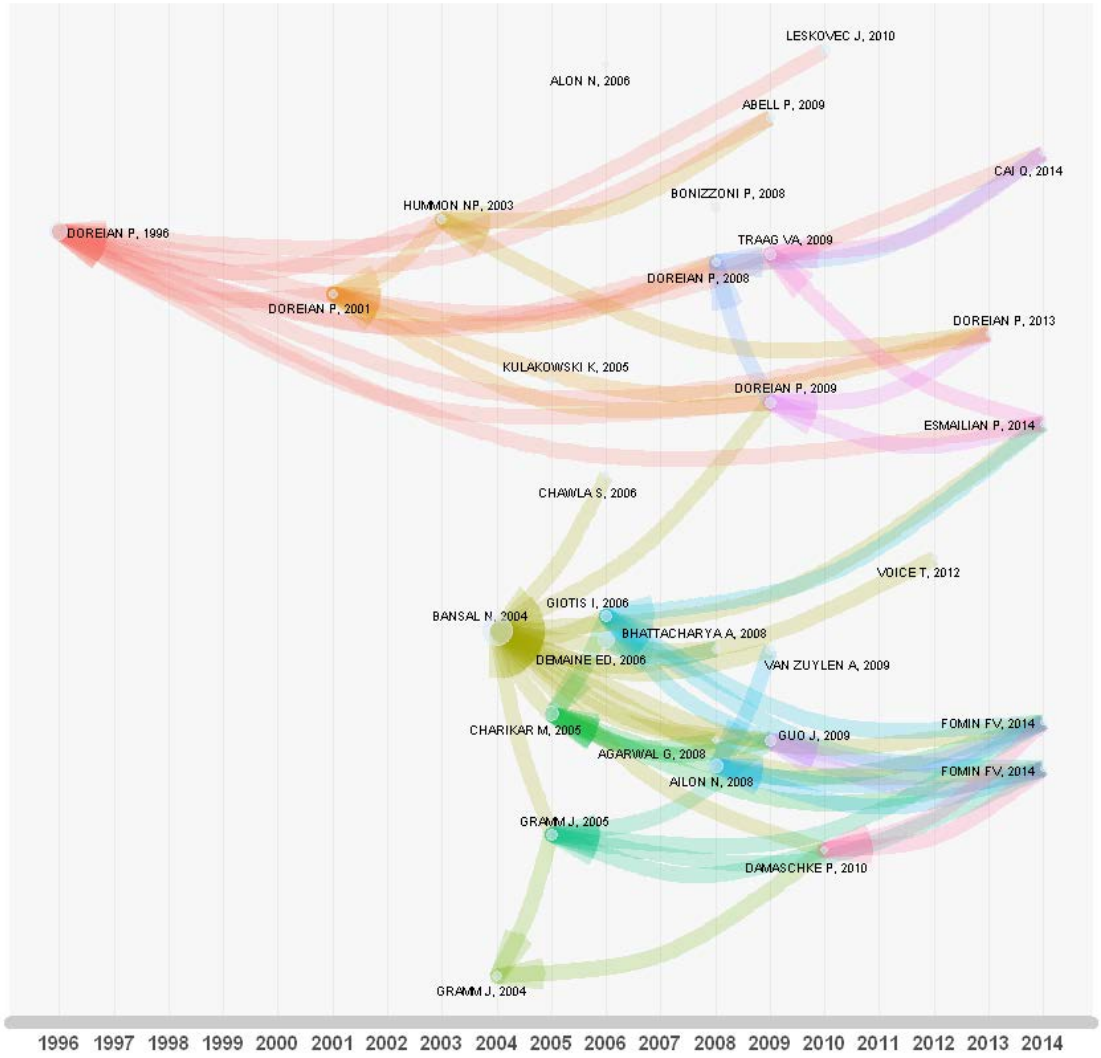


FIGURE 2.11: Direct citation network for the correlation clustering literature.

social sciences community.

On the other hand, the lower knowledge cluster articles cite Bansal et al. (Bansal et al.; 2004) and can be associated with the computer science community. Interestingly, in the direct citation network, we can also notice that the articles in the social science cluster did not receive any citation from the computer science cluster. From this observation, we can say that the research developments on correlation clustering have been conducted inside these two scientific communities somewhat independently. This lack of citations between two scientific areas may create repetition of works. The possible reason behind this citation gap may be using different names for this problem. In social science, the CC problem is known as the structural balance partitioning problem. In Fig. 2.11, we can see that few articles, such as Doreian and Mrvar (1996) and Esmailian et al. (2014) from the social science cluster, are trying to mitigate this citation gap. However, we do not see such efforts from the articles that belong to the computer science cluster. We hope that this review will reverse this trend and help connect these two fields and create a fruitful interdisciplinary stream of research in this area.

2.5.6 Dominant Research Focus Areas

The title and abstract demonstrate the key research concepts of an article that can be associated with a particular scientific field. In addition to providing article properties, such as title, authors, abstracts, authors' keywords, references, and citations, WoS provides "*Keywords Plus*", automatically generated text keywords from the article title, abstracts, and cited references titles (Garfield; 1990; Garfield and Sher; 1993). Keywords Plus generally covers most of the authors' keywords and effectively analyzes scientific fields' knowledge structures (Zhang et al.; 2016). Several studies used Keywords Plus to analyze dominant knowledge structure in different scientific areas (González-Álvarez and Cervera-Crespo; 2017; Khasseh et al.; 2017; Rigolon et al.; 2018; Zhao et al.; 2018; Wahid et al.; 2022).

To understand the ongoing dominant research that focuses on CC, we formulated a keyword co-occurrence network with 665 nodes and 3085 edges. Each node represents a keyword from the articles' title and abstract in this network, and each weighted edge represents the co-occurrence incident(s) of two keywords in an article. We selected a subnetwork of the top 50 nodes (according to the weighted degree) from the keywords co-occurrence network. Then, we applied the modularity algorithm (implemented in Gephi) to identify communities in this sub-network. In Fig. 2.12, we show the three identified clusters of closely related keywords from the keyword co-occurrence subnetwork. Next,

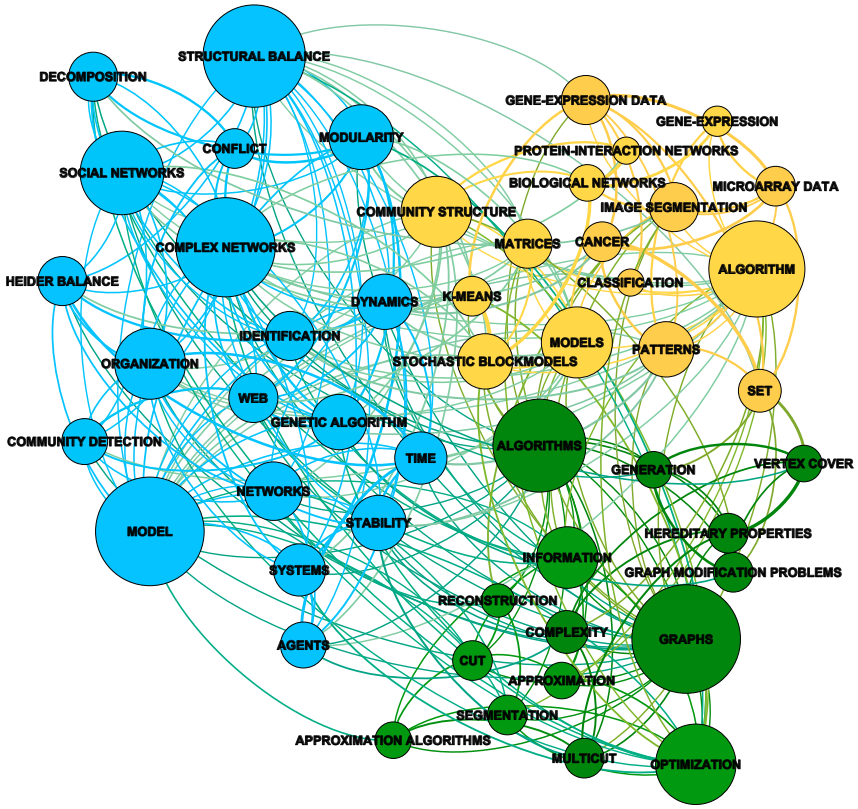


FIGURE 2.12: Keyword Plus's co-occurrence network of size 50 nodes. Nodes are scaled with the corresponding weighted degree.

according to the weighted degree, we present the top 5 keywords from each of the three identified clusters and their corresponding research focus in Table 2.10. This table shows that the keywords associated with Cluster 1 can be linked to the research areas of optimization algorithms and complexity analysis. Similarly, by examining the keywords associated with Cluster 2, we can connect its research focus to analyzing structural balance and community detection in complex social networks. Finally, the research focus of Cluster 3 can be labelled as identifying communities and patterns in complex biological networks.

2.5.7 Future Research Directions and Open Problems

The results from the community detection on keyword's co-occurrence network, presented in Table 2.10, shows that the current research can be categorized in three areas: *Optimization algorithm and complexity analysis*, *Structural balance and community detection in complex social networks*, and *Community and pattern detection in biological*

Cluster & Top 5 Keywords	**Research Focus Areas**
Cluster # 1: Graph, Algorithm(s), Vertex Cover, Approximation (Algorithms), Complexity,	Optimization algorithms and complexity analysis.
Cluster # 2: Model, Structural Balance, Complex Networks, Social Network, Genetic Algorithm	Structural balance and community detection in complex social networks.
Cluster # 3: Community Structure, Patterns, Gene-Expression (Data), Biological Network	Community and pattern detection in biological complex networks

TABLE 2.10: Three identified clusters from the keywords co-occurrence network and their corresponding research area.

complex networks.

The studies related to the first research area explore designing efficient algorithms and provide theoretical guarantees to the solution's accuracy. Bansal et al. (2004) provided a list of open problems in this area of research related to the MINDISAGREE and MAXAGREE, and most of these open problems have been solved in subsequent studies. Demaine et al. (2006) proved that the optimality gap for MINDISAGREE and MAXAGREE approximation is $\mathcal{O}\log n)$. However, there is no such guarantee for the MaxCorr problem. So far, a PTAS with $\omega(1/\log n))$-factor approximation (Charikar and Wirth; 2004), and two of 2-factor approximations (Coleman et al.; 2008) appeared in the research literature, but there is still a lot of room for improvement.

Other two research areas presented in Table 2.10 concentrate on applying the CC problem in different scientific fields. Since this problem is NP-hard, it faces scalability issues while identifying communities in large-scale real-world networks (Pan et al.; 2014). In recent years, several parallel algorithms have appeared in the research literature (Chierichetti et al.; 2014; Pan et al.; 2014, 2015). All of these algorithms can solve very large-scale networks but do not provide any guarantees to the solution's accuracy. Recently, Cambus et al. (2021) proposed a parallel algorithm with worst-case $(1+\epsilon)$-approximation guarantees for the complete signed network. However, the problems of finding efficient parallel algorithms for the MINDISAGREE, MAXAGREE, and MAXCORR problems on the general weighted signed network with a guarantee of the solutions' accuracy are still open.

Another promising area for research is the development of efficient solution approaches for large-scale CC instances that make use of the inherent structural properties

of the problem. One possible direction is to develop specialized decomposition algorithms such as the recent effort by Keuper et al. (2019). They used a novel approach to solve the CC problem using parallel Bender's decomposition in which each node is associated with a Benders' subproblem.

Finally, we present a little-explored but significant scope for future research, comparing the CC with well-known machine learning algorithms. In machine learning, agnostic learning aims to find the best hypothesis for the target function from a given hypothesis class containing the node clusters (Ben-David et al.; 2001; Kearns et al.; 1994). CC can be viewed as a type of agnostic learning, where the link labels are the examples (either positive or negative), and it is only allowed to use partitioning as the hypothesis for the target function (Bansal et al.; 2004). Therefore, the question of comparison between the CC with other well-known ML algorithms should arise naturally. To the best of our knowledge, only Pozzi et al. (2005) investigated this issue. They focus on the comparative analysis of correlation clustering and a machine learning method called support vector machine clustering (SVC). The SVC, proposed in Ben-Hur et al. (2001) is very similar to the well-known support vector machine (SVM) process (Suthaharan; 2016). They showed that SVC performs well compared to the CC in small-scale data with prior knowledge of the number of clusters. In the future, researchers should focus on investigating the performance comparison of CC with other ML algorithms.

2.6 Conclusion

This paper presents a comprehensive survey on the CC problem to identify the taxonomic and bibliometric developments. In the study of taxonomic development, we briefly focused on discussing different solution approaches and their classifications in various scientific communities. Additionally, we summarized various correlation clustering problems with specific constraints. Several novel clustering problems have been proposed in the literature from the inspiration of the CC problem over the years. In this paper, we also briefly discussed two of these clustering problems. Furthermore, as part of taxonomic evolution, we reviewed the relationship of CC with two well-known optimization problems. Finally, we emphasized presenting the problem in different optimization models and their inter-relationships.

In the next part of this paper, we presented the bibliometric evolution of the CC problem. We analyzed and identified several critical bibliometric metrics, such as the

top list of corresponding journals and influential authors. Also, from the clustering analyses in collaboration and co-citation networks, we identified authors and article clusters corresponding to different scientific areas. Furthermore, we pointed out a significant citation gap in the direct citation network that can be attributed to this problem being proposed independently across several disciplines with distinct names. Finally, we identified dominant research topics corresponding to the correlation clustering problem and provided future research directions.

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Acknowledgement We acknowledge support from Natural Sciences and Engineering Research Council (NSERC) Discovery (Award Number: RGPIN-2020-06792) and Mitacs Accelerate Fellowship (Award Number: IT16025) programs.

Data Availability Data derived from public domain resources.

Chapter References

Abdelnasser, A., Hossain, E. and Kim, D. I. (2014). Clustering and resource allocation for dense femtocells in a two-tier cellular ofdma network, *IEEE Transactions on wireless communications* **13**(3): 1628–1641.

Abramo, G., D'Angelo, C. A. and Di Costa, F. (2019). The collaboration behavior of top scientists, *Scientometrics* **118**(1): 215–232.

Achtert, E., Böhm, C., Kriegel, H.-P., Kröger, P. and Zimek, A. (2007). Robust, complete, and efficient correlation clustering, *Proceedings of the 2007 SIAM International Conference on Data Mining*, SIAM, pp. 413–418.

Ahn, K., Cormode, G., Guha, S., McGregor, A. and Wirth, A. (2015). Correlation clustering in data streams, *International Conference on Machine Learning*, PMLR, pp. 2237–2246.

Ailon, N., Avigdor-Elgrabli, N., Liberty, E. and Van Zuylen, A. (2012). Improved approximation algorithms for bipartite correlation clustering, *SIAM Journal on Computing* **41**(5): 1110–1121.

Ailon, N., Bhattacharya, A. and Jaiswal, R. (2018). Approximate correlation clustering using same-cluster queries, *Latin American Symposium on Theoretical Informatics*, Springer, Cham, pp. 14–27.

Ailon, N., Charikar, M. and Newman, A. (2008). Aggregating inconsistent information: ranking and clustering, *Journal of the ACM (JACM)* **55**(5): 1–27.

Akorli, J., Namaali, P. A., Ametsi, G. W., Egyirifa, R. K. and Pels, N. A. P. (2019). Generational conservation of composition and diversity of field-acquired midgut microbiota in anopheles gambiae (sensu lato) during colonization in the laboratory, *Parasites & vectors* **12**(1): 1–9.

Albin, T., Drews, P., Heßeler, F., Ivanescu, A. M., Seidl, T. and Abel, D. (2011). A hybrid control approach for low temperature combustion engine control, *2011 50th IEEE Conference on Decision and Control and European Control Conference*, IEEE, pp. 6846–6851.

Alush, A. and Goldberger, J. (2012). Ensemble segmentation using efficient integer linear programming, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(10): 1966–1977.

Ambrosi, K. (1984). Aggregation binärer relationen in der qualitativen datenanalyse, *Metrika* **31**(1): 274.

Andrade, C. E., Resende, M. G., Karloff, H. J. and Miyazawa, F. K. (2014). Evolutionary algorithms for overlapping correlation clustering, *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pp. 405–412.

Aria, M. and Cuccurullo, C. (2017). bibliometrix: An r-tool for comprehensive science mapping analysis, *Journal of Informetrics* **11**(4): 959–975.

Aszalcós, L. and Bakó, M. (2017). Correlation clustering: a parallel approach?, *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, IEEE, pp. 403–406.

Aszalós, L. and Mihálydeák, T. (2015). Correlation clustering by contraction, *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, IEEE, pp. 425–434.

Avidor, A. and Langberg, M. (2007). The multi-multiway cut problem, *Theoretical Computer Science* **377**(1-3): 35–42.

Bair, E. (2013). Semi-supervised clustering methods, *Wiley Interdisciplinary Reviews: Computational Statistics* **5**(5): 349–361.

Bakó, M. (2018). The efficiency of classification in imperfect databases: comparing knn and correlation clustering, *Annales Mathematicae et Informaticae*, Vol. 49, Eszterházy Károly Egyetem Líceum Kiadó, pp. 11–20.

Bansal, N., Blum, A. and Chawla, S. (2004). Correlation clustering, *Machine Learning* **56**(1): 89–113.

Barik, S., Das, S. and Vikalo, H. (2018). Qsdpr: Viral quasispecies reconstruction via correlation clustering, *Genomics* **110**(6): 375–381.

Barthelemy, J. P. and Monjardet, B. (1981). The median procedure in cluster analysis and social choice theory, *Mathematical social sciences* **1**(3): 235–267.

Bastian, M., Heymann, S. and Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks, *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 3, pp. 361–362.

Basu, S., Banerjee, A. and Mooney, R. (2002). Semi-supervised clustering by seeding, *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*, Citeseer.

Batagelj, V. (1997). Notes on blockmodeling, *Social Networks* **19**(2): 143–155.

Belyaeva, A., Venkatachalapathy, S., Nagarajan, M., Shivashankar, G. and Uhler, C. (2017). Network analysis identifies chromosome intermingling regions as regulatory hotspots for transcription, *Proceedings of the National Academy of Sciences* **114**(52): 13714–13719.

Ben-David, S., Long, P. M. and Mansour, Y. (2001). Agnostic boosting, *International Conference on Computational Learning Theory*, Springer, pp. 507–516.

Ben-Dor, A., Shamir, R. and Yakhini, Z. (1999). Clustering gene expression patterns, *Journal of Computational Biology* **6**(3-4): 281–297.

Ben-Hur, A., Horn, D., Siegelmann, H. T. and Vapnik, V. (2001). Support vector clustering, *Journal of Machine Learning Research* **2**(Dec): 125–137.

Bessonov, K., Walkey, C. J., Shelp, B. J., van Vuuren, H. J., Chiu, D. and van der Merwe, G. (2013). Functional analyses of nsf1 in wine yeast using interconnected correlation clustering and molecular analyses, *PloS One* **8**(10): e77192.

Bhattacharya, A. and De, R. K. (2008). Divisive correlation clustering algorithm (dcca) for grouping of genes: detecting varying patterns in expression profiles, *Bioinformatics* **24**(11): 1359–1366.

Bhattacharya, A. and De, R. K. (2010). Average correlation clustering algorithm (acca) for grouping of co-regulated genes with similar pattern of variation in their expression values, *Journal of Biomedical Informatics* **43**(4): 560–568.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E. (2008). Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10): P10008.

Böcker, S. and Baumbach, J. (2013). Cluster editing, *Conference on Computability in Europe*, pp. 33–44.

Bonchi, F., García-Soriano, D. and Gullo, F. (2022). Correlation clustering, *Synthesis Lectures on Data Mining and Knowledge Discovery* **12**(1): 1–149.

Bonchi, F., Gionis, A., Gullo, F., Tsourakakis, C. E. and Ukkonen, A. (2015). Chromatic correlation clustering, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **9**(4): 1–24.

Bonchi, F., Gionis, A. and Ukkonen, A. (2013). Overlapping correlation clustering, *Knowledge and Information Systems* **35**(1): 1–32.

Bonizzoni, P., Della Vedova, G., Dondi, R. and Jiang, T. (2008). On the approximation of correlation clustering and consensus clustering, *Journal of Computer and System Sciences* **74**(5): 671–696.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine, *Computer networks and ISDN systems* **30**(1-7): 107–117.

Cambus, M., Choo, D., Miikonen, H. and Uitto, J. (2021). Massively parallel correlation clustering in bounded arboricity graphs, *arXiv preprint* .

Cartwright, D. and Harary, F. (1956). Structural balance: a generalization of heider's theory., *Psychological Review* **63**(5): 277.

Cesa-Bianchi, N., Gentile, C., Vitale, F. and Zappella, G. (2012). A correlation clustering approach to link classification in signed networks, *Conference on Learning Theory*, JMLR Workshop and Conference Proceedings, pp. 34–1.

Chagas, G. O., Lorena, L. A. N. and dos Santos, R. D. C. (2019). A hybrid heuristic for the overlapping cluster editing problem, *Applied Soft Computing* **81**: 105482.

Charikar, M., Guruswami, V. and Wirth, A. (2005). Clustering with qualitative information, *Journal of Computer and System Sciences* **71**(3): 360–383.

Charikar, M. and Wirth, A. (2004). Maximizing quadratic programs: Extending grothendieck's inequality, *45th Annual IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 54–60.

Chawla, S., Makarychev, K., Schramm, T. and Yaroslavtsev, G. (2015). Near optimal lp rounding algorithm for correlationclustering on complete and complete k-partite graphs, *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 219–228.

Chen, Z.-Z., Jiang, T. and Lin, G.-H. (2001). Computing phylogenetic roots with bounded degrees and errors, *Workshop on Algorithms and Data Structures*, Springer, pp. 377–388.

Chierichetti, F., Dalvi, N. and Kumar, R. (2014). Correlation clustering in mapreduce, *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 641–650.

Chunaev, P. (2020). Community detection in node-attributed social networks: a survey, *Computer Science Review* **37**: 100286.

Cohen, W. W. and Richman, J. (2002). Learning to match and cluster large high-dimensional data sets for data integration, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 475–480.

Cohn, D., Caruana, R. and McCallum, A. (2003). Semi-supervised clustering with user feedback, *Constrained Clustering: Advances in Algorithms, Theory, and Applications* **4**(1): 17–32.

Coleman, T., Saunderson, J. and Wirth, A. (2008). A local-search 2-approximation for 2-correlation-clustering, *European Symposium on Algorithms*, Springer, pp. 308–319.

Daim, T. U., Rueda, G., Martin, H. and Gerdsri, P. (2006). Forecasting emerging technologies: Use of bibliometrics and patent analysis, *Technological forecasting and social change* **73**(8): 981–1012.

Davis, J. A. (1967). Clustering and structural balance in graphs, *Human relations* **20**(2): 181–187.

De Nooy, W., Mrvar, A. and Batagelj, V. (2018). *Exploratory social network analysis with Pajek: Revised and expanded edition for updated software*, Vol. 46, Cambridge university press.

Demaine, E. D., Emanuel, D., Fiat, A. and Immorlica, N. (2006). Correlation clustering in general weighted graphs, *Theoretical Computer Science* **361**(2-3): 172–187.

Demaine, E. D. and Immorlica, N. (2003). Correlation clustering with partial information, *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, Springer, Berlin, Heidelberg, pp. 1–13.

Donath, W. E. and Hoffman, A. J. (2003). Lower bounds for the partitioning of graphs, *Selected Papers Of Alan J Hoffman: With Commentary*, World Scientific, pp. 437–442.

dong Xie, X., Zou, J. and Huang, X. (2015). Optimization for massive data query method in database, *2015 International Conference on Automation, Mechanical Control and Computational Engineering*, Atlantis Press.

Doreian, P. (2008). A multiple indicator approach to blockmodeling signed networks, *Social Networks* **30**(3): 247–258.

Doreian, P. and Krackhardt, D. (2001). Pre-transitive balance mechanisms for signed networks, *Journal of Mathematical Sociology* **25**(1): 43–67.

Doreian, P. and Mrvar, A. (1996). A partitioning approach to structural balance, *Social Networks* **18**(2): 149–168.

Doreian, P. and Mrvar, A. (2009). Partitioning signed social networks, *Social Networks* **31**(1): 1–11.

Doyle, J. R. (1992). MCC—multiple correlation clustering, *International journal of man-machine studies* **37**(6): 751–765.

Drummond, L., Figueiredo, R., Frota, Y. and Levorato, M. (2013). Efficient solution of the correlation clustering problem: An application to structural balance, *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 674–683.

Esmailian, P., Abtahi, S. E. and Jalili, M. (2014). Mesoscopic analysis of online social networks: The role of negative ties, *Physical Review E* **90**(4): 042817.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise., *kdd*, Vol. 96, pp. 226–231.

Figueiredo, R. and Frota, Y. (2014). The maximum balanced subgraph of a signed graph: Applications and solution approaches, *European Journal of Operational Research* **236**(2): 473–487.

Figueiredo, R. and Moura, G. (2013). Mixed integer programming formulations for clustering problems related to structural balance, *Social Networks* **35**(4): 639–651.

Firman, M., Thomas, D., Julier, S. and Sugimoto, A. (2013). Learning to discover objects in rgb-d images using correlation clustering, *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 1107–1112.

Flake, G. W., Lawrence, S. and Giles, C. L. (2000). Efficient identification of web communities, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 150–160.

Flake, G. W., Tarjan, R. E. and Tsioutsiouliklis, K. (2004). Graph clustering and minimum cut trees, *Internet Mathematics* **1**(4): 385–408.

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

Fortunato, S., Latora, V. and Marchiori, M. (2004). Method to find community structures based on information centrality, *Physical Review E* **70**(5): 056104.

Fukunaga, T. (2019). Lp-based pivoting algorithm for higher-order correlation clustering, *Journal of Combinatorial Optimization* **37**(4): 1312–1326.

Galagedera, D. U. (2013). A new perspective of equity market performance, *Journal of International Financial Markets, Institutions and Money* **26**: 333–357.

Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability*, Vol. 174, freeman San Francisco.

Garfield, E. (1990). KeyWords Plus-ISI's breakthrough retrieval method, *Current Contents* **32**: 5–9.

Garfield, E. (2004). Historiographic mapping of knowledge domains literature, *Journal of Information Science* **30**(2): 119–145.

Garfield, E. and Sher, I. H. (1993). KeyWords Plus [TM]-algorithmic derivative indexing, *Journal-American Society For Information Science* **44**: 298–298.

Geerts, F. and Ndindi, R. (2016). Bounded correlation clustering, *International Journal of Data Science and Analytics* **1**(1): 17–35.

Gionis, A., Mannila, H. and Tsaparas, P. (2007). Clustering aggregation, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **1**(1): 4–es.

Giotis, I. and Guruswami, V. (2006). Correlation clustering with a fixed number of clusters, *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp. 1167–1176.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks, *Proceedings of the National Academy of Sciences* **99**(12): 7821–7826.

Gleich, D. F., Veldt, N. and Wirth, A. (2018). Correlation clustering generalized, *arXiv preprint* .

González-Álvarez, J. and Cervera-Crespo, T. (2017). Research production in high-impact journals of contemporary neuroscience: A gender analysis, *Journal of Informetrics* **11**(1): 232–243.

Grira, N., Crucianu, M. and Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey, *A Review of Machine Learning Techniques for Processing Multimedia Content* **1**: 9–16.

Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K. and Samatova, N. (2014). Community detection in large-scale networks: a survey and empirical evaluation, *Wiley Interdisciplinary Reviews: Computational Statistics* **6**(6): 426–439.

Heider, F. (1946). Attitudes and cognitive organization, *The Journal of psychology* **21**(1): 107–112.

Hinneburg, A., Keim, D. A. et al. (1998). An efficient approach to clustering in large multimedia databases with noise, *KDD*, Vol. 98, pp. 58–65.

Hmimida, M. and Kanawati, R. (2015). Community detection in multiplex networks: A seed-centric approach, *Networks & Heterogeneous Media* **10**(1): 71.

Hsu, J.-w. and Huang, D.-w. (2011). Correlation between impact and collaboration, *Scientometrics* **86**(2): 317–324.

Hu, X., Leydesdorff, L. and Rousseau, R. (2020). Exponential growth in the number of items in the wos, *ISSI Newsletter* **16**(2): 32–38.

Hua, J., Yu, J. and Yang, M.-S. (2021). Star-based learning correlation clustering, *Pattern Recognition* **116**: 107966.

Huang, Y. and Wang, H. (2016). Consensus and multiplex approach for community detection in attributed networks, *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, pp. 425–429.

Il'ev, V., Il'eva, S. and Kononov, A. (2016). Short survey on graph correlation clustering with minimization criteria, *International Conference on Discrete Optimization and Operations Research*, Springer, pp. 25–36.

Isogai, T. (2014). Clustering of japanese stock returns by recursive modularity optimization for efficient portfolio diversification, *Journal of Complex Networks* **2**(4): 557–584.

Jain, A. K., Murty, M. N. and Flynn, P. J. (1999). Data clustering: A review, *ACM computing surveys (CSUR)* **31**(3): 264–323.

Javed, M. A., Younis, M. S., Latif, S., Qadir, J. and Baig, A. (2018). Community detection in networks: A multidisciplinary review, *Journal of Network and Computer Applications* **108**: 87–111.

Ji, S., Xu, D., Du, D. and Gai, L. (2020). Approximation algorithm for the balanced 2-correlation clustering problem on well-proportional graphs, *International Conference on Algorithmic Applications in Management*, Springer, pp. 97–107.

Joglekar, S. R. (2014). Two-stage stock portfolio construction: Correlation clustering and genetic optimization, *The Twenty-Seventh International Flairs Conference*.

Kappes, J. H., Speth, M., Reinelt, G. and Schnörr, C. (2016). Higher-order segmentation via multicuts, *Computer Vision and Image Understanding* **143**: 104–119.

Kearns, M. J., Schapire, R. E. and Sellie, L. M. (1994). Toward efficient agnostic learning, *Machine Learning* **17**(2): 115–141.

Keuper, M., Lukasik, J., Singh, M. and Yarkony, J. (2019). Massively parallel benders decomposition for correlation clustering, *arXiv preprint* .

Khasseh, A. A., Soheili, F., Moghaddam, H. S. and Chelak, A. M. (2017). Intellectual structure of knowledge in imetrics: A co-word analysis, *Information Processing & Management* **53**(3): 705–720.

Kim, S., Nowozin, S., Kohli, P. and Yoo, C. D. (2012). Task-specific image partitioning, *IEEE Transactions on Image Processing* **22**(2): 488–500.

Kim, W. (2009). Parallel clustering algorithms: Survey, *Parallel Algorithms, Spring* **34**: 43.

Klavans, R. and Boyack, K. W. (2017). Which type of citation analysis generates the most accurate taxonomy of scientific and technical knowledge?, *Journal of the Association for Information Science and Technology* **68**(4): 984–998.

Klein, P. N., Mathieu, C. and Zhou, H. (2015). Correlation clustering and two-edge-connected augmentation for planar graphs, *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Krasowski, N., Beier, T., Knott, G., Köthe, U., Hamprecht, F. A. and Kreshuk, A. (2017). Neuron segmentation with high-level biological priors, *IEEE transactions on medical imaging* **37**(4): 829–839.

Lancichinetti, A. and Fortunato, S. (2012). Consensus clustering in complex networks, *Scientific Reports* **2**(1): 1–7.

Levorato, M., Drummond, L., Frota, Y. and Figueiredo, R. (2015). An ILS algorithm to evaluate structural balance in signed social networks, *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pp. 1117–1122.

Levorato, M., Figueiredo, R., Frota, Y. and Drummond, L. (2017). Evaluating balancing on social networks through the efficient solution of correlation clustering problems, *EURO Journal on Computational Optimization* **5**(4): 467–498.

Li, P., Dau, H., Puleo, G. and Milenkovic, O. (2017). Motif clustering and overlapping clustering for social network analysis, *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, IEEE, pp. 1–9.

Lingas, A., Persson, M. and Sledneu, D. (2014). Iterative merging heuristics for correlation clustering, *International Journal of Metaheuristics* **3**(2): 105–117.

López-Sastre, R. J., Tuytelaars, T., Acevedo-Rodríguez, F. J. and Maldonado-Bascón, S. (2011). Towards a more discriminative and semantic visual vocabulary, *Computer Vision and Image Understanding* **115**(3): 415–425.

Lorrain, F. and White, H. C. (1971). Structural equivalence of individuals in social networks, *The Journal of Mathematical Sociology* **1**(1): 49–80.

Maatouk, A., Hajri, S. E., Assaad, M. and Sari, H. (2018). On optimal scheduling for joint spatial division and multiplexing approach in fdd massive mimo, *IEEE Transactions on Signal Processing* **67**(4): 1006–1021.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, Oakland, CA, USA, pp. 281–297.

Makarychev, K., Makarychev, Y. and Vijayaraghavan, A. (2015). Correlation clustering with noisy partial information, *Conference on Learning Theory*, PMLR, pp. 1321–1342.

Marcotorchino, J. and Michaud, P. (1981a). Heuristic approach of the similarity aggregation problem, *Methods of Operations Research* **43**: 395–404.

Marcotorchino, J. and Michaud, P. (1981b). Optimization in exploratory data analysis, *Proceedings of 5th International Symposium on Operations Research*, Physica Verlag Köln.

Marra, F., Poggi, G., Sansone, C. and Verdoliva, L. (2016). Correlation clustering for prnu-based blind image source identification, *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, pp. 1–6.

Marra, F., Poggi, G., Sansone, C. and Verdoliva, L. (2017). Blind prnu-based image clustering for source identification, *IEEE Transactions on Information Forensics and Security* **12**(9): 2197–2211.

Mathieu, C. and Schudy, W. (2010). Correlation clustering with noisy input, *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, pp. 712–728.

Mehta, A., Ashapure, A. and Dikshit, O. (2016). Segmentation-based classification of hyperspectral imagery using projected and correlation clustering techniques, *Geocarto International* **31**(10): 1045–1057.

Mimno, D., McCallum, A. and Mann, G. S. (2006). Bibliometric impact measures leveraging topic analysis, *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries (JCDL'06)*, IEEE, pp. 65–74.

Mirkin, B. (1974). The problems of approximation in space of relations and qualitative data analysis, *Information and Remote Control* **35**(1424-1431): 2.

Mogee, M. E. (1991). Using patent data for technology analysis and planning, *Research-Technology Management* **34**(4): 43–49.

Mondragon, R. J., Iacovacci, J. and Bianconi, G. (2018). Multilink communities of multiplex networks, *PloS One* **13**(3): e0193821.

Morris, S., DeYong, C., Wu, Z., Salman, S. and Yemenu, D. (2002). Diva: A visualization system for exploring document databases for technology forecasting, *Computers & Industrial Engineering* **43**(4): 841–862.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms, *The Computer Journal* **26**(4): 354–359.

Néda, Z., Florian, R., Ravasz, M., Libál, A. and Györgyi, G. (2006). Phase transition in an optimal clusterization model, *Physica A: Statistical Mechanics and its Applications* **362**(2): 357–368.

Néda, Z., Sumi, R., Ercsey-Ravasz, M., Varga, M., Molnár, B. and Cseh, G. (2009). Correlation clustering on networks, *Journal of Physics A: Mathematical and Theoretical* **42**(34): 345003.

Nelson, K. (1973). Some evidence for the cognitive primacy of categorization and its functional basis, *Merrill-Palmer Quarterly of Behavior and Development* **19**(1): 21–39.

Newman, M. E. (2004). Coauthorship networks and patterns of scientific collaboration, *Proceedings of the National Academy of Sciences* **101**(suppl 1): 5200–5205.

Nga, N. T. T., Khanh, N. K. and Hong, S. N. (2016). Entropy-based correlation clustering for wireless sensor networks in multi-correlated regional environments, *IEIE Transactions on Smart Processing and Computing* **5**(2): 85–93.

Nguyen, H.-L., Woon, Y.-K. and Ng, W.-K. (2015). A survey on data stream clustering and classification, *Knowledge and Information Systems* **45**(3): 535–569.

Opitz, O. and Schader, M. (1984). Analyse qualitativer daten: Einführung und übersicht, *Operations-Research-Spektrum* **6**(2): 67–83.

Pan, X., Papailiopoulos, D., Oymak, S., Recht, B., Ramchandran, K. and Jordan, M. I. (2015). Parallel correlation clustering on big graphs, *Advances in Neural Information Processing Systems* **28**.

Pan, X., Papailiopoulos, D., Recht, B., Ramchandran, K. and Jordan, M. I. (2014). Scaling up correlation clustering through parallelism and concurrency control, *DISCML workshop at International Conference on Neural Information Processing Systems*.

Pandove, D., Goel, S. and Rani, R. (2018). Correlation clustering methodologies and their fundamental results, *Expert Systems* **35**(1): e12229.

Papenhausen, E., Wang, B., Ha, S., Zelenyuk, A., Imre, D. and Mueller, K. (2013). Gpu-accelerated incremental correlation clustering of large data with visual feedback, *2013 IEEE International Conference on Big Data*, IEEE, pp. 63–70.

Pons, P. and Latapy, M. (2005). Computing communities in large networks using random walks, *International Symposium on Computer and Information Sciences*, pp. 284–293.

Pozzi, S., Zoppis, I. and Mauri, G. (2005). Combinatorial and machine learning approaches in clustering microarray data, *Biological and Artificial Intelligence Environments*, Springer, pp. 63–71.

Puleo, G. J. and Milenkovic, O. (2015). Correlation clustering with constrained cluster sizes and extended weights bounds, *SIAM Journal on Optimization* **25**(3): 1857–1872.

Rebagliati, N., Rota Bulò, S. and Pelillo, M. (2013). Correlation clustering with stochastic labellings, *International Workshop on Similarity-Based Pattern Recognition*, Springer, Berlin, Heidelberg, pp. 120–133.

Régnier, S. (1965). On some mathematical aspects of automatics classification problems, *ICC bulletin* **4**(3): 175.

Rigolon, A., Browning, M. H., Lee, K. and Shin, S. (2018). Access to urban green space in cities of the global south: A systematic literature review, *Urban Science* **2**(3): 67.

Rokach, L. (2009). A survey of clustering algorithms, *Data Mining and Knowledge Discovery Handbook*, Springer, pp. 269–298.

Rosch, E. (1977). Classification of real-world objects: Origins and representations in cognition, *Thinking: Readings in Cognitive Science* pp. 212–222.

Roy, A. and Pokutta, S. (2017). Hierarchical clustering via spreading metrics, *Journal of Machine Learning Research* **18**: 1–35.

Samal, M., Saradhi, V. V. and Nandi, S. (2018). Scalability of correlation clustering, *Pattern Analysis and Applications* **21**(3): 703–719.

Schaeffer, S. E. (2007). Graph clustering, *Computer Science Review* **1**(1): 27–64.

Shamir, R., Sharan, R. and Tsur, D. (2004). Cluster graph modification problems, *Discrete Applied Mathematics* **144**(1-2): 173–182.

Slaoui, S. C., Dafir, Z. and Lamari, Y. (2018). E-transitive: An enhanced version of the transitive heuristic for clustering categorical data, *Procedia Computer Science* **127**: 26–34.

Solera, F. and Calderara, S. (2013). Social groups detection in crowd through shape-augmented structured learning, *International Conference on Image Analysis and Processing*, Springer, pp. 542–551.

Sumi, R. and Neda, Z. (2008). Molecular dynamics approach to correlation clustering, *International Journal of Modern Physics C* **19**(09): 1349–1358.

Suthaharan, S. (2016). Support vector machine, *Machine Learning Models and Algorithms for Big Data Classification*, pp. 207–235.

Swamy, C. (2004). Correlation clustering: maximizing agreements via semidefinite programming., *SODA*, Vol. 4, pp. 526–527.

Vassy, Z., Kosa, I. and Vassanyi, I. (2017). Correlation clustering of stable angina clinical care patterns for 506 thousand patients, *Journal of Healthcare Engineering* **2017**.

Veldt, N., Gleich, D. F., Wirth, A. and Saunderson, J. (2019). Metric-constrained optimization for graph clustering algorithms, *SIAM Journal on Mathematics of Data Science* **1**(2): 333–355.

Veldt, N., Wirth, A. I. and Gleich, D. F. (2017). Correlation clustering with low-rank matrices, *Proceedings of the 26th International Conference on World Wide Web*, pp. 1025–1034.

von Luxburg, U. (2006). A tutorial on spectral clustering (tech. rep. 149), *Max Planck Institute for Biological Cybernetics* .

Vragović, I. and Louis, E. (2006). Network community structure and loop coefficient method, *Physical Review E* **74**(1): 016105.

Wahid, D. F. (2017). *Random models and heuristic algorithms for correlation clustering problems on signed social networks*, PhD thesis, University of British Columbia.

Wahid, D. F., Ezzeldin, M., Hassini, E. and El-Dakhakhni, W. W. (2022). Common-knowledge networks for university strategic research planning, *Decision Analytics Journal* **2**: 100027.

Wahid, D. F. and Hassini, E. (2022). A literature review on correlation clustering: Cross-disciplinary taxonomy with bibliometric analysis, **3**(3): 1–42.

Wang, H., Tan, S. X.-D., Swarup, S. and Liu, X.-X. (2013). A power-driven thermal sensor placement algorithm for dynamic thermal management, *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, pp. 1215–1220.

Wang, N. and Li, J. (2013). Restoring: A greedy heuristic approach based on neighborhood for correlation clustering, *International Conference on Advanced Data Mining and Applications*, Springer, pp. 348–359.

Wei, F., Sakata, K., Asakura, T., Kikuchi, J. et al. (2018). Systemic homeostasis in metabolome, ionome, and microbiome of wild yellowfin goby in estuarine ecosystem, *Scientific Reports* **8**(1): 1–12.

Williamson, D. P. and Shmoys, D. B. (2011). *The design of approximation algorithms*, Cambridge university press.

Wirth, A. (2010). *Correlation Clustering*, Springer US, Boston, MA, pp. 227–231.

Wirth, A. I. (2005). *Approximation algorithms for clustering*, Princeton University.

Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms, *Annals of Data Science* **2**(2): 165–193.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms, *IEEE Transactions on Neural Networks* **16**(3): 645–678.

Xu, X., Yuruk, N., Feng, Z. and Schweiger, T. A. (2007). Scan: a structural clustering algorithm for networks, *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 824–833.

Yarkony, J., Ihler, A. and Fowlkes, C. C. (2012). Fast planar correlation clustering for image segmentation, *European Conference on Computer Vision*, Springer, pp. 568–581.

Zahn, Jr., C. T. (1964). Approximating symmetric relations by equivalence relations, *Journal of the Society for Industrial and Applied Mathematics* **12**(4): 840–847.

Zaw, C. W., Tun, Y. K. and Hong, C. S. (2017). User clustering based on correlation in 5g using semidefinite programming, *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, IEEE, pp. 342–345.

Zhan, H. C. J., Rea, W. and Rea, A. (2015). An application of correlation clustering to portfolio diversification, *arXiv preprint* .

Zhang, C., Yarkony, J. and Hamprecht, F. A. (2014). Cell detection and segmentation using correlation clustering, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, pp. 9–16.

Zhang, J., Yu, Q., Zheng, F., Long, C., Lu, Z. and Duan, Z. (2016). Comparing keywords plus of wos and author keywords: A case study of patient adherence research, *Journal of the Association for Information Science and Technology* **67**(4): 967–972.

Zhang, Z., Cheng, H., Chen, W., Zhang, S. and Fang, Q. (2008). Correlation clustering based on genetic algorithm for documents clustering, *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE, pp. 3193–3198.

Zhao, Q., Xiong, C., Yu, C., Zhang, C. and Zhao, X. (2016). A new energy-aware task scheduling method for data-intensive applications in the cloud, *Journal of Network and Computer Applications* **59**: 14–27.

Zhao, Q., Xiong, C., Zhang, K., Yue, Y. and Yang, J. (2016). A data placement algorithm for data intensive applications in cloud, *International Journal of Grid and Distributed Computing* **9**(2): 145–156.

Zhao, W., Mao, J. and Lu, K. (2018). Ranking themes on co-word networks: Exploring the relationships among different metrics, *Information Processing & Management* **54**(2): 203–218.

Zhu, Z. and Cao, G. (2011). Toward privacy preserving and collusion resistance in a location proof updating system, *IEEE Transactions on Mobile Computing* **12**(1): 51–64.

Zimek, A. (2009). Correlation clustering, *ACM SIGKDD Explorations Newsletter* **11**(1): 53–54.

Chapter 3

Common-Knowledge Networks for University Strategic Research Planning

The content of this chapter is the published manuscript for publication under the following citation and cited as Wahid et al. (2022):

Wahid, D. F., Ezzeldin, M., Hassini, E., & El-Dakhakhni, W. W. (2022). Common-knowledge networks for university strategic research planning. *Decision Analytics Journal*, 2, 100027. URL.

Common-Knowledge Networks for University Strategic Research Planning

Dewan F. Wahid
School of Computational Science and Engineering
McMaster University, Hamilton, ON, Canada
Email: wahidd@mcmaste.ca

Mohamed Ezzeldin
The INTERFACE Institute
McMaster University, Hamilton, ON, Canada
Email: ezzeldms@mcmaster.ca

Elkafi Hassini
DeGroote School of Business
McMaster University
Email: hassini@mcmaster.ca

Wael W. El-Dakhakhni
The INTERFACE Institute
McMaster University, Hamilton, ON, Canada
Email: eldak@mcmaster.ca

Abstract

We defined the concept of a common-knowledge network of authors in a research institution and used it to identify communities of authors using a new heuristic algorithm for clustering editing problems on weighted similarity measure networks. We analyzed dominant research topics based on most frequent keywords, publications and collaboration incident counts for each identified research community. Our methodology can be used to create multidisciplinary research clusters in universities and support senior management in setting investment strategies for fostering large-scale innovative collaborative initiatives across different disciplines.

Keywords: common-knowledge network; academic research collaboration network; clustering editing; network community.

3.1 Introduction

Knowledge is a driving force behind all social and economical development programs (Phelps et al.; 2012). Many studies indicate that knowledge-based communities benefit their members by enhancing their learning process, contributing to their expertise to understand a given phenomenon, and integrating knowledge to overcome difficulties (Phelps et al.; 2012; Ardichvili et al.; 2003; Malmberg and Maskell; 2002). Interdisciplinary research environments require analysis of knowledge creation, cohesion, and structures to help stimulate innovation and collaboration inside organizations (Gaviria-Marin et al.; 2019; Guan and Liu; 2016; Huang et al.; 2020).

Analyzing knowledge-based research communities also helps the research institutions adopt appropriate policies for knowledge-creating investments, including utilizing the latest technology to stimulate research and developing activities (Connelly et al.; 2012; Malmberg and Maskell; 2002). In order to identify and investigate the research communities in an institution or a scientific area, several network formulation approaches have been adopted in the literature to capture the knowledge structure of that institution or scientific field. As examples of networks, we have research communities based on authors' collaborations (Clauset et al.; 2004; Hu et al.; 2019), co-citations (Ding; 2011; Muñoz-Muñoz and Mirón-Valdivieso; 2017), coauthorships and research impact (Li et al.; 2013), and co-word network based on publication keywords (Katsurai; 2017; Zhao et al.; 2018). Motivated by a practical need from the authors' institution, in this paper we proposed a novel approach to study research communities based on the researchers' commonality of knowledge background.

A community of researchers who share similar knowledge background and use common methodologies can be defined as a collection of researchers who have similar scholarly knowledge or expertise (Malmberg and Maskell; 2002) . To identify researchers' communities in a certain academic area, we need to know how to recognize the intellectual abilities and knowledge associated with a researcher. In many studies, the knowledge elements of an author's published articles are used to define their areas of research (Guan et al.; 2015; Rawlings et al.; 2015; Wang et al.; 2014). In general, a patent based on an article is considered as the author's intellectual property of knowledge, which is represented by the corresponding article (Carnabuci and Operti; 2013). In many recent

studies, the keywords of an article were considered as the knowledge elements that can be used to represent that article (Guan and Liu; 2016; Guan and Zhang; 2018; Muñoz-Leiva et al.; 2012). A keywords-based knowledge network represents the mutual interaction between the different areas of intellectual or scientific knowledge. Several applications of knowledge networks to analyze various empirical phenomena in different scientific areas have been reported in Li et al. (2016); Lozano et al. (2019); Olmeda-Gómez et al. (2017); Zhang et al. (2013).

In this study, we considered the keywords of a publication as the knowledge elements that represent the publication author's areas of research. Instead of investigating and identifying knowledge-based communities in the co-keywords network for a scientific field, in this study, we were interested in identifying and analyzing the researchers' communities based on the commonality of knowledge elements (keywords) among the corresponding researchers (authors) from a specific research institution. Inside each knowledge similarity-based researchers' community, we were interested in investigating the synergy and interaction of the researchers in terms of collaboration, publication count, and dominating research topics. Similar studies of focusing on a particular institution instead of focusing on a specific scientific area can be found in Dahlander and McFarland (2013); Mohsen (2021); Rawlings et al. (2015); Tripathi et al. (2018). Our contribution is to introduce the concept of a common-knowledge network using the keywords associated with their published articles. This network is used with collaboration networks to investigate the influence of the keywords on the dominating research topics, the number of publications, and the collaborations in a community of authors. To analyze such networks, we proposed a similarity correlation clustering model, a variant of correlation clustering on general weighted networks(Bansal et al.; 2004; Demaine et al.; 2006), and apply a heuristic algorithm to identify network communities in which the weighted link represents the similarity between two nodes (i.e., researchers)[1]. By using this heuristic algorithm, we identified communities of authors from the formulated common knowledge-based network. We also recognized possible dominating research topics and analyzed the publication numbers and collaborations corresponding to each designated community. Our results showed that the publications number and collaboration incidents corresponding to identified communities tend to increase over the studied period. As a case study, we applied our methodology to publications by the authors' institutions' researchers and their collaborators from 2011 to 2016. This study was conducted in early 2018, and we collected published journal information from the Web of

[1]The published paper had a typo: "proposed a similarity correlation clustering, a variant of correlation clustering" should be "used the clustering editing."

Science (WoS), an online citation indexing service. The choice of the study period was motivated by a practical need at that institution, where one of the authors was involved in a committee that was tasked with developing a 5-year strategic plan for research driven by a vision to encourage multidisciplinary research activities and devising appropriate investment strategies to nurture such initiatives. The ultimate goal was to develop strong multidisciplinary research groups that can successfully compete in major national grant competitions. Given that most research investments are derived from government funding agencies. We chose to study a six-year period to correspond with the period used by government funding agencies to evaluate researchers' excellence.

Our study was motivated by the university's drive towards interdisciplinary research and an ensuing debate on how to define research communities and encourage interdisciplinary research within those groups. This issue became paramount in discussions within a committee of researchers that were tasked with devising a strategic research plan for the university, among whom was one the authors of this paper. The study demonstrates the need to devise an objective tool to define possible interdisciplinary research communities and use that tool to target funding for the different research clusters.

Section 3.2 defined the collaboration network (CN) and the common-knowledge network (CKN) and presented network formulation and descriptive analysis. Section 3.3 discussed the community clustering approach for identifying research communities and presented a heuristic algorithm for the clustering editing problem on weighted networks to identify research communities in CKN. We discussed the limitations of our study in Section 3.4. Finally, Section 3.5 summarizes our findings and proposes possible future research directions.

3.2 Network Definitions and Descriptive Analytics

3.2.1 Collaboration Network

Research CN, also known as the co-authorship network, has been investigated for several years to analyze collaborations in a research community and identify key researchers in a particular research area (Liu et al.; 2015). The CN represents a form of scholarly collaboration among researchers that includes scientific interactions and collective coordination of conducting research and finally producing results in the form of a publication (Abbasi et al.; 2014).

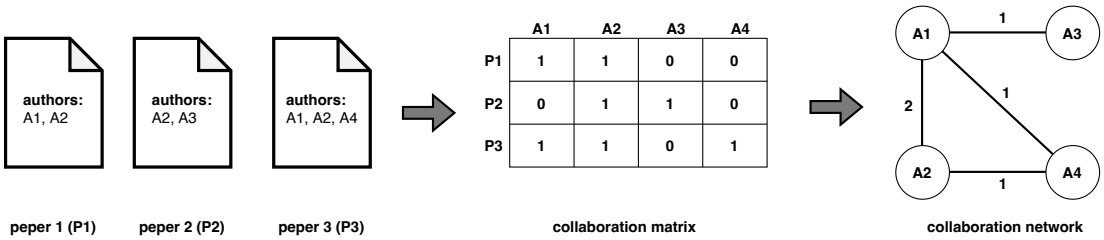


FIGURE 3.1: The collaboration matrix and CN formulation from three journal articles (papers) and, consequently, from the collaboration matrix. Here, $A1$, $A2$, A, and $A4$ are the authors of the three papers/articles $P1$, $P2$, and $P3$[2].

In a CN, each node represents an author of a paper, while each link represents the collaboration incident (coauthorship) between two authors in an article. The number of co-authored articles represents the number of the collaboration incident between two authors. Therefore, the weight of a link in a CN can be defined as the number of collaboration incidents between two authors in the network. In the data processing step, we use a collaboration matrix $CM = (c_{ij}), i = 1, \ldots, n;\ j = 1, \ldots, m$, to formulate CN. Each node represents an author, and each link represents a collaboration incident between two authors. Each non-zero entry in CM indicates the authorship of an article by the corresponding author. Any non-zero row for any two distinct columns in this matrix represents a collaboration incident and consequently represents a link between two corresponding authors in the collaboration network. The total number of non-zero rows between two distinct columns can be defined as the weight of the corresponding link in CN. For any two authors p and q, the number of collaboration incident (link weight in CN) w_{pq} can be defined as follows:

$$w_{pq} = \sum_{i=1}^{n} c_{pi} c_{qi} \tag{3.1}$$

where $c_{pi} = 1$ if author p has an authorship in article i, otherwise $c_{pi} = 0$. In Fig. (3.1), we illustrate CM and CN for three articles and four authors.

3.2.2 Common-Knowledge Network

Based on the knowledge elements (keywords) of a publication, we can define the authors' knowledge as the dimensions and class of the intellectual areas in which these authors

[2]The published paper a had typo: the top right node 'A4' node should be 'A3'.

have expertise. As such, we can also specify the term common-knowledge as the common area of expertise between two authors. Subsequently, the common-knowledge between two authors can be represented by a set of common keywords corresponding to these two authors' publications. In this research, we investigate the influence of the common-knowledge (keywords) on the dominating research topics, the number of publications and the collaboration in a community of authors.

We defined a CKN based on the available common knowledge (keywords) between two authors. In this network, each node represents an author from the research community, while each link and its corresponding weight represent the number of common-knowledge (keywords) in their published articles. Similar to the CM formulation, we matrix $CKM = (k_{ij}), i = 1, \ldots, n;\ j = 1, \ldots, m$, to formulate CKN. Any non-zero row for any two distinct columns in this matrix represents a common-knowledge and consequently represents a link between two corresponding authors in the CKN. The total number of non-zero rows between two distinct columns can be defined as the weight of the corresponding link in CKN. For any two authors p and q, the number of common-knowledge elements (link weight in CKN) w_{pq} can be defined as follows:

$$w_{pq} = \sum_{i=1}^{n} k_{pi} k_{qi} \tag{3.2}$$

where $k_{pi} = 1$ if author p is familiar to keyword i, otherwise $k_{pi} = 0$. A brief representation of the common-knowledge matrix and the common-knowledge network is given in Fig. (3.2).

3.2.3 Data Collection and Processing

Web of Science (WoS) is an online citation indexing service that provides access to multiple cross-reference databases. In addition to providing article properties, such as title, authors, abstract, keywords, and references, they provide *Keywords Plus* and several other document database identifiers and classifications. *Keywords Plus* are automatically generated by text mining from titles of cited references (Garfield; 1990; Garfield and Sher; 1993).

We collected all journal information (e.g., authors, keywords, affiliation, citations, etc.) published by McMaster University-based authors and their collaborators between the years 2011 and 2016 inclusive. Table 3.1 shows the numbers of yearly published articles, along with the corresponding numbers of articles with authors' defined keywords

Year	Number of Articles	Number of Articles with Authors' Keywords	Number of Articles with Keywords Plus
2011	4296	2130	3202
2012	4618	2281	3373
2013	4840	2530	3598
2014	5032	2508	3694
2015	5322	2804	3941
2016	5375	2834	4120
Total	29483	15087	21928

TABLE 3.1: Number of research articles from 2011 to 2016 in the initial data collection.

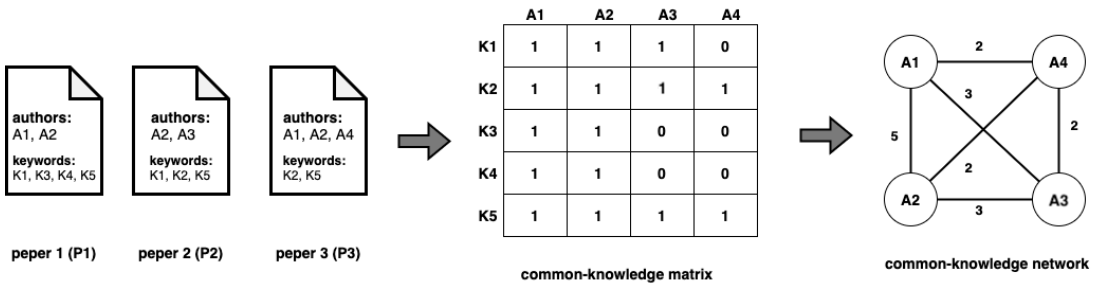


FIGURE 3.2: Common-Knowledge Network (CKN) formulation from three journal articles. Here, $A1$, $A2$, $A3$, and $A4$ are the authors, and $K1, ..., K5$ are the keywords associated with the three articles[3].

and *Keywords Plus*. Table 3.2 shows that about 74% of the collected articles have Keywords Plus as opposed to only about 51% having author keywords. Some of the articles that do not have authors' keywords used field classification such as Physics and Astronomy Classification Scheme (PASC), or Mathematics Subject Classification (MSC). Therefore, we used *Keywords Plus* records to formulate CKNs. *Keywords Plus* covers most of the authors' keywords and found it more effective for analyzing scientific fields' knowledge structures (Tripathi et al.; 2018; Zhang et al.; 2016). Recently, several authors used *Keywords Plus* to analyze knowledge cohesion and structure in different scientific areas (González-Álvarez and Cervera-Crespo; 2017; Khasseh et al.; 2017; Rigolon et al.; 2018; Zhao et al.; 2018). By focusing only on articles with *Keywords Plus*, we ended up with 21,928 research articles published by authors from McMaster University and their collaborators. For simplicity, in the sequel, we will use keywords to refer to *Keywords Plus*. Since our focus is on McMaster University-based researchers, we removed all

[3]The published paper two had typos: the common-knowledge matrix entry (K2, A3) should be 1, and the bottom right node 'A4' node should be 'A3'.

Year	Links		Nodes	#Of articles
	CN	CKN		
2011	3560	27040	2513	3202
2012	3731	34618	2606	3373
2013	4458	39845	2835	3598
2014	4128	47218	2767	3694
2015	5355	64498	3115	3941
2016	5164	68061	3052	4120
2011-16 (merged)		529766	9022	21928

TABLE 3.2: The number of links and nodes in the yearly CNs, and yearly and merged CKNs from 2011 to 2016.

authors' names who do not have an affiliation with McMaster University.

3.2.4 Removing Authors' Name Repetition

We noticed that many authors' names appear in different articles in different formats. In order to overcome this name formatting issue, we developed a procedure for creating an author-name dictionary to avoid the repetition of the same author-name in different formats. The steps of this procedure are described in *Appendix A*.

After removing duplicate authors, we found that 9,022 unique authors were affiliated with McMaster University, representing 16.6% of the 57,306 authors. To create a complete data set for analysis, we merged the cleansed publication data sets for 2011–2016.

3.2.5 Networks Formation

First, we formulated yearly CNs from the yearly cleaned publication data from 2011 to 2016. Since the articles with a single author do not yield any collaboration incident, we ignored all articles with a sole author to formulate the CN.

Second, we formulated CKNs from the yearly and merged cleaned publication data from 2011 to 2016. To prevent link creation due to the keywords that are not directly related to the authors' expertise (e.g., country, province, and states names), in this step, we maintained a dictionary of keywords that we need to avoid. We ignored these keywords during the formulation of CKNs. We called the formulated CKNs from the yearly and merged data as yearly CKN and merged CKN, respectively. From the merged CKN, we removed links with a weight of less than two. This is mainly to avoid weak links between two authors due to too general keywords. The numbers of nodes and links in the formulated CNs and CKNs are given in Table 3.2.

3.2.6 Network Properties and Centrality Measures

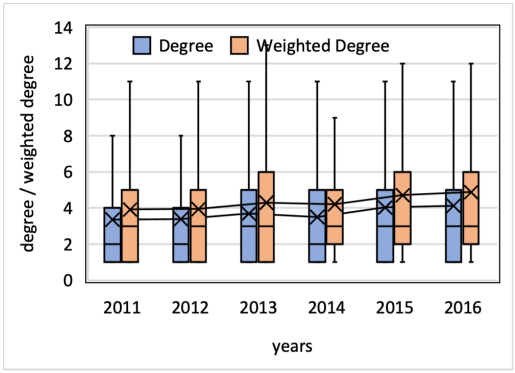
The temporal analysis of a network helps to understand the network structure's evaluation (Diaz and Poblete; 2019; Holme and Saramäki; 2012). To analyze the temporal behaviours in yearly CNs and CKNs, we examined the following well-known network properties: average degree, average weighted degree, average clustering coefficient, and network diameters (McFadyen and Cannella Jr; 2004; Watts and Strogatz; 1998). The centrality measures in network analysis give nodes' position, significance, and pivotal role in the network's local and global scale (Bringmann et al.; 2019; Gupta et al.; 2016; Lee et al.; 2021). We considered betweenness centrality (Girvan and Newman; 2002) and closeness centrality (Brandes; 2005) in the merged CKN.

Temporal Analysis in Yearly CNs and CKNs

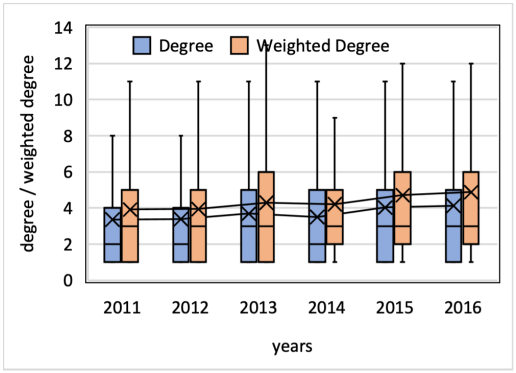
We analyzed the degree and weighted degree distributions, average clustering coefficient, and network diameter values for the CNs and CKNs to evaluate McMaster University's authors' networks' temporal characteristics.

Note that each node in CN and CKN represents an author of a publication. The weighted link between two authors (nodes) in CN represents the number of publications co-authored by such authors, whereas the weighted link between two authors in CKN represents the number of common-knowledge elements (keywords). Therefore, the degree in CN represents the number of co-authors of the corresponding author, whereas the degree in CKN represents the number of researchers with whom the author shares similar expertise or knowledge. Similarly, the weighted degree in CN represents the number of publications in which the corresponding author is a co-author, while the weighted degree in the CKN represents the commonality of the knowledge elements among the author in his/her local networks.

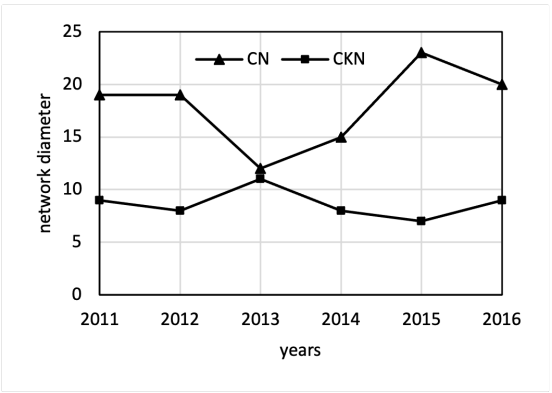
The network diameter represents the maximum shortest distance between two authors in CN through a number of intermediate authors due to the co-authored publications. The CKN network diameter represents the maximum shortest distance between two authors regarding the number of intermediate authors resulting from the common-knowledge elements. Similarly, the average clustering coefficient in CN indicates the average measure of coauthorship among the authors in this network, while in a CKN it represents the average measure of similarity of the authors in this network due to the common-knowledge elements.



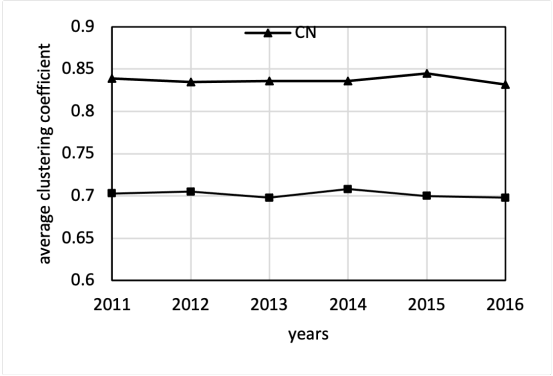
(A) CN



(B) CKN



(C) Network diameters



(D) Average clustering coefficients

FIGURE 3.3: Degrees and weighted degrees in yearly (A) CNs, (B) CKNs; (C) Network diameter in the yearly CNs and CKNs; (D) Average clustering coefficients in the yearly CNs and CKNs, for each year from 2011 to 2016.

Year	**Average number of references per article**	**Average N_k (Keywords Plus) per author**
2011	44.87	13.55
2012	45.41	13.53
2013	46.03	13.81
2014	45.53	14.25
2015	46.98	14.18
2016	49.22	14.59

TABLE 3.3: The average number of references per article and the average number of Keywords Plus per author for each year from 2011 to 2016.

Figs. 3.3a and 3.3b show that the average degree and the weighted degree values in CNs and CKNs increased from 2011 to 2016. In both types of networks, most authors have a degree and weighted degree below the average. In addition, Fig. 3.3a suggests that a significant number of authors (from the top long whisker) in this university were collaborating above average. Similarly, Fig. 3.3b also indicates that some authors had more than average knowledge elements commonality with others. As for the network diameter (Fig. 3.3c) and the average clustering coefficient (Fig. 3.3d), we did not observe any statistically significant trends in both CN and CKN. This observation suggests that the authors' number of overall collaboration incidents and commonality of knowledge elements increased within their local network (peers). In contrast, they remained constant in the whole network from 2011 to 2016.

Given that an article had on average 22.68 *Keywords Plus* and 1.67 authors and that each author had on average 3.55 publications, it is not surprising that the diameter of the CKNs is smaller than that of the CNs. The fact that the commonality of knowledge element within an author's peers increased in CKNs can be explained by the growth in the number of references per article over time, as shown in Table 2. One can observe that the average number of references per article increased from around 44.87 to 49.22 from 2011 to 2016. This phenomenon has been observed in previous studies (Biglu; 2008; Bornmann and Mutz; 2015; Price; 1965). Since *Keywords Plus* were mined from reference titles, growth in the number of titles would lead to a growth in the number of *Keywords Plus* per article and its corresponding author(s) as shown in Table 3.3. Therefore, the tendency to increase the number of *Keyword Plus* corresponding to an author led to an increase of the commonality in knowledge (keywords) with their peers.

Based on such observations, we can see that the changes in collaboration incidents and the commonality of knowledge elements occur only within the peers' network. This

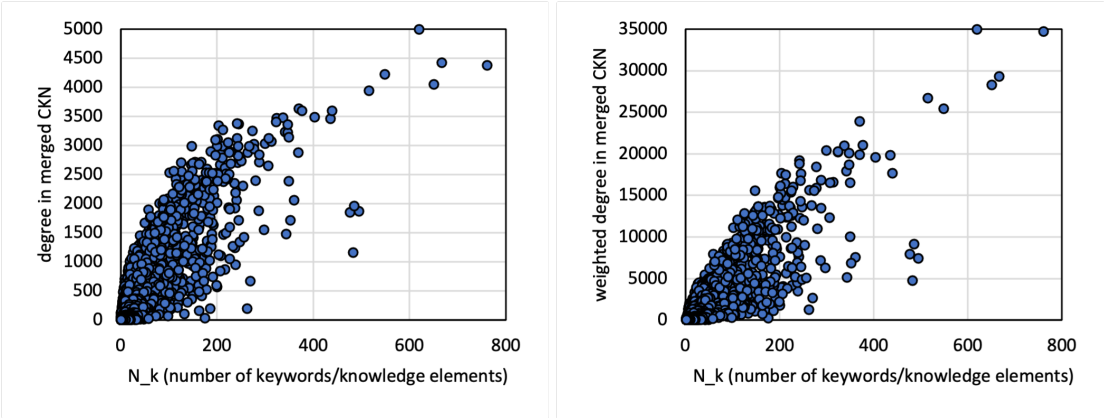
suggests that there were no significant change in the intensity of interdisciplinary research among different groups in the university. Given the university's current investment in interdisciplinary funding initiatives, it would be worthwhile to conduct a similar analysis in the future to find out whether or not these investments have increased the intensity of interdisciplinary research.

Centralities in Merged CKN

In this section, we examined four centrality measures for the merged CKN over the period 2011-2016 to analyze the authors' interactions in a research community due to common-knowledge elements. The centrality measures are used to identify the central nodes in the network. The centrality of an author in the CKN illustrates the possibility of sharing similar knowledge element(s) with other authors of the network. The unweighted degree centrality can be interpreted as the number of authors that share relevant expertise or knowledge with an author. Similarly, the weighted degree centrality of a node can be defined as the number of knowledge elements (keywords) that are shared by the author with other authors in the network. The betweenness centrality, in CKN, of an author can refer to how an author's knowledge elements can be used as a bridge between any two authors in the network. Finally, an author's closeness centrality can be interpreted as the degree of similar expertise (keywords/ knowledge elements) with other authors in the research community network. An author, who has knowledge in diverse areas, can have more common knowledge with other authors in the network. Therefore, based on the CKN definition, an author with a large number of knowledge elements may receive a high central position in the network.

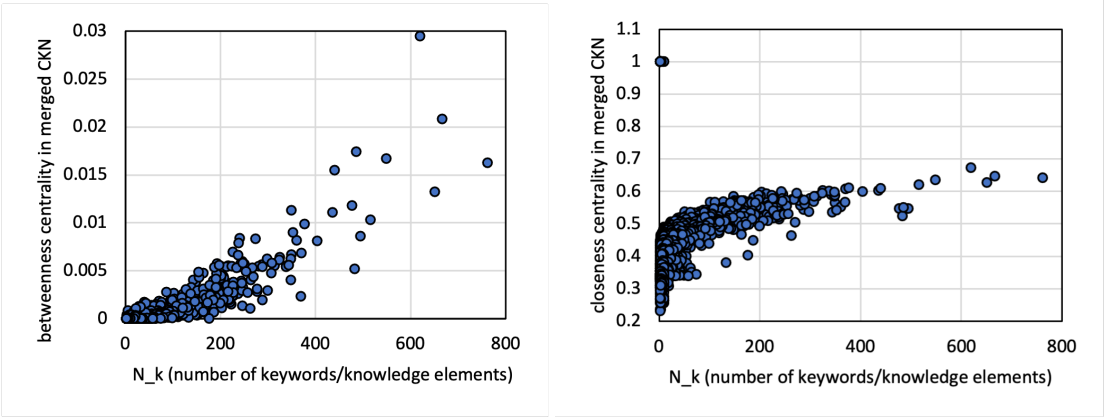
In order to investigate the relationship between N_k, the number of knowledge elements (keywords) that an author is familiar with, and other centrality measures, first, we calculated N_k associated with each author in the merged CKN. Afterwards, we plotted the degree, weighted degree, betweenness, and closeness centralities of each author versus the corresponding N_k value. Figs. 3.4a and 3.4b show the relationships between N_k associated with each author(s) and the corresponding degree and weighted degree centralities in the merged CKN, respectively. In both figures, the degree and weighted degree centralities showed an increasing trend in N_k.

Fig. 3.5a and 3.5b present the relation between N_k associated with each author and the corresponding betweenness and closeness centralities in the merged CKN, respectively. In Fig. 3.5a, we see that the betweenness centrality increased with N_k. Similarly, in Fig. 3.5b, that the closeness centrality increased with N_k albeit at a lower rate. In



(A) (B)

FIGURE 3.4: (A)Degree of the authors, and (B) weighted degree of the authors in the merged CKN versus their corresponding N_k.



(A) (B)

FIGURE 3.5: (A) Betweenness centrality, and (B) closeness centrality of the authors in the merged CKN versus their corresponding N_k.

FIGURE 3.6: Word-cloud corresponding to the outliers authors in Fig. 3.5b.

this figure, we also observed that there are 54 outliers (in the top-left corner of Fig. 3.5b). Despite of having a low number of N_k, these outlier authors have high closeness centrality in the merged CKN. To further investigate the outliers, we plotted their corresponding keywords-cloud in Fig. 3.6. This word cloud provided an idea about the keywords responsible for achieving such high closeness centrality values by the outliers authors. Some of the most frequent keywords were: *jokulhlaups, perception, sub-channel, flow deflectors, magnetic data, 3D inversion, model and nerve.* These keywords could not be linked to a particular area of research. One possible explanation is that these knowledge elements or keywords were either very basic or popular across the research disciplines in the McMaster researchers community. To conclude, we can say that the degree, weighted degree and betweenness centralities were positively related to the corresponding N_k values.

As a matter of managerial insights for Universities' research offices, the centrality measures in CKNs can be used to identify the most shared keywords between an institution's research community. For example, in our case study, we observed that from the closeness centrality graph, the keywords that were most unifying occur when N_k between 25 and 75. In that range, the marginal increase in centrality was maximized.

3.3 Community Clustering in a Network

In a network, a community is a group of nodes that are strongly connected to each other. The definition of strongly connected can be drawn from different desirable properties such as high link density, low diameter, or sharing other similar features (Brandes; 2005; Clauset et al.; 2004). Based on different elements, most community detection algorithms can be classified into two categories: an agglomerative algorithm or a divisive algorithm (Fortunato; 2010). Hierarchical and spectral clustering are two examples of agglomerative algorithms that identify communities by recursively merging similar nodes or communities in the network (Blondel et al.; 2008; Newman; 2006; Pons and Latapy; 2006). On the other hand, cluster edit is a divisive algorithm that identifies communities from the network by link deletion and, (or) insertion followed by an optimization process (Böcker and Baumbach; 2013; Clauset et al.; 2004; Girvan and Newman; 2002; Radicchi et al.; 2004)

In order to identify communities in the CKN, we proposed a new heuristic algorithm for Cluster Editing (CE) for weighted networks, which is a variant of the correlation clustering problem. Our motivation was to use CE that allows the detection of communities in various sizes and densities (Fortunato; 2010). To assess the performance of identifying communities using our proposed heuristic algorithm, we compare it with three well-known algorithms for community clustering: First Unfolding Modularity Clustering (Blondel et al.; 2008), Girvan-Newman Clustering (Girvan and Newman; 2002), and Leiden Clustering (Traag et al.; 2019).

3.3.1 Clustering Editing for Weighted Network

Identifying communities in a network based on the node similarities (equivalently on link weights) is a well-studied problem (Schaeffer; 2007). This problem is referenced with different names in the literature depending on the network types. For example, *Community Clustering* (Clauset et al.; 2004), *Clique Partitioning* Problem for the complete unweighted network (Grötschel and Wakabayashi; 1989), *Graph Clustering* (Dhillon et al.; 2007), and Graph Clustering Editing for general weighted/unweighted network (Böcker et al.; 2011; Böcker and Baumbach; 2013). In an unweighted network, the link represents the mutual similarity between two nodes, whereas, in a weighted network, the link's weight represents the degree of the mutual similarity between two nodes. Community detection on weighted/unweighted networks is NP-hard (Böcker and Baumbach; 2013; Delvaux and Horsten; 2004). Over the years, several community detection algorithms have been proposed in the literature. A significant number of these community

detection algorithms are based on either link cuts or node partitioning ((Alpert et al.; 1999; Chopra and Rao; 1993; Flake et al.; 2004; Nascimento and De Carvalho; 2011; Shi and Malik; 2000). These algorithms are not fit for large-scale networks and can solve networks under 1,000 nodes (Böcker and Baumbach; 2013).

To identify communities from a weighted network, in which the weighted links are representing the mutual nodes similarity, we formulated a clustering editing problem (Böcker et al.; 2011; Křivánek and Morávek; 1986). The clustering editing (CE) problem focuses on the cliques' property in the network structure to identify the communities. A clique is a set of nodes with a direct link between any two nodes.

To define the CE problem, we assume we have a given weighted network $G = (V, E)$ where V is a set of n nodes and E is the links set. We derive a complete shadow network $G' = (V, E')$ from the node set V. The goal is to find the minimum number of link modifications (deletion and insertion) in G' such that the modified network is a partition $\mathcal{P}_1, ..., \mathcal{P}_k$ of k disjoint cliques. Each weighted link $(i, j) \in E$ in G represents a mutual similarity relation between two nodes $i, j \in V$ and k is an arbitrary number. Let $E(\mathcal{P}_i, \mathcal{P}_j)$ be the set of links that will be deleted in the process of making disjoint cliques $\mathcal{P}_i$ and $\mathcal{P}_j$. Similarly, let $\bar{E}(\mathcal{P}_i)$ be the set of links that are added to formulate the clique $\mathcal{P}_i$. The CE for a weighted network can be formulated as the following optimization problem:

$$\min_{1 \geq k \geq n} \min_{\cup_{i=1}^{k} \mathcal{P}_i = V;\, \mathcal{P}_i \cap \mathcal{P}_j = 0;\, \forall i \neq j} \sum_{i=1}^{k} \sum_{j=i+1}^{k} E(\mathcal{P}_i, \mathcal{P}_j) + \sum_{i=1}^{k} \bar{E}(\mathcal{P}Ii) \tag{3.3}$$

3.3.2 Integer Program Formulation for CE on a Weighted Network

For solution benchmarking purposes, we proposed an integer program formulation for the problem given in Eq. (3.3). Demaine et al. (Demaine et al.; 2006) formulated an integer program for the correlation clustering problem on general weighted networks. On the other hand, Böcke et al. Böcker et al. (2011) and Grötschel et al. (Grötschel and Wakabayashi; 1989) proposed integer program formulations for the clustering problem on complete and general weighted networks, respectively. Starting from these formulations, we proposed an integer program for CE on a weighted network.

We take w_{ij} as a positive weight associated with link $(i, j) \in E$ representing a measure of similarity between the nodes i and $j \in V$. For the complete shadow network G', we

assign a binary decision variable x_{ij} for each distinct link $(i, j) \in E'$ as:

$$x_{ij} = \begin{cases} 0; & \text{if } i \text{ and } j \text{ are in the same cluster (a link in the shadow network),} \\ 1; & \text{otherwise.} \end{cases} \tag{3.4}$$

The CE on a weighted network $G = (V, E)$ can be formulated by using the link edition on the complete shadow network $G' = (V, E')$ as follows:

$$\min \quad \sum_{(i,j)\in E} w_{ij}x_{ij} + \sum_{(i,j)\notin E} \delta_{ij}(1 - x_{ij}); \quad \forall i, j \in V \tag{3.5}$$

$$\text{s. t.:} \quad x_{ij} + x_{jk} \geq x_{ik}; \quad \forall i, j, k \in V, \tag{3.6}$$

$$x_{ij} = x_{ji}; \quad \forall i, j \in V, \tag{3.7}$$

$$x_{ij} \in \{0, 1\}; \quad \forall i, j \in V. \tag{3.8}$$

Where the objective function, in Eq. (3.5), represents the clustering error due to the similarity links across the partitions (link deletion in the shadow network) and nodes that are not linked within a partition (link insertion in the shadow network). The constant δ_{ij} represents the link insertion cost for putting two non-connected nodes i and j in one partition (cluster). The triangle inequality constraints, in Eq. (3.6), force any node k to be inside a cluster P when that cluster has its two neighbours i and j. Constraints, in Eq. (3.7), represent the undirected properties of the similarity link $(i, j) \in E$. The last set of constraint, in Eq. (3.8), represent the binary requirements for x_{ij}.

Several approaches have been suggested for calculating δ_{ij} (Böcker et al.; 2011; Böcker and Baumbach; 2013; McAssey and Bijma; 2015; Opsahl and Panzarasa; 2009; Serrano et al.; 2006). As mentioned in (Opsahl and Panzarasa; 2009), the choice of δ_{ij} should be based on the research question. Here we chose to use an arithmetic mean for its simplicity and the absence of extreme weights given that the number of keywords in articles is usually limited:

$$\delta_{ij} = \min_k \{ \frac{w_{ik} + w_{kj}}{2} : (ik), (jk) \in E, k \in N(i) \cap N(j), (ij) \notin E \} \tag{3.9}$$

where $N(i)$ and $N(j)$ are the set of neighbour nodes for i and j, respectively.

3.3.3 Heuristic Algorithm for CE on Weighted Network (`HACEWN`)

Křivánek and Morávek (Křivánek and Morávek; 1986) showed the CE problem is NP-hard. Therefore, this problem can not be solved efficiently with available commercial solvers (e.g., CPLEX, GUROBI) for large scale networks (Böcker et al.; 2011; Böcker and Baumbach; 2013; Queiroga et al.; 2021). To identify communities in large-scale network, we propose using a CE for a weighted network and design a heuristic algorithm (`HACEWN`) to identify communities from the merged CKN of McMaster University-affiliated authors over the period 2011-2016. Our heuristic algorithm builds on previous work by Chierichetti et al. (Chierichetti et al.; 2014) and Pan et al. (Pan et al.; 2015). The main premise of the heuristic is to decompose the network into smaller networks, on which the CE problem can be solved to optimality, and then use their solutions to construct an approximate solution to the original large scale network. The pseudo-code of `HACEWN` is given in *Algorithm 3.1.* It has three main recursive steps that are described in more details in the subsections below.

Induced Network Selection

In the first major step of `HACEWN`, we select a node v from G uniformly at random (u.a.r) and create $N_v = \{u \in V \mid (u, v) \in E\}$, a set of all neighbours of v. We then consider $IN_G(v) = G(N_v, E_(N_v)\ )$ as the induced network for N_v, where $E_(N_v) = (u, v) \in E \mid u, v \in N_v$ is the set of all links among the nodes in N_v. We note that $IN_G(v)$ is a subnetwork of G. If the number of nodes in $IN_G(v)$ is greater than a given threshold, l, we reconstitute $IN_G(v)$ by randomly selecting its l nodes from $N_v \cup v$.

The threshold network parameter l can be determined by the maximum size of the network on which we can efficiently and exactly solve the integer linear program formulation of the CE problem. In our computational experiments, we set $l = 20$.

Solving Exact CE on Induced Network $IN_G(v)$

In the second step of `HACEWN`, we execute $\mathcal{E}xact_{CE}(IN_G(v))$ to find an exact CE on the induced network $IN_G(v)$ by using the integer program formulation Eqs. (3.5)-(3.8). Let the set of clusters returned by the solution to this integer program be $\{C_1, ..., C_k\}$, where each $C_k \subseteq N_V$ and $C_i \cap C_j =; \forall i \neq j$.

Algorithm 3.1 Heuristic Algorithm for CE on Weighted Network (`HACEWN`)

procedure HEURISTIC_CE($G(V, E)$)
  $l \leftarrow constant$ ▷ maximum size of induced network to be solved
  while $V \neq 0$ **do**
    $v \leftarrow rand(V)$ ▷ select a node v u.a.r from V
    $N_v \leftarrow \{u \in V \mid (u, v) \in E\}$
    $E_{N_v} \leftarrow \{(u, w) \in E \mid u, w \in N_v \cup v\}$
    $IN_G(v) \leftarrow G(N_v, E_{N_v})$ ▷ induced network of $N_v \cup v$
    $m \leftarrow size(IN_G(v))$ ▷ induced network size
    if $m \leq l$ **then** ▷ if size less that limit l
      $\{C_1, .., C_k\} \leftarrow \mathcal{E}xact_{CE}(IN_G(v))$ ▷ clusters generated by ...
      ▷ ... solving exact CE by using integer program
      $\mathcal{I}ntegrate(C)$ ▷ integrate each clusters $C \in \{C_1, ..., C_k\}$ in ....
      ▷ ... G as a single node
    else ▷ if size not less that limit l
      ${}_lN_v \leftarrow$ randomly selected l nodes from $N_v \cup v$
      ${}_lIN_G(v) \leftarrow$ induced graph for ${}_lN_v$
      $\{C_1, .., C_k\} \leftarrow \mathcal{E}xact_{CE}({}_lIN_G(v) \leftarrow)$ ▷ clusters generated by ...
      ▷ ... solving exact CE by using integer program
      $\mathcal{I}ntegrate(C)$ ▷ integrate each clusters $c \in \{c_1, ..., c_k\}$ in ....
      ▷ ... G as a single node
    $V \setminus \{N_v \cup v\}$ ▷ remove all nodes in $IN_G(v)$ from V
  $C_{final} \leftarrow \{C_1, .., C_q\}$ ▷ set of all clusters
  return C

Integrating Clusters to the Weighted Network

In the third step, we execute the procedure $\mathcal{I}ntegrate(C)$, where $C \in \{C_1, ..., C_k\}$, to integrate the clusters $C_1, ..., C_k$, obtained from the previous step to the original weighted network. To do so, we consider each cluster C_i as a single node in the network. Suppose there exist two links $(v_1, u), (v_2, u) \in E$ with respective weights $w_{(v_1u)}, w_{(v_2u)} > 0$ such that $v_1, v_2 \in C_i$ and $u \notin C_i$. Then, after integrating the cluster C_i in the weighted network, the weight of the new link between the node C_i and u can be calculated as $w_{(uC_i)} = w_{(v_1u)} + w_{(v_2u)}$. The algorithm repeats this process for integrating all obtained clusters to the original weighted network.

3.3.4 Implementation of the Heuristic Algorithm for CE on Weighted Network

We implemented `HACEWN` by using Java and JGraphT graph package (Michail et al.; 2020) to solve the integer program Eqs. (3.5)-(3.8) IBM Cplex 12.10 for each execution of the procedure $\mathcal{E}xact_{CE}$.

3.3.5 Analyzing Identified Communities in Merged CKN

We applied four community clustering algorithms, including `HACEWN`, on the merged CKN from 2011 to 2016. Table 3.4 shows the sizes of the top ten clusters obtained from these four community clustering algorithms. We noticed that `HACEWN` provides the least variability in cluster sizes. From these clustering algorithms' outputs and our domain knowledge, we found that `HACEWN` gives meaningful clusters corresponding to the research communities affiliated with an education institution. Therefore, in the rest of the study, we focused on presenting further analysis based on the clusters obtained from `HACEWN` .

Using `HACEWN`, we identified 291 communities from the merged CKN from 2011 to 2016. To focus on more significant research groups, we ignored all communities with less than ten authors. This step resulted in 30 communities of authors in the merged CKN. In Table 3.5, we presented the top ten authors (based on their degree in the merged CKN) associated with the ten largest communities among the 30 identified communities. Also, in Table 3.6, we give the top productive author (in terms of the number of publications) and their total number of publications in 2011-2016 corresponding to top largest identified communities.

From a practical point of view, an individual author can use these two types of tables to identify other researchers who share similar knowledge of expertise and who are most productive in their common-knowledge-based community, which can influence a future collaborative project. University research officials can use this data to inform them in forming groups for developing grant proposals for national funding competitions.

3.3.6 Dominant Research Topics and Authors' Affiliations in the Identified Communities

In this part, we focused on recognizing the dominant research topics for each of the identified communities. Table 3.7 shows the number of authors (size), the number of unique keywords, and the possible dominant research topics (based on the manually observing all unique keywords) corresponding to the thirty identified communities. We observed

Cluster-ID	Cluster Sizes			
	Fast Unfolding (Modularity)	**Girvan-Newman (Edge Betweenness)**	**Leiden Clustering (Modularity)**	**HACEWN**
1	3546	8876	8879	1833
2	2926	6	7	912
3	2261	5	6	305
4	265	4	5	213
5	4	4	4	117
6	3	4	4	95
7	3	4	4	86
8	3	3	3	62
9	3	3	3	61
10	3	3	3	49

TABLE 3.4: Ten largest communities (Cluster-ID) and their corresponding sizes given by the four studied clustering algorithms. The cluster size represents the number of authors in the corresponding cluster.

Cluster-ID	Top Ten Authors Names (based on the degree in CKN)
1	Thabane L.; Yusuf S.; Bhandari M.; Schuenemann H. J.; Beyene J.; Guyatt G. H.; Guyatt G.; Cairney J.; Cook D. J.; Adachi J. D.
2	Cunningham C.; Lonn E.; McCabe R. E.; McDonald S. D.; Meade M. O.; Mehta S. R.; Mertz D.; Sharma M.; Shaw E.; Yusuf S.
3	Brennan J. D.; Brook M. A.; DeMatteo C; Dunn J R.; Eikelboom J. W.; Ellis P. M.; Lavis J. N.; Richardson J.; Simunovic M.; Szatmari P.
4	Arthur H.; Chaudhry H.; Deal K.; Dokainish H.; Miller J.; Pond G.; Tang A.; Velianou J. L.; Velikonja D.; Yousefi-Nooraie R.
5	Alhazzani W.; Bos D.; Gandhi S.; Kean W. F.; Levine M.; Li G.; Mansouri A.; Rowa K.; Samiee-Zafarghandy S.; Wang J.
6	Chen S.; Gonzaga F.; Inman M. D.; Marshall D.; Moffat K. A.; Schwarcz H. P.; Wang Y.; Warren L. A.; Wilson M. N.; Wojcik J.
7	Bates S. M.; Ghert M.; Ha V.; Haines T.; Hillis C.; Julian J.; Khan A.; Linkins L.; Schulman S.; Warkentin T. E.
8	Bolli P.; Braga L. H. P.; Gangji A.; Harlock J.; Leontiadis G. I.; Markle-Reid M.; Sne N.; Valettas N.; Wismer D.; Wong K. M.
9	Alklabi A.; Bain A. D.; Deen J.; Gohel T.; Mitchell C. J.; Naidoo, A.; Shaler C. R.; Wang K.; Winegard K. J.; Zhang T.
10	Fayed N.; Chan T.; Findlay S.; Ismaila A, S.; Jeremic A.; Teo K.; Santaguida P. L.; Arora S.; Natarajan M.; Richardson J. D.

TABLE 3.5: Top ten authors names (based on the degree) corresponding to the ten largest identified communities (Cluster-ID) from the merged CKN over 2011 to 2016.

Cluster-ID	Author Name	# Of Publications
1	Thabane L.	252
2	Yusuf S.	163
3	Eikelboom	170
4	Pond G. R.	125
5	Levine M.	39
6	Marshall D.	38
7	Schulman S.	168
8	Braga L. H.	37
9	Deen J.	95
10	Teo K.	70

TABLE 3.6: Most productive author from the ten largest communities (Cluster-ID) over the period 2011–2016 and their corresponding number.

Cluster-ID	#of Authors	#of Unique Keywords	Dominant Research Topic
C01	1833	29147	Children and women cancer
C02	912	12029	Children and women cancer
C03	305	6839	Material science: nanoparticles
C04	213	2134	Climate changes
C05	117	1433	Kinesiology
C06	95	1544	Gastrointestinal diseases
C07	86	1127	Trauma and mental disorder
C08	62	1070	Children and infants diseases
C09	61	1050	Cardiovascular disease
C10	49	913	Early childhood mortality & diseases
C11	42	527	Memory and brain research
C12	40	891	Women's pregnancy & child's health
C13	35	695	Deformation of ice glaciers in the arctic region
C14	34	446	Canadian geology
C15	34	949	Immunization and vaccine
C16	31	347	Radiation therapy
C17	31	708	Child blood pressure
C18	31	403	Study of a trout
C19	30	449	Semiconductor
C20	27	568	Fish population in Hamilton harbour area
C21	26	351	Electronics
C22	26	701	Crystal structured chemistry
C23	24	493	Reproduction and infections
C24	24	349	Clinical testing mouse model
C25	23	320	Astrophysics
C26	22	467	Study on Oscar fish
C27	22	292	Arthritis and osteotomy
C28	22	279	Bone connected neurons and control
C29	21	326	Pacemaker and its usability
C30	21	349	Hereditary and III-deficiency

TABLE 3.7: The size (number of authors), the number of unique keywords, and possible dominating research topics (based on top 20 frequent keywords) corresponding to the top thirty identified communities from merged CKN over 2011-2016.

C01		C02		C03		C04		C05	
Depts./ Research Inst.	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**
Medicine	111	Biology	49	Material Science & Engineering	29	Mechanical Engineering	18	Kinesiology	17
Inst. Infectious Disease Research	92	Geography & Earth Science	41	Physics & Astronomy	25	Material Science & Engineering	15	Medicine	16
Physics & Astronomy	79	Pediatric	39	Chemistry & Chem Biology	20	Computing & Software	12	Civil Engineering	8
Pathology & Molecular Medicine	76	Chemistry & Chem Biology	37	Chemical Engineering	20	Geography & Earth Science	10	Pediatrics	8
Chemistry & Chem Biology	65	Medicine	28	Engineering Physics	18	Physics & Astronomy	10	Biology	5
Kinesiology	64	Inst. Infectious Disease Research	27	Pathology & Molecular Medicine	14	Engineering Physics	9	Electrical & Computer Engineering	5
Biology	56	Clinical Epidemiology & Biostat	26	Inst. Infectious Disease Research	13	Psychiatry & Behavioural Neuroscience	8	Pathology & Molecular Medicine	5
Surgery	53	Kinesiology	26	Mechanical Engineering	10	Psychology, Neuroscience & Behaviour	7	Surgery	4
Psychiatry & Behavioural Neuroscience	50	Pathology & Molecular Medicine	23	Medicine	10	Chemistry & Chem Biology	4	Chemistry & Chem Biology	3
Electrical & Computer Engineering	40	Psychiatry & Behavioural Neuroscience	23	Psychiatry & Behavioural Neuroscience	10	Medicine	4	Clinical Epidemiology & Biostat	3
Other Depts./Inst.	580	Other Depts./Inst.	373	Other Depts./Inst.	0	Other Depts./Inst.	0	Other Depts./Inst.	36
No Affiliation Information	607	No Affiliation Information	220	No Affiliation Information	57	No Affiliation Information	62	No Affiliation Information	7
C06		**C07**		**C08**		**C09**		**C10**	
Depts./ Research Inst.	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**	**Depts./ Research Inst.**	**# of Authors**
Geography & Earth Science	11	Biology	22	Biochemistry & Biomedical Science	9	Medicine	11	Psychiatry & Behavioural Neuroscience	11
Biology	9	Inst. Infectious Disease Research	22	Inst. Infectious Disease Research	9	Clinical Epidemiology & Biostat	4	Geography & Earth Science	6
Farncombe family Digest Health Res. Inst.	8	Biochemistry & Biomedical Science	9	Pathology & Molecular Medicine	7	Health Science	4	Pediatrics Surgery	6
Biochemistry & Biomedical Science	6	Psychiatry & Behavioural Neuroscience	9	Rehabilitation Science	6	Population Health Res. Inst.	4	Computing & Software	4
Rehabilitation Science	6	Chemistry & Chem Biology	7	Medicine	5	Pathology & Molecular Medicine	2	Population Health Res. Inst.	4
Chemical Engineering	4	Nephrology	7	Health Science	4	Cardiac Arrhythmia	1	Inst. Transport & Logistics	3
Clinical Epidemiology & Biostat	4	Pathology & Molecular Medicine	3	Thrombosis & Atherosclerosis Res Inst.	4	Biology	1	Medicine	3
Medicine	4	Chemistry & Chem Biology	2	Clinical Epidemiology & Biostat	3	Chanchlani Research Centre	1	Health Science	3
Psychiatry & Behavioural Neuroscience	3	Juravinski Cancer Center	1	Psychiatry & Behavioural Neuroscience	3	Chemistry & Chem Biology	1	Pediatrics & Pediatrics gastroenterol	1
Electrical & Computer Engineering	2	Mathematics & Statistics	1	Pediatrics	2	DBCVS Research Inst	1	Child Health & Exercise Medicine	1
Other Depts./Inst.	0	Other Depts./Inst.	3	Other Depts./Inst.	10	Other Depts./Inst.	6	Other Depts./Inst.	2
No Affiliation Information	5	No Affiliation Information	0	No Affiliation Information	0	No Affiliation Information	26	No Affiliation Information	5

TABLE 3.8: Top 10 list of affiliated departments or research institutes for authors corresponding to the top 10 identified research communities $C01, ..., C10$.

that out of the thirty-one communities, the authors associated with six communities ($C01, C02, C08, C10, C12, C17$) were involved in research that is closely connected with women and children health-related issues. These results were validated by the fact that McMaster University is one of the world-leading institutions in the area of women and children's health sciences, as well as the fact that McMaster University houses one of the leading children's hospitals in Canada. It is also worth noting that our analysis did not account for mediating factors such as how common it is to have multiple authors in certain fields. This may explain why some of the top clusters are in fields where research is done in labs and papers often have a large number of authors.

In this section, for each identified community, we also analyzed authors' affiliations (e.g., departments, research centers, institute) inside the university. In the merged CKN, among 9,022 authors, we identified department affiliation for 8,009 authors. And for the rest of 1,013 authors, we did not find any specific affiliation information other than McMaster University. In this case of multiple affiliations for an author, we only considered their primary department/institute. Table 3.8 shows the top ten affiliations with corresponding authors numbers for the first ten identified clusters ($C01, ..., C10$). We observe that authors from multiple departments/research institutes clustered together based on their common-knowledge elements. For example, from Table 3.7 , the dominant research topic in $C01$ is "Children and Women Cancer." In Table 3.8, we see that besides authors from health science departments (e.g., Medicine, Inst. of Infectious Diseases, Surgery, etc.), authors from across disciplines (e.g., Chemistry & Chemical Biology, Electrical & Computer Engineering) are also included in this cluster.

From a practical insight viewpoint, our findings in this section can be used to define the leading focus areas for an institution's strategic research plan. By performing such analysis, an institution can also uncover ways for positioning the institution in terms of funding to promote collaborations across different disciplines (departments, research institutions). This is particularly important in current times when the funding agencies are advocating differentiation and a metric-based approach as a basis for funding universities.

3.3.7 Collaboration Incident Counts in the Communities

Researchers in a knowledge community may have more tendency to collaborate (Israel et al.; 1998). We know that the weighted link in the collaboration network represents the number of collaboration incidents between two authors. We can define the community induced collaboration network (`ComSubCN`) as the subnetwork of CN for a selected

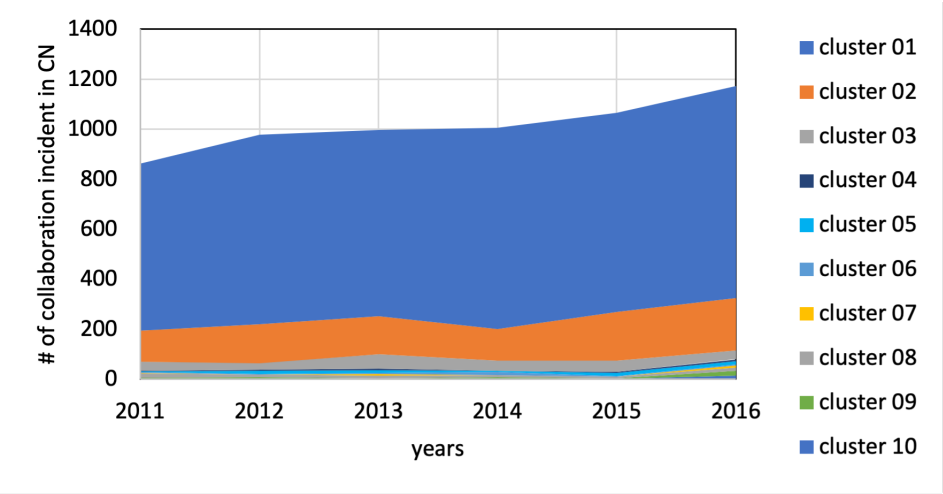


FIGURE 3.7: The variations in the number of collaboration incidents inside the ten largest communities over the period from 2011 to 2016.

subset of authors from the overall identified community. Therefore, for the set of authors associated with any community, the total weight of the links in the corresponding `ComSubCN`represents the total number of collaboration incidents among this community. For a CKN, the authors that belong to a community have more common-knowledge among each other. Having more common-knowledge may lead to having similar interests among the authors. These similar interests may positively effect the collaboration among the authors corresponding to the community over a period.

First, we investigated the variation of the collaboration incidents inside the ten largest communities (according to size) from 2011 to 2016. In Fig. 3.7, we notice that, overall, the number of collaboration incidents inside the communities increased over the years 2011-2016.. Therefore, we can conclude that the total number of collaboration incidents inside research communities increased over time.

3.3.8 Comparison of Collaboration and Common-Knowledge Network

One of the main aims of this study is to identify common-knowledge-based research communities in McMaster University to facilitate collaboration between researchers. In Table 9, we compared research communities produced by CN and CKN. CN represents the current research collaborations and CKN represents the possible future collaborations. Network visualizations of the top ten research communities (from Table 3.7) in CN and CKN are presented in Fig. 8.

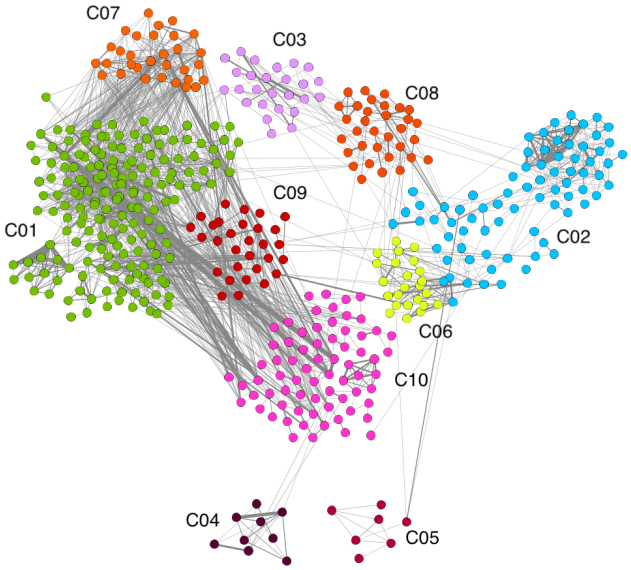
Cluster-ID	Number of Nodes (Authors)		Number of Links		Average Degree	
	CN	CKN	CN	CKN	CN	CKN
C01	1549	1833	164454	185490	4.4	202.39
C02	752	912	6737	8616	1.52	19.61
C03	279	305	1325	1398	1.35	9.48
C04	92	213	89	261	1.93	2.81
C05	67	117	53	151	1.58	2.16
C06	62	95	104	123	2.22	2.39
C07	40	86	47	227	2.35	4.83
C08	28	62	23	122	1.64	2.77
C09	30	61	21	298	1.4	7.64
C10	37	49	39	117	2.11	3.12

TABLE 3.9: Number of nodes (authors), links, average degree, and average weighted degree corresponding to top 10 identified research communities from the Table 3.7.

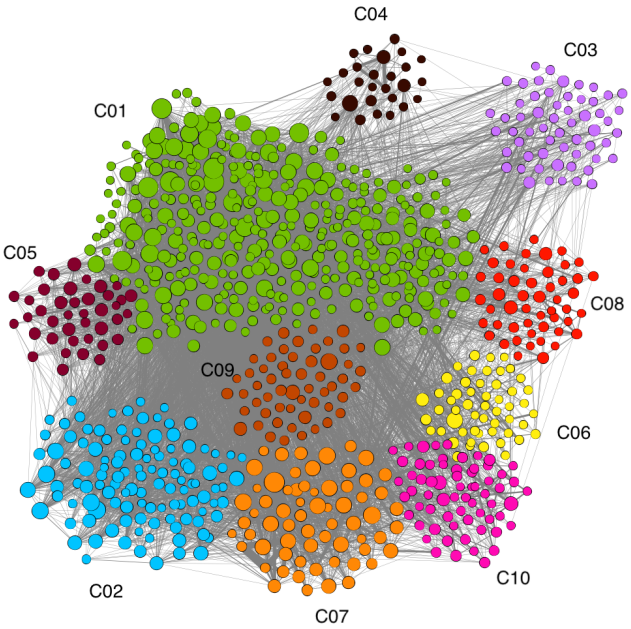
We noticed that the number of authors and links in CKN was larger than in CN for each identified research topic. It indicated that CKN provided richer communities. In the example of ten clusters in Fig. 8, CKN provided 12% more collaboration opportunities and involved 22% of researchers than CN. The increase in the average degree values, as much as 11.5 times higher in CKN, indicates that CKN found more collaboration opportunities and were also more connected.

3.4 Limitations

In this section, we focus briefly on the limitation of our study. The first limitation in our study was that we did not categorize different types of authors (e.g., Professor, Post Doctorate Researcher, Graduate Student, or Research associate). Besides professors, generally, all other researchers may only be at the university for a short time. However, our proposed common-knowledge network, as opposed to collaboration networks, would still identify possible collaborations between different full-time researchers as they would typically be co-authors with graduate students and post-doctoral fellows. Another limitation of our study is that we did not consider some discipline features that may impact the resultant CKN. For instance, some disciplines, such as medical and natural sciences, tend to have more authors per paper. In the future, to limit the effect of many more authors per paper, we can consider the first few authors, including the corresponding author. In addition, we can also regard the fact that the length of the paper and the review cycle are different across disciplines in the future.



(A) CN



(B) CKN

FIGURE 3.8: The merged (2011-16) (A) CN and (B) CKN for the McMaster University. The coloured clusters in networks are the top ten identified clusters $(C01, C02, ..., C10)$ mentioned in Table 3.7.

3.5 Conclusions and Future Research

We introduced the common-knowledge network (CKN) of authors affiliated with a research institution based on their mutual commonality in the keywords associated with their published articles. To identify CKNs, we propose the use of clustering editing (CE). We formulated CEs as integer programs and presented a heuristic algorithm, `HACEWN`, to solve large-scale CKNs. Motivated by a practical application in rationalizing universities' investments in interdisciplinary research, we applied our methodology to identify thirty communities among the authors (nodes) in a CKN for McMaster University over the years 2011 to 2016. We also identified the dominant research topics and authors' affiliations corresponding to each community based on unique keywords from the authors' published articles associated with this community. Afterwards, we studied the publication and collaboration incident counts for each identified community in the CKN. In addition, we proposed three hypotheses to capture different attributes of the CKN. Our study showed that collaboration incident counts corresponding to the identified communities in CKNs increase over time. We have also offered some practical managerial insights. In particular, our findings can aid universities' research officers in strategically investing in supporting targeted interdisciplinary research groups. We also find that current investments in interdisciplinary research tend to have an increasing marginal return, judging by the increase in the number of co-authored works, as inferred from CNs, as well as the topics, as inferred from the CKNs.

Our study can be extended in several ways. First, one can use our model to investigate the impact of strategic research investments on interdisciplinary research productively over time as well as perform inter-university comparative studies. Second, a researcher may get inactive or fail to publish any article due to different reasons such as retirement, job transfer, or lack of funding. This could affect the publication and collaboration incident counts for the identified communities in a CKN. In order to simplify this study, we assume a linear relationship between two factors: having a more common-knowledge and positive effect on the collaboration incidents. We think this casual relationship might be much more complicated and non-linear. In the future, a possible extension of this work is to examine such factors that may influence the publication and collaboration, incident counts. Another possible extension is to fuse the WoS data with ORCiD data. The latter is becoming more acceptable with researchers and provides additional reach data such as information of group grants. Finally, our study relied entirely on published works with no other secondary empirical data. One promising line of future research is to combine our bibliometric network analysis with a survey or interview of authors

to support and validate the proposed hypotheses as well as incorporate other relevant factors such as research grants, infrastructure and graduate student activity.

Declaration of Competing Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements We acknowledge support from Natural Sciences and Engineering Research Council (NSERC) Discovery and CREATE grants as well as Canada Foundation for Innovation (CFI), Ontario Research Fund (ORF), and Mitacs Accelerate Fellowship programs. We also like to acknowledge the feedback from anonymous reviewers that has significantly improved the quality of our paper.

Appendix A

Procedure of Removing Authors' Name Repetition We follow the steps in the process of removing authors' name repetition:

(i) *Finding distances between two authors' names*: We calculated the Levenshtein Distance between any two authors' names. The Levenshtein distance, first introduced by Levenshtein et al. (Levenshtein et al.; 1966), is a metric for measuring the difference between two string sequences. It computes the required minimum number of single-character edits (insertions or deletions) as the distance metric to convert one string to another.

(ii) *Identifying the same author-name with different formats*: We listed all authors-names in which the mutual Levenshtein distance is less than 5%, i.e., we catalogued all author-names which are at least 95% similar. We also listed the school/department name affiliation corresponding to each author-name format.

(iii) *Creating an author-name dictionary*: We investigated each of the 95% similar author-names lists individually and determined whether they represented the same author. Then we filtered all author-names formats representing the corresponding author and chose one name format to replace all of the other name formats. We created an author-name dictionary by logging the selected format and all other similar name formats.

(iv) *Clean publication data*: We cleaned the publication data by using the author-name dictionary. Therefore, in the final publication data, each author has precisely one name format.

Chapter References

Abbasi, A., Wigand, R. T. and Hossain, L. (2014). Measuring social capital through network analysis and its influence on individual performance, *Library & Information Science Research* **36**(1): 66–73.

Alpert, C. J., Kahng, A. B. and Yao, S.-Z. (1999). Spectral partitioning with multiple eigenvectors, *Discrete Applied Mathematics* **90**(1-3): 3–26.

Ardichvili, A., Page, V. and Wentling, T. (2003). Motivation and barriers to participation in virtual knowledge-sharing communities of practice, *Journal of Knowledge Management* .

Bansal, N., Blum, A. and Chawla, S. (2004). Correlation clustering, *Machine Learning* **56**(1): 89–113.

Biglu, M. (2008). The influence of references per paper in the sci to impact factors and the matthew effect, *Scientometrics* **74**(3): 453–470.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E. (2008). Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10): P10008.

Böcker, S. and Baumbach, J. (2013). Cluster editing, *Conference on Computability in Europe*, pp. 33–44.

Böcker, S., Briesemeister, S. and Klau, G. W. (2011). Exact algorithms for cluster editing: Evaluation and experiments, *Algorithmica* **60**(2): 316–334.

Bornmann, L. and Mutz, R. (2015). Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references, *Journal of the Association for Information Science and Technology* **66**(11): 2215–2222.

Brandes, U. (2005). *Network analysis: Methodological foundations*, Vol. 3418, Springer Science & Business Media.

Bringmann, L. F., Elmer, T., Epskamp, S., Krause, R. W., Schoch, D., Wichers, M., Wigman, J. T. and Snippe, E. (2019). What do centrality measures measure in psychological networks?, *Journal of Abnormal Psychology* **128**(8): 892.

Carnabuci, G. and Operti, E. (2013). Where do firms' recombinant capabilities come from? intraorganizational networks, knowledge, and firms' ability to innovate through technological recombination, *Strategic Management Journal* **34**(13): 1591–1613.

Chierichetti, F., Dalvi, N. and Kumar, R. (2014). Correlation clustering in mapreduce, *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 641–650.

Chopra, S. and Rao, M. R. (1993). The partition problem, *Mathematical programming* **59**(1): 87–115.

Clauset, A., Newman, M. E. and Moore, C. (2004). Finding community structure in very large networks, *Physical Review E* **70**(6): 066111.

Connelly, C. E., Zweig, D., Webster, J. and Trougakos, J. P. (2012). Knowledge hiding in organizations, *Journal of Organizational Behavior* **33**(1): 64–88.

Dahlander, L. and McFarland, D. A. (2013). Ties that last: Tie formation and persistence in research collaborations over time, *Administrative Science Quarterly* **58**(1): 69–110.

Delvaux, S. and Horsten, L. (2004). On best transitive approximations to simple graphs, *Acta Informatica* **40**(9): 637–655.

Demaine, E. D., Emanuel, D., Fiat, A. and Immorlica, N. (2006). Correlation clustering in general weighted graphs, *Theoretical Computer Science* **361**(2-3): 172–187.

Dhillon, I. S., Guan, Y. and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(11): 1944–1957.

Diaz, J. and Poblete, B. (2019). Car theft reports: a temporal analysis from a social media perspective, *Companion Proceedings of The 2019 World Wide Web Conference*, pp. 779–782.

Ding, Y. (2011). Scientific collaboration and endorsement: Network analysis of coauthorship and citation networks, *Journal of Informetrics* **5**(1): 187–203.

Flake, G. W., Tarjan, R. E. and Tsioutsiouliklis, K. (2004). Graph clustering and minimum cut trees, *Internet Mathematics* **1**(4): 385–408.

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

Garfield, E. (1990). KeyWords Plus-ISI's breakthrough retrieval method, *Current Contents* **32**: 5–9.

Garfield, E. and Sher, I. H. (1993). KeyWords Plus [TM]-algorithmic derivative indexing, *Journal-American Society For Information Science* **44**: 298–298.

Gaviria-Marin, M., Merigó, J. M. and Baier-Fuentes, H. (2019). Knowledge management: A global examination based on bibliometric analysis, *Technological Forecasting and Social Change* **140**: 194–220.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks, *Proceedings of the National Academy of Sciences* **99**(12): 7821–7826.

González-Álvarez, J. and Cervera-Crespo, T. (2017). Research production in high-impact journals of contemporary neuroscience: A gender analysis, *Journal of Informetrics* **11**(1): 232–243.

Grötschel, M. and Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem, *Mathematical Programming* **45**(1): 59–96.

Guan, J. and Liu, N. (2016). Exploitative and exploratory innovations in knowledge network and collaboration network: A patent analysis in the technological field of nano-energy, *Research Policy* **45**(1): 97–112.

Guan, J. and Zhang, J. (2018). The dynamics of partner and knowledge portfolios in alternative energy field, *Renewable and Sustainable Energy Reviews* **82**: 2869–2879.

Guan, J., Zhang, J. and Yan, Y. (2015). The impact of multilevel networks on innovation, *Research Policy* **44**(3): 545–559.

Gupta, N., Singh, A. and Cherifi, H. (2016). Centrality measures for networks with community structure, *Physica A: Statistical Mechanics and its Applications* **452**: 46–59.

Holme, P. and Saramäki, J. (2012). Temporal networks, *Physics Reports* **519**(3): 97–125.

Hu, Z., Lin, A. and Willett, P. (2019). Identification of research communities in cited and uncited publications using a co-authorship network, *Scientometrics* **118**(1): 1–19.

Huang, L., Liu, F. and Zhang, Y. (2020). Overlapping community discovery for identifying key research themes, *IEEE Transactions on Engineering Management* **68**(5): 1321–1333.

Israel, B. A., Schulz, A. J., Parker, E. A. and Becker, A. B. (1998). Review of community-based research: assessing partnership approaches to improve public health, *Annual Review of Public Health* **19**(1): 173–202.

Katsurai, M. (2017). Bursty research topic detection from scholarly data using dynamic co-word networks: A preliminary investigation, *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, IEEE, pp. 115–119.

Khasseh, A. A., Soheili, F., Moghaddam, H. S. and Chelak, A. M. (2017). Intellectual structure of knowledge in imetrics: A co-word analysis, *Information Processing & Management* **53**(3): 705–720.

Křivánek, M. and Morávek, J. (1986). Np-hard problems in hierarchical-tree clustering, *Acta Informatica* **23**(3): 311–323.

Lee, L.-F., Liu, X., Patacchini, E. and Zenou, Y. (2021). Who is the key player? a network analysis of juvenile delinquency, *Journal of Business & Economic Statistics* **39**(3): 849–857.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, Vol. 10, Soviet Union, pp. 707–710.

Li, E. Y., Liao, C. H. and Yen, H. R. (2013). Co-authorship networks and research impact: A social capital perspective, *Research Policy* **42**(9): 1515–1530.

Li, H., An, H., Wang, Y., Huang, J. and Gao, X. (2016). Evolutionary features of academic articles co-keyword network and keywords co-occurrence network: Based on two-mode affiliation network, *Physica A: Statistical Mechanics and its Applications* **450**: 657–669.

Liu, J., Li, Y., Ruan, Z., Fu, G., Chen, X., Sadiq, R. and Deng, Y. (2015). A new method to construct co-author networks, *Physica A: Statistical Mechanics and its Applications* **419**: 29–39.

Lozano, S., Calzada-Infante, L., Adenso-Díaz, B. and García, S. (2019). Complex network analysis of keywords co-occurrence in the recent efficiency analysis literature, *Scientometrics* **120**(2): 609–629.

Malmberg, A. and Maskell, P. (2002). The elusive concept of localization economies: towards a knowledge-based theory of spatial clustering, *Environment and Planning A: Economy and Space* **34**(3): 429–449.

McAssey, M. P. and Bijma, F. (2015). A clustering coefficient for complete weighted networks, *Network Science* **3**(2): 183–195.

McFadyen, M. A. and Cannella Jr, A. A. (2004). Social capital and knowledge creation: Diminishing returns of the number and strength of exchange relationships, *Academy of Management Journal* **47**(5): 735–746.

Michail, D., Kinable, J., Naveh, B. and Sichi, J. V. (2020). JGraphT—A Java library for graph data structures and algorithms, *ACM Transactions on Mathematical Software (TOMS)* **46**(2): 1–29.

Mohsen, M. A. (2021). A bibliometric study of the applied linguistics research output of saudi institutions in the web of science for the decade 2011-2020, *The Electronic Library* .

Muñoz-Leiva, F., Viedma-del Jesús, M. I., Sánchez-Fernández, J. and López-Herrera, A. G. (2012). An application of co-word analysis and bibliometric maps for detecting the most highlighting themes in the consumer behaviour research from a longitudinal perspective, *Quality & Quantity* **46**(4): 1077–1095.

Muñoz-Muñoz, A. M. and Mirón-Valdivieso, M. D. (2017). Analysis of collaboration and co-citation networks between researchers studying violence involving women., *Information Research: An International Electronic Journal* **22**(2): n2.

Nascimento, M. C. and De Carvalho, A. C. (2011). Spectral methods for graph clustering–a survey, *European Journal of Operational Research* **211**(2): 221–231.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices, *Physical Review E* **74**(3): 036104.

Olmeda-Gómez, C., Ovalle-Perandones, M.-A. and Perianes-Rodríguez, A. (2017). Co-word analysis and thematic landscapes in spanish information science literature, 1985–2014, *Scientometrics* **113**(1): 195–217.

Opsahl, T. and Panzarasa, P. (2009). Clustering in weighted networks, *Social Networks* **31**(2): 155–163.

Pan, X., Papailiopoulos, D., Oymak, S., Recht, B., Ramchandran, K. and Jordan, M. I. (2015). Parallel correlation clustering on big graphs, *Advances in Neural Information Processing Systems* **28**.

Phelps, C., Heidl, R. and Wadhwa, A. (2012). Knowledge, networks, and knowledge networks: A review and research agenda, *Journal of Management* **38**(4): 1115–1166.

Pons, P. and Latapy, M. (2006). Computing communities in large networks using random walks, *J. Graph Algorithms Appl.*

Price, D. J. D. S. (1965). Networks of scientific papers: The pattern of bibliographic references indicates the nature of the scientific research front., *Science* **149**(3683): 510–515.

Queiroga, E., Subramanian, A., Figueiredo, R. and Frota, Y. (2021). Integer programming formulations and efficient local search for relaxed correlation clustering, *Journal of Global Optimization* **81**(4): 919–966.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. and Parisi, D. (2004). Defining and identifying communities in networks, *Proceedings of the National Academy of Sciences* **101**(9): 2658–2663.

Rawlings, C. M., McFarland, D. A., Dahlander, L. and Wang, D. (2015). Streams of thought: Knowledge flows and intellectual cohesion in a multidisciplinary era, *Social Forces* **93**(4): 1687–1722.

Rigolon, A., Browning, M. H., Lee, K. and Shin, S. (2018). Access to urban green space in cities of the global south: A systematic literature review, *Urban Science* **2**(3): 67.

Schaeffer, S. E. (2007). Graph clustering, *Computer Science Review* **1**(1): 27–64.

Serrano, M. Á., Boguñá, M. and Pastor-Satorras, R. (2006). Correlations in weighted networks, *Physical Review E* **74**(5): 055101.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation, *IEEE Transactions on pattern analysis and machine intelligence* **22**(8): 888–905.

Traag, V. A., Waltman, L. and Van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities, *Scientific Reports* **9**(1): 1–12.

Tripathi, M., Kumar, S., Sonker, S. and Babbar, P. (2018). Occurrence of author keywords and keywords plus in social sciences and humanities research: A preliminary study, *COLLNET Journal of Scientometrics and Information Management* **12**(2): 215–232.

Wahid, D. F., Ezzeldin, M., Hassini, E. and El-Dakhakhni, W. W. (2022). Common-knowledge networks for university strategic research planning, *Decision Analytics Journal* **2**: 100027.

Wang, C., Rodan, S., Fruin, M. and Xu, X. (2014). Knowledge networks, collaboration networks, and exploratory innovation, *Academy of Management Journal* **57**(2): 484–514.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks, *Nature* **393**(6684): 440–442.

Zhang, C., Yu, Q., Fan, Q. and Duan, Z. (2013). Research collaboration in health management research communities, *BMC Medical Informatics and Decision Making* **13**(1): 1–13.

Zhang, J., Yu, Q., Zheng, F., Long, C., Lu, Z. and Duan, Z. (2016). Comparing keywords plus of wos and author keywords: A case study of patient adherence research, *Journal of the Association for Information Science and Technology* **67**(4): 967–972.

Zhao, W., Mao, J. and Lu, K. (2018). Ranking themes on co-word networks: Exploring the relationships among different metrics, *Information Processing & Management* **54**(2): 203–218.

Chapter 4

User-Generated Short Text Classification using Cograph Editing-based Network Clustering with an Application in Invoice Categorization

The content of this chapter is a revision of the manuscript text submitted for publication under the following title:

Wahid, D. F., Hassini, E. User-Generated Short Text Classification using Cograph Editing-based Network Clustering with an Application in Invoice Categorization. URL.

User-Generated Short Text Classification using Cograph Editing-based Network Clustering with an Application in Invoice Categorization

Dewan F. Wahid
School of Computational Science and Engineering
McMaster University, Hamilton, ON, Canada
Email: wahidd@mcmaste.ca

Elkafi Hassini
DeGroote School of Business
McMaster University, Hamilton, ON, Canada
Email: hassini@mcmaster.ca

Abstract

Rapid adaptation of online business platforms in every sector creates an enormous amount of user-generated textual data related to providing products or service descriptions, reviewing, marketing, invoicing and bookkeeping. These data are often short in size, noisy (e.g., misspellings, abbreviations), and do not have accurate classifying labels (line-item categories). Classifying these user-generated short text data with appropriate line-item categories is crucial for corresponding platforms to understand users' needs. This paper proposed a framework for user-generated short text classification based on identified line-item categories. In the line-item identification phase, we used cograph editing-based clustering on keywords network, which can be formulated from short texts. We also proposed integer linear programming (ILP) formulations for cograph editing on weighted networks and designed a heuristic algorithm to identify clusters in large-scale networks. Finally, we outlined an application of this framework to categorize invoices in an empirical setting. Our framework showed promising results in identifying invoice line-item categories for large-scale data.

Keywords: short-text classification; line-item category; invoice categorization; keywords network; network clustering; cograph editing

4.1 Introduction

With the advancement of technology and online-based business platforms, consumers are daily generating millions of electronic texts (Cevahir and Murakami; 2016) in different service industries' platforms for describing products, reviewing services, creating invoices and bookkeeping. Understanding users' inputs to design products and customize management plans accordingly is critical for business success (Zhu et al.; 2018; Greco and Polli; 2020). Any organization that studies and designs products around customers' needs gets significant market advantages compared to its competitors (Trivedi et al.; 2018; Liu et al.; 2019). There are, however, several challenges that analysts encounter while classifying short-texts.

Generally, user-generated short-texts in online platforms have a limited number of words (Inches et al.; 2010) and consist of different types of noises (e.g., nonstandard or misspelling, grammatical errors, abbreviations) (Hadar and Shmueli; 2021). Additionally, unlike long texts, user-generated short-texts lack contextual information (Song et al.; 2011) and semantic properties (Sriram et al.; 2010) due to word limitation, which creates a challenge in short-text classification problems. Furthermore, millions of user-generated short-texts appear daily in online platforms and marketplaces with new line-item categories and without accurate labelling (Cevahir and Murakami; 2016). Therefore, the lack of accurate line-item categories (classifying labels) for these generating short-texts creates another significant challenge to this classification problem (Hadar and Shmueli; 2021).

In general, traditional natural language processing (NLP)-based techniques such as 'Bags of Words (BoW)' and Latent Dirichlet Allocation (LDA) (Blei et al.; 2003)-based models ignore mutual relationships between keywords and do not perform well in short-text documents (Sriram et al.; 2010; Syed and Spruit; 2017). In addition, for short-text classification, large pre-trained language models tend to exhibit data sparsity quickly (Chen et al.; 2011).

In this paper, we proposed a framework for classifying user-generated short and noisy texts based on the mutual co-existing relationship of keywords. It also outlined a process of identifying line-item categories (classifying labels) for short-text classification using the Cograph Editing-based clustering on keyword network and cluster labelling. With this framework, we provided an integer linear programming (ILP) formulation for Cograph Editing on weighted networks with a procedure for calculating link insertion costs. In addition, we designed a heuristic algorithm to identify clusters in a large-scale network.

Furthermore, this paper outlined an application of the proposed framework in a project in which we partnered with a cloud-based invoicing and accounting services (CB-AIS) company to identify and classify invoices based on line-item categories. We used standard natural language processing (NLP) libraries to extract keywords from historical invoice data to formulate a keyword network for further clustering and identifying line-item categories.

The motivation for this project came from an application of the short-texts classification framework for identifying and categorizing invoices generated in cloud-based invoicing and accounting service (CB-AIS) platforms in real time when the line-item categories (accurate labels) are unknown. The *line-item* category for an invoice refers to the product or service category (e.g., web designing, landscaping) provided by the business (in this case, CB-AIS users) to its clients. Generally, in an invoice, the description field reflects the corresponding service or product category, a.k.a. line-item category. In recent years, most of the small and medium-sized enterprises (SMEs) have been using electronic invoices and accounting systems due to their efficiency and reliability for tracking and processing products, cost, revenue and taxes (Cedillo et al.; 2018). For CB-AIS companies to identify existing users' business line-item categories in their platforms, and classify users according to these categories, the commercial motivations include understanding users' needs, designing and offering customized products and target marketing(Hempstalk; 2017; Lesner et al.; 2019; Wang et al.; 2020; Liu et al.; 2021; Munoz et al.; 2022). In this project, we partnered with a CB-AIS company to apply the proposed shot-texts classification framework to identify and categorize invoices based on line-items.

The contributions of this paper are as follows:

i. A framework for classifying user-generated short-text based on identifying line-item categories using a keyword network with cograph editing-based network clustering.

ii. An ILP formulation for Cograph Editing on weighted networks and a process of determining the cost for all possible inserting links.

iii. A heuristic algorithm for Cograph Editing on large-scale weighted networks to identify keyword clusters.

iv. A case study to identify line item categories of the generating invoices in a CB-AIS platform in real-time. A keyword network was formulated from the invoice

descriptions to recognize line-item categories.

v. Maximum association probability and maximum associative cluster were formulated to determine the corresponding line-item category for new invoices. This simple process of categorizing new invoices can be done daily due to its computational simplicity.

The rest of this paper is organized as follows. In Section 4.2, we outline a brief literature review of related works. Section 4.3 presents a high-level architecture for the proposed short and noisy text classification framework. Next, we define a keyword network that can be formulated from the short-texts in Section 4.4. Section 4.5 presents Cograph Editing-based network clustering and its ILP formulations on weighted networks. In this section, we also illustrate a heuristic algorithm for Cograph Editing-based network clustering. Section 4.7 outlines an application of our proposed framework to identify and categorize invoices based on line-items. Finally, we discuss limitations, future works, and concluding remarks in Section 4.8.

4.2 Related Works

This section reviews relevant literature on short-text classification methodologies and invoice categorization.

4.2.1 Short-text Classification

Generally, 'text classification' deals with large documents containing rich content. Traditional natural language processing (NLP)-based techniques such as BoW and Latent Dirichlet Allocation (LDA) (Blei et al.; 2003) and its extensions perform well in large text documents because high word occurrences and frequencies are enough to capture the semantic properties (Sriram et al.; 2010). Kowsari et al. (2019) provided a survey on text classification problems. On the other hand, user-generated short-texts do not provide meaningful contextual information, often carry noise (e.g., misspelling) (Hadar and Shmueli; 2021), and the data sparsity problem arises quickly (Chen et al.; 2011). Furthermore, since general BoW-based models ignore mutual relations between words, these models are unsuitable for short-text classification problems (Sriram et al.; 2010).

In recent years, several studies proposed in the research literature for short-text classification for cases with and without accurate labelling. Different data processing, feature extractions and algorithmic approaches have been adopted in these studies. For example,

Škrlj et al. (2021) used a text2vec algorithm to construct taxonomy-based semantic features for classification, Alsmadi and Hoon (2019b) calculated term (keyword) weighting utilizing semantics classes for indexing, and Chua et al. (2019) used a semantic-based clustering approach. Hadar and Shmueli (2021) proposed an Ensembles Transferred Embedding framework that used a relatively small labelled data set to leverage the line-items from a larger data set. A review of early approaches to short-text classification can be found in Alsmadi and Hoon (2019a).

4.2.2 Invoice Categorization

The task of classifying/categorizing invoices using textual description is a desired feature on CB-AIS companies' platforms. Several CB-AIS platforms, such as QuickBooks and Xero, have been giving attention to categorizing invoice-line items (Hempstalk; 2017; Lesner et al.; 2019; Wang et al.; 2020; Liu et al.; 2021; Munoz et al.; 2022). However, we found that this growth in the industry interest in short-text classification has not been matched with an equivalent interest from the research community. This is largely due to the lack of publicly available data for analysis and benchmarking, given the strict confidentiality required when dealing with financial data (Cedillo et al.; 2018; Munoz et al.; 2022). On the other hand, CB-AIS companies have the advantage of having their own users' data, and as such, they are leading major developments in this area. However, several challenges associated with the automation of this process have been reported, especially for hefty invoice line-item categories and labelling invoices from new business contacts (suppliers or customers) (Hempstalk; 2017; Lesner et al.; 2019; Munoz et al.; 2022).

Three types of invoice categorization problems can be observed in the research literature: account code suggestion (ACS), false/fraud invoice detection, and line-item-based categorization (see Fig. 4.1). In the ACS problem, the task is to predict an account code (categories) invoice based on a predefined chart of accounts (CoA). This problem is part of bookkeeping that involves recording, categorizing, and keeping track of financial transactions (Munoz et al.; 2022). There are two main ACS approaches: hierarchical classification (e.g., see Munoz et al. (2022); Hedberg (2020); Bardelli et al. (2020); Liu et al. (2021); Hamza et al. (2007)) and multi-class machine learning (e.g., see Beļskis et al. (2020); Bergdorf (2018); Bengtsson and Jansson (2015)). Detecting fraud or false invoices using data mining techniques is presented in González and Velásquez (2013) and Wang et al. (2020). Line-item-based categorizations primarily used textual attributes or descriptions and treated the problems as text classification tasks. Thus, the related literature used the approaches described in Section 4.2.1(Hadar and Shmueli; 2021).

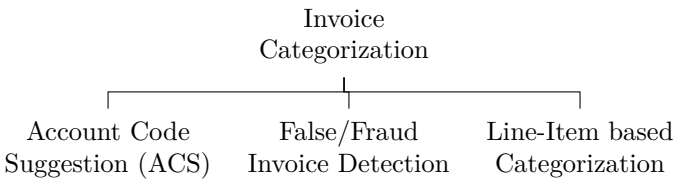


FIGURE 4.1: Types of invoice categorizations.

4.3 Proposed Framework

The proposed framework for classifying user-generated short-text can be divided into two phases: model building and production phases. In the model-building phase, we used historical data to identify line-item categories for short-text classification. There are three parts in the model-building phase: data collection, data processing, and clustering. In the data processing part, we used an NLP-based keyword extraction process from historical short-texts and formulated a keyword network as explained in Section 4.4. Next, we used a Cograph Editing-based network clustering to identify clusters from the formulated keyword network. Finally, we labelled each cluster by a category name with the aid of human experts and saved them as a model.

In the production phase, in addition to using keyword processing and the saved model, we used the metrics *maximum association probability* (defined in Eq. 4.9) and *maximum associative cluster* (defined in Eq. 4.10) to identify the maximum probability and corresponding line-item category cluster for a given short-text data. A high level of this framework architecture is presented in Fig. 4.2.

4.4 Keyword Network

We propose using a keyword network (KN) in the above framework to identify item categories for user-generated short-text classification. The concept of using KNs can be observed in many studies (Beliga et al.; 2015). Regarding topology, KN is a network or graph in which each node represents a unique keyword (obtained from documents or other data sources), and each link represents some level of a mutual relationship between two keywords (nodes). The definition of the mutual relationship-based link varies in different analyses. For example, in a keyword network-based patent analysis, Choi and Hwang (2014) used text mining to extract keywords from patent applications and defined links as to whether a patent was applied or not. In another study, Yoo et al.

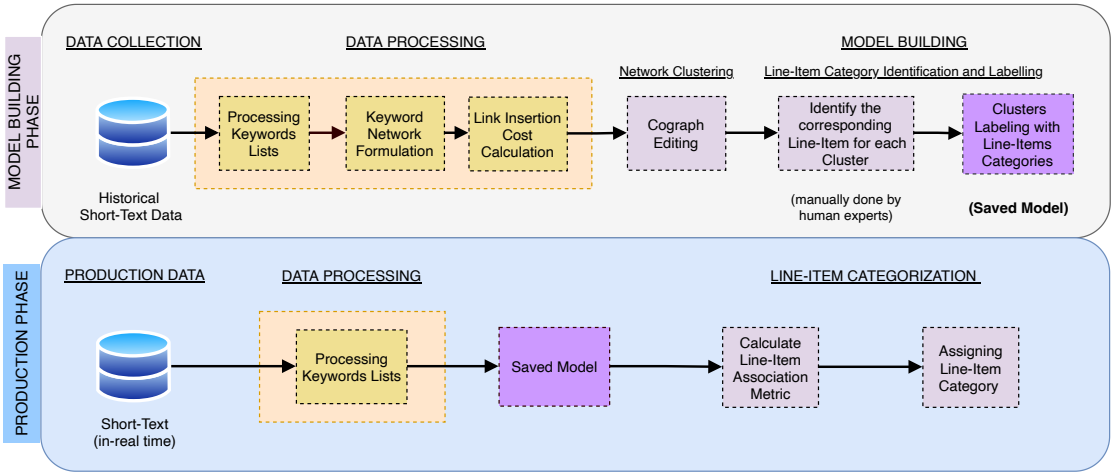


FIGURE 4.2: A high-level architecture for user-generated short-text classification framework.

(2019) used the keyword co-occurrence incident as the link definition to analyze human resource development research themes.

To define KN in this project, we first defined a binary *text-keyword incident matrix* $IM = (c_{ij}); i = 1, ..., n; j = 1, ..., m$, based on processed keywords from a given short-text collection. In this matrix, each row $i = 1, .., n$ represents a short-text from the collection, and each column $j = 1, ..., m$ represents a unique keyword. Each non-zero entry c_{ij} (*text-keyword incident*) in IM represents the fact that keyword K_j exists in short-text T_i. For each column in IM (unique keyword), a node is defined in KN. Any non-zero row between two distinct columns in IM represents a link between two corresponding nodes. The total number of such non-zero rows represents the weight of the corresponding link in KN. For any two distinct keywords (columns) p and q, the total number of *text-keyword incidents* (link weight in KN) w_{pq} can be defined as follows:

$$w_{pq} = \sum_{i=1}^{n} c_{pi} c_{qi} \tag{4.1}$$

where $c_{pi} = 1$ if keyword K_i exists in short-text T_p, otherwise $c_{pi} = 0$, and similarly $c_{qi} = 1$ if keyword K_i exists in short-text T_q, otherwise $c_{qi} = 0$. An example of the text-keyword incident matrix and KN formulations are illustrated in Fig. 4.3.

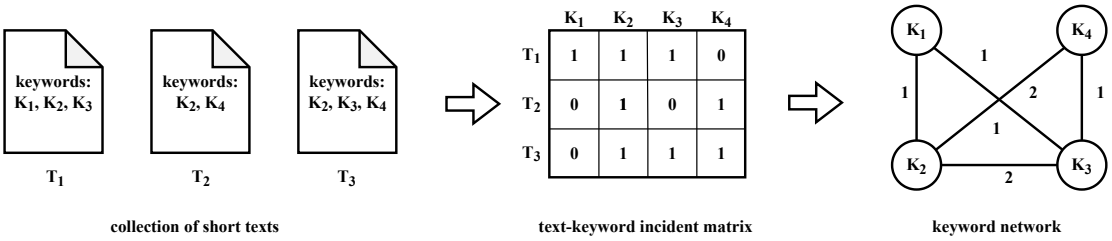


FIGURE 4.3: The *text-keyword incident matrix* and KN formulation from three given short-texts. In this figure, T_1, T_2 and T_3 are the short-texts, and K_1, K_2, K_3, and K_5 are the unique keywords associated with these short-texts.

4.5 Network Clustering

Network clustering is a technique that groups *strongly connected* nodes (objects) based on their mutual relationships (links). The desirable properties of identification of strongly connected groups can be varied, such as having particular structural properties (Shamir et al.; 2004), high density or sharing similar features (Newman and Girvan; 2004; Brandes; 2005). The problem of identifying clusters from a network appears under different names in the research literature, such as Community Detection, Community Clustering, Network/Graph Clustering, and Network/Graph Partitioning (Schaeffer; 2007; Fortunato; 2010). Note that we used the terms graph (subgraph) and network (subnetwork) interchangeably in the rest of this paper. Before moving forward, here are some graph theoretic definitions that will be needed to discuss some concepts in network clustering.

Definition 4.1 *A clique (a particular triad configuration) is a subset of a network such that every two distinct nodes in the clique are adjacent; that is, its induced subnetwork is complete.*

Definition 4.2 *Let P_k and C_k denote the cordless path and cycle on k vertices, respectively (see examples in Figs. 4.4 (a) and 4.4(b)). If G and H are two networks, then G is H-free if no induced subnetwork of G is isomorphic to H. In other words, H is the forbidden structure in the induced subnetwork of G.*

To identify and interpret clusters in a network, the underlying structural definition inside these clusters needs to be defined (Homans et al.; 2017). Many such definitions have been proposed in the literature (Shamir et al.; 2004). For example, Davis and Leinhardt (1967) empirically tested all types of *triad* configuration (i.e., all possible link combinations of three connected nodes) and characterized all permitted and forbidden

underlying structures in the clusters (Davis and Leinhardt; 1967). Many studies used the clique (which is a particular *triad* configuration) as the underlying structural definition of clusters in the network clustering process (Fortunato; 2010; Mishra et al.; 2007). In other words, a clique is a P_3-free subnetwork, where P_3 is a cordless path of three nodes (see Figs. 4.4 (a) and 4.4(b)). The Clustering Editing (Křivánek and Morávek; 1986; Böcker et al.; 2011), one of the most well-known clustering problems, identify underlying clusters by using P_3 as the forbidden structure (i.e., P_3-free clustering) with minimal link editions (deleting or inserting). Therefore, we can alternatively define the Clustering Editing problem as follows:

Problem 1 P_3-Free Link Editing (aka Clustering Editing)
***Input*:** An unweighted network $G(V, E)$ and P_3 (forbidden structure).
***Task*:** Find a node partition set $\{\mathcal{P}_1, ..., \mathcal{P}_k\}$ in V such that each $\mathcal{P}_i \subseteq V$ is P_3-free (i.e., clique) in network $G^*(V, (E \cup E^+) \setminus E^-)$ with minimized $|E^+| + |E^-|$.

Some other well-known network clustering problems that use the P_3 as the forbidden structure are: Clustering Deletion (Shamir et al.; 2004), Maximal Cliques Clustering (Biswas et al.; 2013) and Correlation Clustering (Bansal et al.; 2004; Charikar et al.; 2005). Similar to these traditional clustering problems, in recent years, several network clustering approaches have used some combinations of 4 nodes as the forbidden structures to define the underlying clusters in networks. For example, Cograph Clustering (Seinsche; 1974) uses P_4 as the forbidden structure (i.e., P_4 free clustering) and Quasi-Threshold Clustering (Nastos and Gao; 2013) uses (P_4, C_4) as the forbidden structure (i.e., (P_4, C_4)-free clustering).

In order to generalize, let $\mathcal{F}$ be the class of forbidden structures. Therefore, $\mathcal{F}$-free link editing-based clustering problem can be defined as follows:

Problem 2 $\mathcal{F}$-Free Link Editing Clustering
***Input*:** An unweighted network $G(V, E)$ and the forbidden structure $\mathcal{F}$.
***Task*:** Find a node partition set $\{\mathcal{P}_1, ..., \mathcal{P}_k\}$ in V such that each $\mathcal{P}_i \subseteq V$ is $\mathcal{F}$-free in network $G^*(V, (E \cup E^+) \setminus E^-)$ with minimized $|E^+| + |E^-|$.

In our computational setting (shown later in Table 4.5), we observed that using P_3 as the forbidden structure resulted in more constrained and uneven sizes of clusters (very few are huge, and the rest are very small) in our empirical data. As shown in Table 4.5, HACEWN, a P_3-free algorithm, identified the minimum number of significant clusters, i.e., the majority of the reminding clusters were very small. On the other hand, our computational experiments confirm that using P_4 as the forbidden structure (which is a more relaxed version of defining underlying clusters) produced the most meaningful clusters from the network. Therefore, in this study, we opted to use P_4-free link editing-based clustering (i.e., P_4- as the forbidden structure) to identify clusters in the keyword network generated from short-texts. A P_4-free network is formally defined as follows:

Definition 4.3 *A cograph is a P_4-free network (i.e., P_4 is the forbidden structure) (Seinsche; 1974; Brandstädt et al.; 1999).*

In addition, the P_4-free link editing-based clustering is also formally known as the Cograph Editing problem. From this point, we only focus on discussing the Cograph Editing problem as the network clustering.

4.5.1 Cograph Editing-based Network Clustering

In cograph (P_4-free)-based clustering, the goal is to achieve a cograph as the underlying structure for each obtained cluster (Liu et al.; 2012). Several versions of the cograph-based clustering, such as Cograph Deletion (Gao et al.; 2013), Cograph Completion (Liu et al.; 2012), and Cograph Editing (Gao et al.; 2013) on both unweighted and weighted networks, appeared in the literature. For Cograph Deletion, only link deletion is allowed; for Cograph Completion, only link additions are allowed; and for Cograph Editing, both link addition and deletion are allowed. Due to the relaxation of the cograph definition and based on our computational experiments, we observed that the Cograph Editing approach produces the maximum number of meaningful clusters (line-item identified clusters). Therefore, in this paper, we focused on the Cograph Editing problem to identify clusters in networks.

The Cograph Editing problem has been applied in several studies to identify underlying structures from empirical networks (e.g., (Kühnl; 2014; Hellmuth et al.; 2015; Dondi et al.; 2017)). For a given network, Cograph Editing finds minimum link edition (insertion or deletion) such that the induced network corresponding to each cluster is a cograph (i.e., P_4-free) (Crespelle; 2021). Therefore, in terms of up to 4-nodes configurations, all permitted arrangements of links and nodes are acceptable except the P_4.

Fig. 4.4 presents all permitted and forbidden configurations of nodes and links in the Cograph Editing problem.

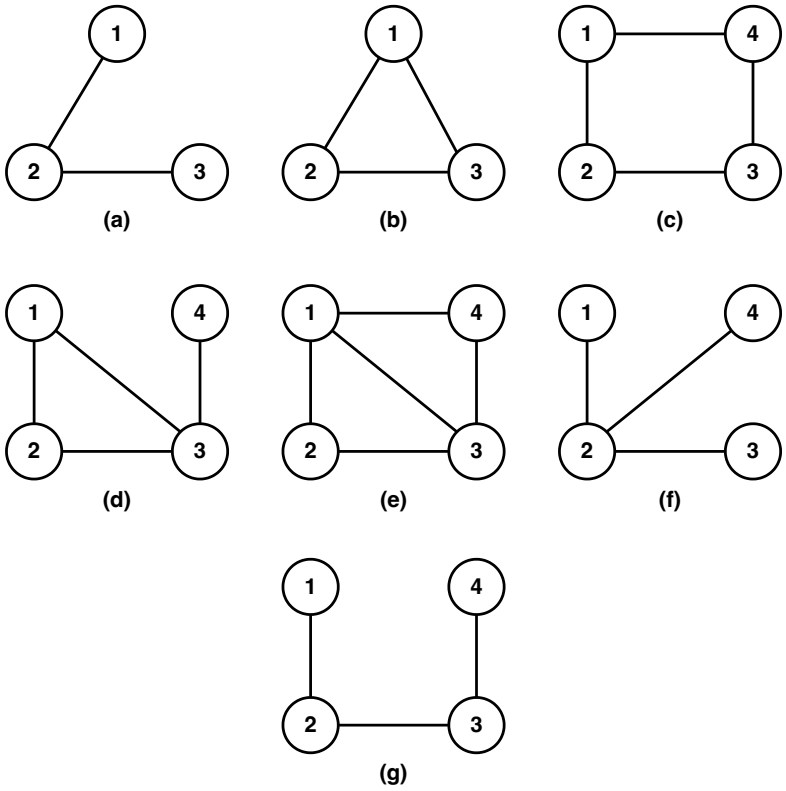


FIGURE 4.4: Example of (a) a P_3, (b) a clique (P_3-free), (c) a C_4, (d) paw, (e) diamond, (f) claw, and (g) P_4 subgraphs. Each link is associated with a positive weight. All subgraphs from (a)-(f) are permitted in Cograph Editing; only (f) is forbidden.

To formally define the Cograph Editing problem, assume an unweighted network $G(V, E)$, where V is the set of nodes and E is the set of links, and $|V| = n$. For a given network G, the goal of the clustering editing problem is to find a node partition set $\{\mathcal{P}_1, ..., \mathcal{P}_k\}$ in V with a minimum number of link modifications (deletion or insertion) such that each partition (cluster) $\mathcal{P}_i \subseteq V$ represents a cograph. Also, consider E^+ and E^- as the sets of all inserted and deleted links, respectively, to convert $G(V, E)$ to $G^*(V, (E \cup E^+) \setminus E^-)$, where each $\mathcal{P}_i \subseteq V$ is a cograph. In the modified network $G^*(V, (E \cup E^+) \setminus E^-)$, the number of partitions (clusters) k is arbitrary and an outcome of the optimization process in an unsupervised manner. Each link $(i, j) \in E$ in G represents

a mutual relationship between two corresponding nodes $i, j \in V$. Finally, we can define the Cograph Editing problem on G as follows:

Problem 3 Cograph Editing on Unweighted Network
***Input*:** An unweighted network $G(V, E)$
***Task*:** Find a node partition set $\{\mathcal{P}_1, ..., \mathcal{P}_k\}$ in V such that each $\mathcal{P}_i \subseteq V$ is a cograph (i.e., P_4-free) in network $G^*(V, (E \cup E^+) \setminus E^-)$ with minimized $|E^+| + |E^-|$.

A cograph can be recognized in linear time complexity (Corneil et al.; 1985). For a given at most k number of operations, Cograph Deletion, Cograph Completion and Cograph Editing problems are NP-complete (Liu et al.; 2012; El-Mallah and Colbourn; 1988) and also known to be fixed-parameter tractable (FPT) when combined with linear-time recognition of cographs (Cai; 1996). For arbitrary $k \geq 2$, Cograph Editing is NP-hard (Liu et al.; 2012; Hellmuth and Wieseke; 2018).

4.5.2 Cograph Editing Problem on Weighted Network

In this section, we extend the definition of the Cograph Editing problem (Problem 3) for weighted networks. In addition to the previous notations, to define this problem formally, assume the network $G(V, E)$ is weighted, V is the set of nodes, and E is the set of weighted links; and $|V| = n$. Each weighted link $(i, j) \in E$ (with wight w_{ij}) in G represents a mutual relationship between two corresponding nodes $i, j \in V$. Also, let $w(E^+)$ and $w(E^-)$ represent the total weights of all inserted and deleted links, respectively, to convert $G(V, E)$ to $G^*(V, (E \cup E^+) \setminus E^-)$, where each $\mathcal{P}_i \subseteq V$ is a disjoint cograph. Therefore, similarly, we can define the Cograph Editing problem on G (weighted) as follows:

Problem 4 Cograph Editing on Weighted Network
***Input*:** A weighted network $G(V, E)$
***Task*:** Find a node partition set $\{\mathcal{P}_1, ..., \mathcal{P}_k\}$ in V such that each $\mathcal{P}_i \subseteq V$ is a cograph (i.e., P_4-free) in network $G^*(V, (E \cup E^+) \setminus E^-)$ with minimized $w(E^+) + w(E^-)$.

P4-Free Conversion Link Insertion Cost Calculation

In clustering using Cograph Editing, the following two types of clustering costs arise: link insertion and deletion costs. For unweighted networks, both costs are considered as 1 (i.e., inserting a link or deleting a link has a unit cost) (Hellmuth et al.; 2015; Liu et al.; 2012). On the other hand, for a weighted network $G(V, E)$, a positive weight w_{ij} is associated with the link $(i, j) \in E$. Determining the link deletion cost is straightforward as the existing link's weight can be considered as the corresponding link deletion cost.

However, complexity arises when determining the link insertion cost in the absence of an existing link in the network. For clique-based clustering editing problems on a weighted network, several approaches have been suggested to calculate link insertion cost Serrano et al. (2006); Böcker et al. (2011); Böcker and Baumbach (2013); McAssey and Bijma (2015). Similar approaches have not been applied for determining link insertion costs in Cograph Editing on weighted networks. In this paper, we propose the following Algorithm 4.1 to calculate the link insertion cost to identify clusters using Cograph Editing in weighted networks.

According to the Cograph Editing problem definition, each node partition in the resultant network is a cograph (i.e., P_4-free). Therefore, the purpose of deleting any existing link from the P_4 or inserting any possible links in P_4 is to render it P_4-free. To calculate possible links' insertion costs using Algorithm 4.1, first consider $P_4^G = \{P_4^1, ..., P_4^t\}$, the set of all P_4s where $P_4^t \in P_4^G$ is the set of all links corresponding to P_4 in G.

Let $P_4^t \in P_4^G$ consist of nodes $i, j, k, l \in V$ and links $(i, j), (j, k), (k, l) \in E$ with corresponding links weights w_{ij}, w_{jk}, w_{kl}, respectively. Therefore, $P_4^t = \{(i, j), (j, k), (k, l)\}$. In terms of link insertion, inserting any of the possible links (i, k), (j, l), or (i, l) converts P_4^t to a cograph. Fig. 4.5 presents all possible link configurations to convert a P_4 to a cograph. Consider, $Q_{P_4}^t = \{(i, k), (j, l), (i, l)\}$, the set of all possible insertion links to convert P_4^t to a cograph. Consequently, $Q_{P_4}^G = \{Q_{P_4}^1, ..., Q_{P_4}^t\}$ is the set of all sets of all possible insertion links to convert $P_4^t \in P_4^G$ to a cograph.

To keep it simple, we consider equal insertion costs for any of the possible links (i, k), (j, l), (i, l) to convert P_4^t to a cograph. The insertion cost is calculated using Eq. (4.2):

Algorithm 4.1 Link Insertion Cost Calculation

1: **procedure** LINKINSERTIONCOST-($G(V, E)$)
2: $\delta_{ik} \leftarrow 0$; $\forall (i,k) \notin E$ ▷ assign zero (0) weight ...
3: ▷ ... to any non-existing link in G
4: $P_4^G = \{P_4^1, ..., P_4^t\}$ ▷ set of all P_4 in G
5: $Q_{P_4}^G = \{Q_{P_4}^1, ..., Q_{P_4}^t\}$ ▷ set of all possible insertion links to ...
6: ▷ ... convert corresponding $P_4^t \in P_4^G$ to a cograph
7:
8: `// iterate through each` P_4
9: **for** $P_4^t \in P_4^G$ **do**:
10: $P_4^t = \{(i,j), (j,k), (k,l)\}$ ▷ all links containing in P_4^t
11: $Q_{P_4}^t = \{(i,k), (j,l), (i,l)\}$ ▷ set of all possible insertion ...
12: ▷ ... links to convert P_4^t to a cograph
13:
14: `// calculate the link insertion cost`
15: $\delta_{ik}^t = \delta_{jl}^t = \delta_{il}^t \leftarrow \left\lfloor \frac{w_{ij}+w_{jk}+w_{kl}}{3} \right\rfloor$ ▷ w_{ij}, w_{jk}, w_{kl} are ...
16: ▷ ... corresponding link weights of $(i,j), (j,k), (k,l) \in E$
17:
18: `// maintaining minimum link insertion cost over` G
19: **if** $\delta_{ik}^t \leq \delta_{ik}$ **then**
20: $\delta_{ik} \leftarrow \delta_{ik}^t$
21: **if** $\delta_{jl}^t \leq \delta_{jl}$ **then**
22: $\delta_{jl} \leftarrow \delta_{jl}^t$
23: **if** $\delta_{kl}^t \leq \delta_{kl}$ **then**
24: $\delta_{kl} \leftarrow \delta_{kl}^t$

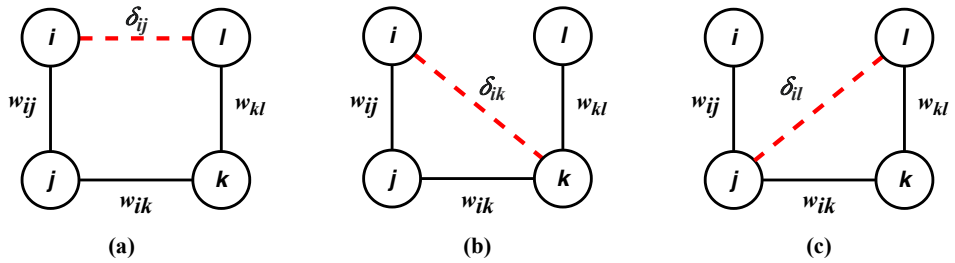


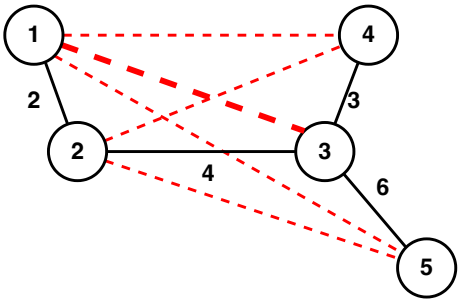
FIGURE 4.5: All possible link insertion configuration for converting $P_4^t \in P_4^G$ to a cograph (i.e., P_4-free). The dotted red lines (i,j), (j,k), (i,l) are possible inserted links and δ_{ik}, δ_{jl}, δ_{il} are corresponding links' insertion costs, respectively. We assumed all possible links' insertion costs for making a P_4 are equal (i.e., $\delta_{ik} = \delta_{jl} = \delta_{il}$).

$$\delta_{ik}^t = \delta_{jl}^t = \delta_{il}^t = \left\lfloor \frac{w_{ij}+w_{jk}+w_{kl}}{3} \right\rfloor, \tag{4.2}$$

where $(i,j),(j,k),(k,l) \in E; (i,k),(j,l),(i,l) \notin E$. Each possible inserting link $(i,k) \notin E$; $\forall i,k \in V$, can be part of converting more than one $P_4^t \in P_4^G$ to a cograph. Corresponding to each P_4^t there is a link insertion cost δ_{ik}^t. Therefore, the insertion cost for link $(i,k) \notin E$ can be calculated as the minimum of all δ_{ik}^t corresponding to the conversion of each $P_4^t \in P_4^G$ to a cograph:

$$\delta_{ik} = \min_{P_4^t \in P_4^G} \{\delta_{ik}^t \mid (i,k) \in Q_{P_4}^t\}. \tag{4.3}$$

An example of calculating possible link insertion costs for a given network is presented in Fig. 4.6. In addition, Fig. 4.7 presents an example of a Cograph Editing problem outcome on a weighted network using the above possible links insertion costs formulations.



(a)

	Possible Inserting Links				
P4	(1,3)	(1,4)	(1,5)	(2,4)	(2,5)
{1,2,3,4}	3	3		3	
{1,2,3,5}	4		4		4
Insertion Cost (Min)	3	3	4	3	4

(b)

FIGURE 4.6: Example of calculating all possible links' insertion costs for converting a given weighted network to a P_4-free (cograph). (a) There are two P_4 in this network and the set of all P_4 is $P_4^G = \{P_4^1, P_4^2\}$; $P_4^1 = \{(1,2),(2,3),(3,4)\}, P_4^2 = \{(1,2),(2,3),(3,5)\}$. Red dotted lines are possible links for converting P_4s to cograph. Therefore, $Q_{P_4}^1 = \{(1,3),(2,4),(1,4)\}$ and $Q_{P_4}^2 = \{(1,3),(2,5),(1,5)\}$ (b) In this table, first, we calculated the possible link insertion cost to make the corresponding P_4-free (using Eq. (4.2)). Since inserting link $(1,3)$ converts both P_4^1 and P_4^2 to cographs, it has two possible insertion costs. We consider the minimum of these two possible insertion costs for this link (column minimum).

Integer Programming Formulation for Cograph Editing on Weighted Network

Hellmuth et al. (2015) proposed an ILP formulation for Cograph Editing on genome-wide networks to resolve phylogenetic trees that were expanded in Hellmuth and Wieseke (2018). This paper proposes a modified ILP formulation for Cograph Editing on general-weighted networks with a link-insertion cost term based on Hellmuth et al. (2015)'s formulation.

Let w_{ij} be a positive weight associated with link $(i,j) \in E$ representing the text-keyword incident count between two keywords (nodes) $i, j \in V$. Therefore, in the process of node partitioning, w_{ij} can be considered as the deletion cost for link (i,j). On the other hand, for any possible link $(j,k) \notin E$, the insertion cost δ_{jk} can be calculated by using Eqs. (4.2) and (4.3). We define a binary decision variable x_{ij} assigned for each link $(i,j) \in E'$, where $i, j \in V$, as follows:

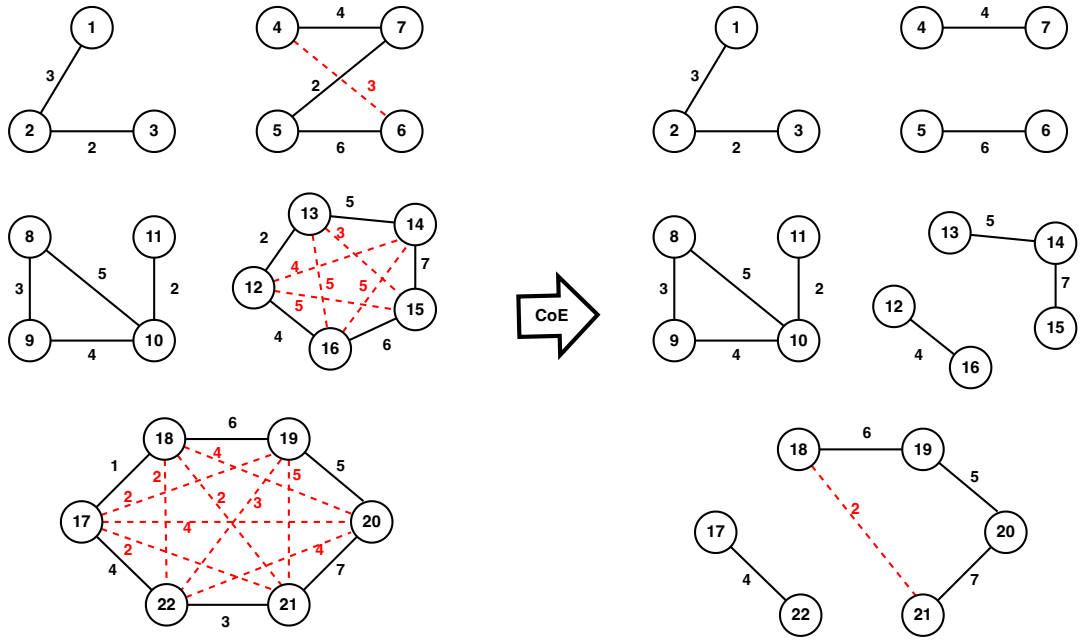


FIGURE 4.7: Example of Cograph Editing problem on the weighted network with 16 link editing costs (minimized). Deleted links are $(5,7)$, $(12,13)$, $(15,16)$, $(17,18)$, and $(21,22)$ with total cost: $2+2+6+1+3=14$. Only one inserted link $(18,21)$ with cost 2. Red dotted lines represent the possible links to make P_4-free with corresponding cost (showed as negative weight), and the solid red line represents the final added links.

$$x_{ij} = \begin{cases} 1; & \text{if } i \text{ and } j \text{ are in the same partition (cluster),} \\ 0; & \text{otherwise.} \end{cases} \tag{4.4}$$

The ILP formulation for the Cograph Editing problem on weighted network $G(V,E)$ can be formulated by using link edition to convert to the cograph network $G^*(V,(E\cup E^+)\setminus E^-)$ as follows:

$$\min \quad \sum_{(i,j)\in E} w_{ij}(1-x_{ij}) + \sum_{(i,j)\notin E} \delta_{ij}x_{ij} \tag{4.5}$$

$$\text{s. t.:} \quad x_{ij}+x_{jk}+x_{kl}-x_{ik}-x_{jl}-x_{il}\leq 2; \quad \forall i,j,k,l\in V, \tag{4.6}$$

$$x_{ij}=x_{ji}; \quad \forall i,j\in V, \tag{4.7}$$

$$x_{ij}\in\{0,1\}; \quad \forall i,j\in V, \tag{4.8}$$

where the objective function, Eq. (4.5), represents the link deletion cost (first part) and link insertion cost (second part) to convert the weighted network $G(V, E)$ to a cograph partitioned network $G^*(V, (E \cup E^+) \setminus E^-)$. The inequality constraint, in Eq. (4.6), enforces the condition of allowing only permitted structures from Fig. 4.4 and forbids P_4 in G^*. Eq. (4.7) represents the undirected properties of G, and finally Eq. (4.8) represents the binary requirement for x_{ij}.

4.5.3 Heuristic Algorithm for Cograph Editing on Weighted Network (HACoEWN)

To identify clusters by using Cograph Editing in a large-scale weighted network, we designed a heuristic algorithm called HACoEWN. Algorithm 4.2 presents the pseudo-code for HACoEWN. This algorithm has three main recursive steps that are described in the following subsections.

Algorithm 4.2 Heuristic Algorithm for Cograph Editing on Weighted Network (HACoEWN)

```
1: procedure HEURISTIC-(G(V, E))
2:     l ← constant                                   ▷ max. size of induced networks
3:
4:     // iterate through each node
5:     while v ∈ V do:                                ▷ select each node from vList
6:
7:         // ** Induce Network Selection **
8:         v ← rand(V)                                ▷ randomly select a node v ∈ V
9:         IN_G^v ←INDUCEDNETWORK(G, v)               ▷ induced network for..
10:                                                   ▷ ..given node v
11:        // ** Solve Exact Cograph Editing using ILP in Eqs. (4.5)-(4.8)
   **
12:        {C_1, ..., C_k} ← EXACTCOE (IN_G^v)
13:        C ← {C_1, ..., C_k}                        ▷ set of clusters generated by EXACTCOE
14:
15:        // ** Integrate Clusters **
16:        INTEGRATE (C)
17:
18:        // marking nodes as visited
19:        V ← V \ lN_v.                              ▷ mark all nodes in lN_v ..
20:                                                   ▷ .. as visited
21:    return
```

Induced Network Selection

In the first step of this algorithm, it randomly selects a node $v \in V$ and goes to subroutine Algorithm 4.3 to select the induced network corresponding to this node. In this subroutine, first, it creates $N_v \leftarrow \{u \in V \mid (u,v) \in E\} \cup \{v\}$, a set of all neighbours of v including v. Next, it identifies the strongest neighbour of v (maximum mutual link's weight) and the set N_u of all neighbours of u. If the size of set $N_{uv} = \{N_u \cup N_v\}$ is less than the given model threshold l, then the algorithm proceeds to the next step with an updated reference ${}_lN_{uv}$. Otherwise, it creates a set ${}_lN_{uv}$ by randomly selecting l elements from N_{uv} including u and v. The model threshold l can be determined based on the system capacity to solve the maximum size of the network using the ILP formulation given in Eqs. (4.5) - (4.8).

Algorithm 4.3 Induced network selection

```
1:  procedure INDUCEDNETWORK((G, v))
2:
3:      // neighbours list of node v'
4:      N_v ← {u ∈ V | (u, v) ∈ E}                    ▷ set of all neighbours of v
5:
6:      // strongest neighbour of node v'
7:      u ← {i | i ∈ N_v, w_iv ≥ w_jv; ∀j ∈ N_v}      ▷ w_iv is the..
8:                                                    ▷ ..weight of link (i, v) ∈ E
9:      // neighbours list of node u'
10:     N_u ← {i ∈ V | (u, i) ∈ E}                    ▷ set of all neighbours of u
11:
12:     // merging v and u's neighbours
13:     N_uv ← N_v ∪ N_u
14:     m ← |N_uv|                                    ▷ size of set N_uv
15:
16:     // check model threshold
17:     if m ≥ l then:
18:         lN_uv ← randomly selected l nodes from N_uv including u and v
19:     else
20:         lN_uv ← N_uv
21:
22:     // get the induced network for node v
23:     E_{N_uv} ← {(u, w) ∈ E | u, w ∈ lN_uv}        ▷ mutual links for nodes in lN_uv
24:     IN_G(lN_uv) ← G(lN_uv, E_{N_uv})              ▷ induced network of node set lN_uv
25:
26:     return IN_G(lN_uv)
```

Finally, based on the node set ${}_lN_{uv}$, the algorithm creates an induced network

$IN_G({}_lN_{uv}) = G({}_lN_{uv}, E_{N_{uv}})$ where $E_{N_{uv}} = \{(u,w) \in E \mid u,w \in {}_lN_{uv}\}$ is the set of all mutual links among the nodes in ${}_lN_{uv}$. Note that, $IN_G({}_lN_{uv})$ is a subnetwork of $G(V,E)$.

Solving Exact Cograph Editing on Induced Weighted Network

In the second step of this algorithm, it uses 'ExactCoE ($IN_G({}_lN_v)$ to solve the Cograph Editing problem on the induced weighted network ($IN_G({}_lN_v)$ by using the ILP formulation in Eqs. (4.5)-(4.8). Let, $\{C_1, ..., C_k\}$, where $C_k \subseteq N_v$ and $C_i \cap C_j = \emptyset, \forall i \neq j$, be the set of clusters returned by this function. ExactCoE can be implemented as another subroutine by using commercial optimization solvers such as Cplex or Gurobi.

Integrate Clusters as a Singleton Node

In the final step, clusters $C = \{C_1, ..., C_k\}$ are integrated from the previous step to the network G by using subroutine 'Integrate(C)' given in Algorithm 4.4. To do so, it adds a new single node u_v^i to represent the cluster C_i. Then it updates links between u_v^i and its all outer neighbours $p \in N_{C_i}$ where $N_{C_i} = \{v \mid (u,v) \in E, u \in C_i, v \in \{V \setminus {}_lN_v\}\}$. The newly updated link weight can be calculated as $w^* = w_{pC_i} - (w_{C_i}/3)$. Here, w_{pC_i} is the total link between node $p \in N_{C_i}$ and nodes in C_i, and w_{C_i} is the total existing links weight in C_i. Finally, it removes all nodes and previously connected links corresponding to the cluster C_i. The algorithm repeats this process for integrating each obtained cluster $C_i \in C$ to G.

4.6 Cluster Labelling and Maximum Associative Cluster

As mentioned above, the proposed framework aims to classify millions of user-generated short-texts appearing in online marketplaces with no accurate labelling (unlabelled) of line item categories. Two main semi-automatic approaches can be observed in the literature to deal with unlabelled texts: prioritizing text for manual labelling and manually assigning line-item categories to 'learn' to apply such categorizations automatically (Zhang and Zhong; 2016; O'Mara-Eves et al.; 2015; Thomas et al.; 2011).

In this paper, we proposed manually assigning line-item categories to each identified keyword cluster in our framework using Algorithm 4.2. In an empirical setting, this process should be done by subject-matter experts in the respective field. After labelling each identified cluster with the corresponding line-item categories, we saved this model to

Algorithm 4.4 Integrating clusters to the given weighted network

1: **procedure** INTEGRATE(C)
2:
3: `// iterate through all clusters`
4: **for** $C_i \in C$ **do**:
5: $u_v^i \leftarrow C_i$ ▷ create a single node in $G(V, E)$
6: $m \leftarrow |C_i|$ ▷ size of the cluster C_i
7:
8: `// update links`
9: $N_{C_i} \leftarrow \{v \mid (u, v) \in E, u \in C_i, v \in \{V \setminus {}_lN_v\}\}$ ▷ outer nbrs. ..
10: ▷ .. set for nodes in C_i
11: $E^* \leftarrow \{(u, v) \mid (u, v) \in E, u \in C_i, v \in \{V \setminus {}_lN_v\}\}$ ▷ links set ..
12: ▷ .. between outer nbrs. and nodes in C_i
13: $w_{C_i} \leftarrow \sum_{i,j \in C_i;\, (i,j) \in E} w_{ij}$ ▷ total links weight in cluster C_i
14:
15:
16: **for** $p \in N_{C_i}$ **do**:
17: `// calculate updated link weight`
18: $N_{C_i}^p \leftarrow p$'s all neighbours in cluster C_i
19: $w_{pC_i} \leftarrow \sum_{q \in C_i; (p,q) \in E} \; w_{pq};\; \forall q \in N_{C_i}^p$ ▷ weight for link (u_v^i, p)
20: ▷ w_{pq} is the weight for the link $(p, q) \in E$
21: $w^* \leftarrow w_{pC_i} - (w_{C_i}/3)$
22:
23: `// add update link and corresponding weight`
24: $G.add\_link\ ((u_v^i, p), weight\ =\ w^*)$ ▷ add link (u_v^i, p)
25:
26: `// remove clustered nodes and links from network`
27: $V \leftarrow V \setminus C_i$ ▷ remove nodes in C_i from G
28: $E \leftarrow E \setminus E^*$ ▷ remove links between nodes N_{C_i} and C_i from G

'learn' for future user-generated short-text classification. An application of this process is illustrated in Section 4.7.4.

To deploy this learned model ('saved model' in Fig. 4.2) in an empirical setting, we defined two simple metrics (with low computation complexity) to measure the association for evaluating short-text to a specific line-item category (labelled cluster). To define these metrics, let $C = \{C_1, ..., C_k\}$ be the set of all identified clusters, and K_T be the set of all keywords in the short-text T_i. We defined *maximum association probability* p_{max}, and *maximum associative cluster* C_T^* for identifying the line-item category for a short-text T as follows:

$$p_{max} = \max_{\forall\, C_i \in C} \{C_i(T)\}, \tag{4.9}$$

$$C_T^* = \arg\max_{\forall\, C_i \in C} \{C_i(T)\}, \tag{4.10}$$

where $C_i(T) = \frac{|\{u : u \in K_T \cap C_i\}|}{|K_T|}$ represents the number of common keywords between the cluster $C_i \in C$ and T. Eq. (4.10) represents the cluster C_T^* with maximum association probability for a given text T. We also calculated *non association probability* for text T, $p_{non}(T) = 1 - p_{max}$. Now if $p_{non} \leq p_{max}$, text T's line-item category was labelled as corresponding to cluster C_T^*'s label; otherwise, it is labelled as a *non-associative cluster*. If a given text T is 'non-associative', we labelled keywords in $k \in K_T$ as in a 'non-associative cluster' C_{non} and integrated them into the keyword network G.

In an application setting, this framework's retaining point can be determined by the size of C_{non}. One possible option can be when the size of C_{non} is greater than the total number of nodes (keywords) in G.

4.7 Invoice Line-Item Identification and Categorization

An invoice is a business document that lists details of products or services provided by a merchant to its consumer. In recent years, most businesses have been using electronic invoices and accounting systems due to their efficiency and reliability (Cedillo et al.; 2018). However, maintaining an electronic invoicing and accounting system is often expensive for many SMEs (Asatiani and Penttinen; 2015). Therefore, in order to maintain globalization, competitive advantages, cost savings and data security, many SMEs are shifting towards CB-AIS as an outsourcing option (Dimitriu et al.; 2014; Asatiani et al.;

2019). Studies show that adopting CB-AIS infrastructures significantly impacts companies' intellectual, human, and relation capitals (Cleary and Quinn; 2016; Kariyawasam; 2019). Following the increasing demand, many CB-AIS companies are coming to market, targeting SMEs worldwide (Eldalabeeh et al.; 2021). These companies are enabling more SMEs to undertake basic book-keeping tasks independently instead of paying external accountants (Ma et al.; 2021). Moreover, now SMEs have access to their data in real-time by using subscription or pay-per-use internet-accessible computing resources instead of keeping an expansive 'in-house' IT department (Christauskas and Miseviciene; 2012).

In order to get an advantage in the competitive marketplace, CB-AIS companies are trying to offer customized products based on their users' needs (Dimitriu et al.; 2014; Alshirah et al.; 2021). Besides other bookkeeping tasks, users (customers) in a CB-AIS platform generally create invoices, send them to their corresponding clients, and receive payment. In a recent trend, these companies are analyzing their users' invoices and categorizing line-items to design different features, understand customers' needs, and improve users' experiences (Hempstalk; 2017; Lesner et al.; 2019; Liu et al.; 2021; Munoz et al.; 2022). As a fast-growing CB-AIS platform, our partner company is also focusing on automating this process and developing invoice line-item category-based features for further analysis.

Typically in an invoice, the description text is the only field where the invoice creator explains the provided line item (or service); it is generally small and concise and can be classified as user-generated short-text (Hadar and Shmueli; 2021; Munoz et al.; 2022). This section presents an application of the proposed short-texts classification framework to identify line-items and categorize invoices from our research partner.

4.7.1 Data Collection and Processing

Invoice Dataset

The obtained invoice data were unlabelled and did not have any list of known line-item categories for classification. Our primary dataset contains approximately 1.8 million invoices. Each invoice in our dataset includes the following fields: User ID, Invoice ID, Creation Date, Invoice Description, Seller's Business Name, Amount, and Paid Status. During the initial data collection, we noticed some invoice description fields were blank, and some were with not-paid status. We also observed that trial users generally create blank and non-paid invoices to get familiar with the CB-AIS platform. Therefore, to

ignore such invoices, we filtered out any invoices with an empty invoice description field or not-paid status. A list of five representative example invoices is given in Table 4.1

Invoice ID	Creation Date	Invoice Description	Seller's Business Name	Amount	Paid Status
1	2019-01-05	SEO Monthly Foundation Building Plan	SciFi Webs	$2309	1
2	2019-03-18	RT +3.75 -2.00 090 ADD 2.50 SEG HT 20 PD 33.5/32.0	Adam Clear Vision Optical	$200	1
3	2019-03-06	Flower arrangement for the big oval table in the lobby	Eve Rose Flowers Trading Ltd	$890	1
4	2019-08-11	$30/per man hour	Jack Housekeeping Services	$156	1
5	2019-10-12	Preparation of General Warranty Deed	John Doe Law	$200	1

TABLE 4.1: Five example invoices and their attributes.

In the data in Table 4.1, the invoice description field is the user's defined text input without any validation or constraint. Fig. 4.8 presents a frequency histogram for the number of word counts per each invoice description. This figure shows that most invoice descriptions have less than ten words. Therefore, we can say that these text inputs are short in size. Due to user-generated text, these typically contain grammatical and spelling errors and abbreviations. In addition, some of the invoices are vague to understand (e.g., see 'Invoice ID 2' in Table 4.1). Based on these observations, the obtained invoice descriptions data can be labelled as noisy, as previously seen in Hadar and Shmueli (Hadar and Shmueli; 2021) and Baldwin et al. (Baldwin et al.; 2015) studies.

Removing Foreign Language Invoices

Our primary goal in this project was to categorize invoices with English language descriptions. In our initial collected data, approximately 13.34% of invoices had foreign language description fields. We used a foreign language filter built on the NLTK (Bird; 2006) English corpus to exclude such invoices.

Overcoming Inadequate Invoice Descriptions

It is generally expected that the invoice description should depict the line-item/service provided by the user. However, in our data, we observed some invoice descriptions that are ambiguous as to the identification of the line items, even for the human reader (see an example 'Invoice ID 2' in Table 4.1). Processing only keywords from description fields for these invoices would lead to unwanted results.

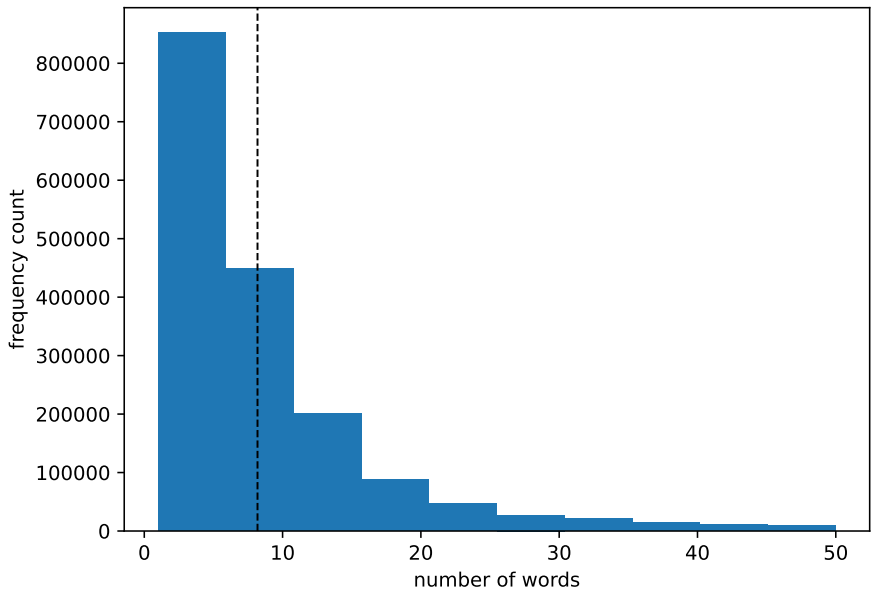


FIGURE 4.8: Frequency histogram of the number of words per each invoice description field in the invoice dataset. The dotted vertical line represents the average number of words per invoice description.

In general, SMEs use a business name that reflects the provided products or services they deliver to increase their customers' valuation (Grewal et al.; 1998). To overcome the issue of vague invoices, in this step, we combined the seller's (invoice creator) business name with the invoice description field. In other words, instead of using only the invoice description, we processed relevant keywords from a string that combines the seller's business names and invoice descriptions for each invoice.

Handling Spelling mistake

As mentioned above, invoice description data are noisy and contain spelling mistakes. We found that approximately 12% of process keywords that we obtained from the combined strings of invoice descriptions and seller's business names are misspelled. To replace misspelled keywords with their corresponding correct forms, we created a misspelling handling dictionary by using Levenshtein Distance Metric (Yujian and Bo; 2007). The steps of this dictionary creation procedure are described in Appendix A.

Processing Invoice Keywords Lists

In this step, we used NLTK (Bird; 2006) and Spacy (Honnibal and Montani; 2017) standard natural language cleaning methods to remove numbers, signs, special characters, stop words, and names (country, states, provinces, city, common names). We obtained a list of relevant keywords corresponding to each invoice. Table 4.2 presents an example of a processed keywords list corresponding to the invoices from Table 4.1.

Invoice ID	**Invoice Description**	**Seller's Business Name**	**Invoice Keywords**
1	SEO Monthly Foundation Building Plan	SciFi Webs	SEO, foundation, building, plan, web
2	RT +3.75 -2.00 090 ADD 2.50 SEG HT 20 PD 33.5/32.0	Adam Clear Vision Optical	clear, vision, optical
3	Flower arrangement for the big oval table in the lobby	Eve Rose Flowers Trading Ltd	flower, arrangement, oval, table, lobby, rose, flowers
4	$30/per man hour	John Housekeeping Services	man, housekeeping, services
5	Preparation of General Warranty Deed	John Doe Law	preparation, general, warranty, deed, law

TABLE 4.2: Processed keyword lists corresponding to the five example invoices given in Table 4.1

Generating Unique Keyword List

We generated a unique keyword list of size 8,280 and their corresponding frequencies from all of the processed invoice-wise keyword lists. The average frequency of this unique keyword list is 668.58 (i.e., on average, one keyword is present in approximately 668 invoices). In addition, we noticed in the keyword frequency distribution that approximately 10% of keywords had frequencies equal to 1. Keywords with unit frequencies would be translated as isolated nodes in the following network formulation step and filtering isolated nodes is a conventional practice before moving to the network clustering step (Furao and Hasegawa; 2006). Therefore, we removed all keywords with a frequency equal to 1. On the other hand, we also noticed that approximately 2% of keywords had frequencies greater than 10000. Possibly these keywords were very generic to describe a product or service. For example, the frequency of the keyword 'service' was 205,508. Thus, we removed those high-frequency keywords from the data. Finally, we obtained a list of unique keywords of size 7,286. Fig. 4.9 presents the top twenty keywords with corresponding frequencies from this final list.

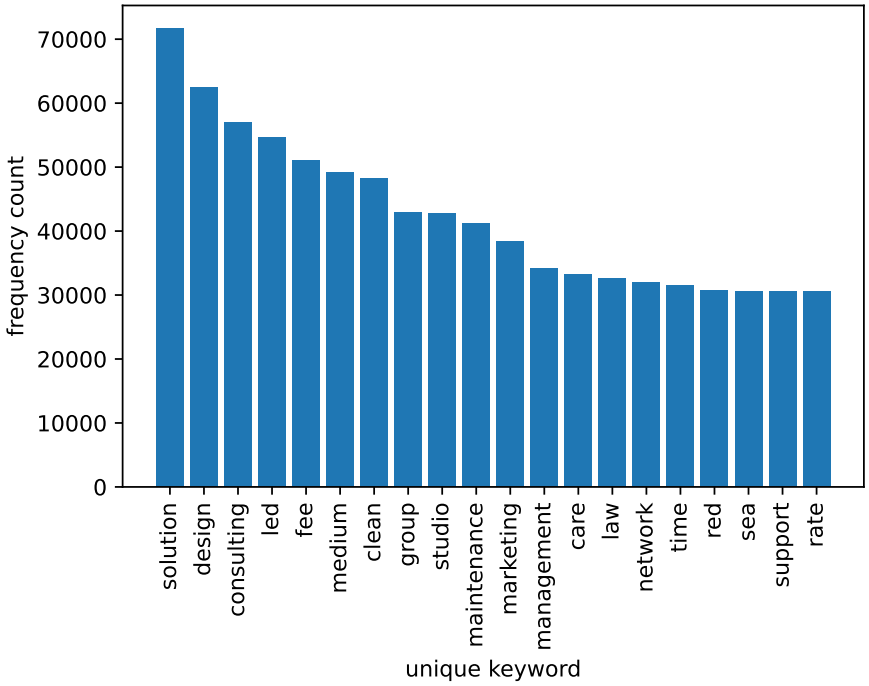


FIGURE 4.9: Top twenty keywords according to the corresponding frequency in processed invoice keyword lists.

4.7.2 Keyword Network Formulation from Invoice Keyword Lists

In the final step of data processing, we constructed a keyword network, as defined in Section 4.4, using the invoice keyword lists processed in the previous step. Therefore, in this network, each node is represented by a unique keyword, and each link is represented by two connecting keywords that are present in the same invoice. The initially formulated keyword network contained 7,286 nodes and 369,348 links. We noticed in the link weights distribution that approximately 30% of links had weights equal to 1, and approximately 0.05% of links had weights more than 50,000. In the next step, we 'cleaned' the initial keywords using the following three steps:

(i) Removing links with weights of more than 50000. Removing these 0.05% of the strong outlier links is significant for clustering strategies (Bellingeri et al.; 2020).

(ii) Removing links with unit weights. Removing these 30% weak links (small-weighted) does not significantly affect the network's connected clusters (Bellingeri et al.; 2020).

(iii) Finally, removing isolating nodes resulted from the previous two steps since network-based clustering does not allow for isolated nodes (Kaiser; 2008).

Table 4.3 presents the comparison overview of initial and final keyword networks and the corresponding approximate information losses after the above-mentioned cleaning process. Importantly, this table shows that though we removed approximately 30% from the initial links set, it only led to a loss of 9% from the initial nodes set.

	Initial Keyword Network	**Final Keyword Network**	**Information Loss (approx.)**
#of nodes	7286	6529	9%
#of links	369348	258,549	31%

TABLE 4.3: Comparison overview between initial and final keyword networks.

4.7.3 Implementation of HACoEWN

We implemented the HACoEWN algorithm (Algorithm 2) by using Python with the NetworkX (Hagberg et al.; 2008) graph package. We also used IBM Cplex 12.10 to solve integer programs in each execution of the subroutine ExactCoE.

4.7.4 Clusters Line-Item Category Identification, Labelling and Benchmarking

We identified 28 significant clusters (of size ≥ 45) from the invoice keyword network by using the HACoEWN algorithm. Among those 28, we identified line-item categories for 21 clusters with the aid of domain experts. In this process, we also used the top 20 keywords based on their degree in the keyword network corresponding to each identified cluster (see Appendix B). Table 4.4 presents the list of 21 identified line-item category clusters with their corresponding cluster sizes. From this table, we can say that users are creating invoices to provide services/products in a wide range of categories, from 'pet care and grooming' to 'web designing'. Also, identified line-item categories indicate that SME-type businesses are providing most of these services/products. We labelled these clusters with corresponding line-item categories and then saved them as a learned model.

4.7.5 Benchmarking

In order to evaluate the HACoEWN algorithm, we also applied the following three heuristic clustering algorithms designed for large-scale networks: Louvain algorithm (Blondel

Cluster ID	Size	Identified line-item category
1	879	Pet care and grooming
2	637	Transportation, holiday and tour services
3	516	Massage therapy
4	436	Farming and agriculture services
5	349	Constructions, renovations, and maintenance
6	321	Hardware and equipment
7	262	Investment and portfolio management
8	177	Taxation and legal or law related services
9	172	Landscape designing
10	160	Home furniture and appliances
11	130	Spa, beauty and natural healing
12	77	IT, software, web designing and maintaining
13	75	Flowers, photography, wedding and party planning
14	68	Automobile services
15	68	Hunting and outdoor activities
16	66	Housekeeping and cleaning services
17	54	Fitness, training and outdoor activities
18	51	Home electronics repair and installation
19	49	Roofing services
20	48	Spiritual services
21	45	Pest control

TABLE 4.4: Line-item category and number of keywords (nodes) corresponding to each identified cluster in KN.

et al.; 2008), Clauset-Newman-Moore algorithm (Clauset et al.; 2004), and HACEWN (Wahid et al.; 2022). Table 4.5 presents the results of this benchmarking. The Louvain algorithm produced the maximum number of significant clusters (size $\geq$45). However, the maximum number of identified line-item clusters was produced by the HACoEWN algorithm. In addition, HACoEWN's line-item-identified clusters covered the maximum percentage of nodes (71%) from the invoice KN. On the other hand, line-item-identified clusters produced by HACEWN, which is a P_3-free algorithm, covered the least percentage of nodes (7%) from the invoice KN. Based on these results, we decided to use the HACoEWN algorithm in the line-item identification process in our proposed framework.

4.8 Limitation, Future Work and Concluding Remark

This study proposed a framework for identifying line-item categories (classifying labels) and classifying user-generated short-text. We used the Cograph Editing problem-based clustering on weighted networks in this process and designed an ILP formulation. Furthermore, we developed a heuristic algorithm called HACoEWN for Cograph Editing

Clustering algorithm	**# of cluster (significant size)**	**# of line-item identified clusters**	**Nodes % (Appx. in identified clusters)**
Louvain algorithm (modularity)	31	12	39%
Clauset-Newman-Moore (modularity)	25	5	18%
HACEWN (P3 free)	12	6	7%
HACoEWN (P4 free/cograph)	28	21	71%

TABLE 4.5: Number of significant clusters (size greater than or equal to 45), the number of identified invoice line-item categories clusters corresponding to four studied clustering algorithms.

on weighted networks to deal with large-scale networks. This framework can be implemented in practice due to computation simplicity in the production phase.

The first limitation of our study is that we only considered the mutual coincident relation between keywords in the network formulation. In some contexts, the position of a keyword in the short-text may have an important role in determining line-item categories. Based on the keywords' position in the short-text, in future, we can extend our framework by introducing a keyword-importance metric, as given in Wang et al. (2016), or a weighting scheme, as seen in Alsmadi and Hoon (2019b), corresponding to each identified cluster.

Kaur and Kumar (2018) showed that clustering-based algorithms could leverage knowledge from documents' taxonomic ontology. In this study, we did not use any platform-specific ontological information in the classification process. In future, besides the keywords clusters, we can extend our framework by adding another level of filtering (or features selection) by using the given short-text's ontology as seen in Škrlj et al. (2021) and Munoz et al. (2022).

Motivated by a practical application of understanding invoices, we applied the proposed framework to identify line-item categories and classify invoices generated by users' in a real-life CB-AIS platform. A future extension of this application could be to use invoice categorization to identify corresponding users' (in this case, SMEs) categories based on business types and design platform features, accordingly. It may also help CB-AIS companies in setting business strategies (e.g., target marketing and promotion) for specific types of users.

Funding

We acknowledge support for the Natural Sciences and Engineering Research Council (NSERC) Discovery (Award Number: RGPIN-2020-06792) and Mitacs Accelerate Fellowship Program (Award Number: IT28317) programs for their support of this project.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Due to the confidentiality of the financial data used in this research, participants of this study did not agree to share their data publicly.

Appendix A

Misspelling and Root Keyword Handling Dictionary

We used the following steps to develop a misspelling-handling dictionary based on our collected invoice description data:

- **Collecting all unique keywords:** First, we collected all unique keywords from the processed invoice keywords fields.
- **Finding the distance between two unique keywords:** Next, to find two close keywords, we calculated the Levenshtein Distance (Levenshtein et al.; 1966) between each of two unique keywords (i.e., find the distance between two string sequences). This metric calculates the minimum number of single-character edits (insertion or deletions) to convert one string to another.
- **Identify misspelled and root keywords:** We listed all pairs of keywords with mutual Levenshtein Distances that are less than 5% (i.e., at least 95% similar pairs of keywords). Then we manually investigated each of the listed pairs and identified misspelled and root keywords. For example, mutual Levenshtein Distances are less than 5% for the following keywords: 'alumni', 'aluminum', 'aluminium', 'aluminiumit', 'alumino', 'aluminumo', 'aluminun'. Only 'aluminum' (North American

English) and 'aluminium' (UK English) are correct spellings; the rest of these misspelled keywords are in our data. To keep it simple, we only used the 'aluminum' as the correct spelling and replaced all others.

Similarly, we identified similar rooted keywords and replaced them with the root keyword. For example, 'clean', 'cleanly', 'cleanse', 'cleansing', 'cleanup', 'cleanable', and 'cleaning' all come from the root keyword 'clean'. Therefore, we replaced all correlated keywords with 'clean'.

- **Creating a misspelling and root keyword handling dictionary:** In the next step, we created a dictionary by listing all corresponding misspelled and root keywords found in our data with their correct spelling.
- **Clean invoice data:** Finally, we used this dictionary to replace all misspelled keywords in processed invoice keywords. We also saved this dictionary to use later to categorize new invoices generated in our partner's CB-AIS in real-time.

Note that we developed this misspelling-handling dictionary based on our collected data. That is, it only identified misspellings contained in our initial invoice data and can not identify any further variation of misspelling or any new misspelled keyword. Therefore, to keep it up-date, we need to add newly appearing misspelled keywords to this dictionary by investigating *non-associative clusters* as explained in Section 4.6.

Appendix B

Identifying invoice line-item categories from top twenty keywords corresponding to each significant cluster

Cluster ID	Cluster size	Top 20 keywords (based on degree in Invoice Keywords Network)	Identified Line-Item Category
1	879	chorea, bleached, ambulatory, remit, rapture, colossal, bandy, kingfisher, errand, injure, cystotomy, valise, canal, sharply, swimmer, button, chalk, allusion, strapped, fortissimo	Pet care and grooming
2	637	ragwort, surveyor, rapture, dotard, fantastic, vigil, valued, intricate, individual, booked, summer, nonbeverage, solar, pit, hauler, canal, coronet, eloquent, unfordable, comet	Holiday and tour packages
3	516	methadone, pervade, flexure, elapse, depressing, pill, criticality, vicar, mutual, marionette, tambourine, pediatric, partner, caress, triceps, mechanic, palliation, trainee, scaly, nape	Massage therapy
4	436	hydraulic, lofty, flourish, snowbird, skinned, flour, roadhouse, grommet, playhouse, steer, ingredient, crackpot, flood, palisade, yard, dilapidated, grayish, amalgamate, rancher, beget	Farm and agriculture services
5	349	crafty, bevelled, lift, vividly, flatiron, tinsel, undeveloped, incise, sheepskin, corral, tenant, renovation, recoupment, buttoned, carrotwood, thunder, buckaroo, survivorship, brute, roller	Constructions and renovations
6	321	raft, esophagitis, stroking, radium, elevation, tar, perpetrator, sandpaper, cooking, perpetuate, melting, scaly, hotfoot, penitentiary, valve, weld, smash, forced, knotted, glazed	Hardware and equipment
7	262	individual, warren, retire, hero, lottery, vara, engagement, fund, flail, intricate, money, stock, buzz, recession, cash, growth, share, portfolio, dependent, quality	Investment and portfolio managements
8	177	shipboard, record, drunk, agreement, group, prior, appeal, justice, case, transaction, permanent, generative, handling, council, evaluate, represent, trail, calorie, gash, immigrant	Legal and law services

TABLE 4.6: List of all line-item category identified clusters with corresponding cluster size and top twenty keywords (according to degree in invoice keywords network) by using HACoEWN.

Continuation of Table 4.6

Cluster ID	**Cluster size**	**Top 20 keywords (based on degree in Invoice Keywords Network)**	**Identified Line-Item Category**
9	172	osteoid, moor, deadpan, laurel, everglade, harborage, slough, vitreous, gladiolus, grate, hedge, archipelago, nitric, champ, sewn, insecticide, toiling, townsman, scour, smokehouse, sisyrinchium	Landscape designing
10	160	exhibit, plowing, chocolate, hat, gloss, hen, nursery, sofa, collar, charcoal, nation, doorbell, acrylic, hasten, heeled, evacuation, bergamot, contractor, flooring, bob-tailed	Farmhouse shops and décor
11	130	surging, fierce, naturally, radium, sunglasses, flashing, lynch, visceral, spa, hydroplane, uplifting, sod-buster, merciful, tedium, grief, sundry, purify, goodwill, distort, spotlight	Spa and natural healing
12	77	suspicious, warning, recast, unauthorized, visibility, antivirus, sluggish, hack, password, availability, available, degree, visitor, vividly, SEO, rebuild, speed, console, loading	Web designing and maintaining
13	75	lifting, closer, posted, groomsman, meeting, intracoastal, renounce, posture, tickled, ladder, sharper, resting, prominence, skittle, gift, headline, limp, bottle, reclamation, curtain	Wedding and party planning
14	68	radiation, quiver, oppressor, colored, spoke, intricate, pervade, transmit, flail, slater, coronet, foaming, flexure, perpetrator, button, topmost, vibrating, tray, tachometer, flashing	Automobile services
15	68	circuit, acquire, malleable, symposium, buttonhole, catapult, playmate, menhaden, growl, dehydrate, drawbridge, monogrammed, combed, tongs, hunting, consonant, masquerade, splendor, supple, collaborate	Hunting and outdoor activities
16	66	pretty, bore, lofty, clip, trunk, crown, extract, pillow, cabin, carry, environs, shelter, exam, chateau, gum, dry, clean, noel, magnet, hygienic	Housekeeping and cleaning services
17	54	caldron, turnaround, translate, red, share, pickpocket, heading, rub, groaning, demonstrable, overview, illustrate, skylight, seamanship, deserter, venom, packer, adventurous, chrysotile, barefoot	Outdoor activities

TABLE 4.6: List of all line-item category identified clusters with corresponding cluster size and top twenty keywords (according to degree in invoice keywords network) by using HACoEWN.

Continuation of Table 4.6

Cluster ID	**Cluster size**	**Top 20 keywords (based on degree in Invoice Keywords Network)**	**Identified Line-Item Category**
18	51	button, retainer, tower, lamp, pot, circuit, semi,feed, pin, socket, point, bracket, electronic, pole, arm, speaker, switch, box, jack, junction	Home electronics installation
19	49	gear, miner, panel, sultan, screen, leakage, downpour, catch, insect, attach, shaft, ratio, heavy, velocity, torn, planter, aluminum, bottle, rubber, roof	Roofing services
20	48	seldomspill, oatmeal, demarcation, nightcap, gall, boost, powdery, droop, forming, booth, psychotherapy, footing, copper, clarion, witness, monk, pua, botanical, cogwheel, treetop, shogunate, upon, adequate, ruin	Roofing services
21	45	chalk, cornice, convoy, tailpipe, bedbug, sensation, hawk, healer, evasive, blindfold, foam, bootlicker, gruff, temperament, damage, radiated, pump, maisonette, lodgepole, electrocute	Pest control

End of Table 4.6

Chapter References

Alshirah, M., Lutfi, A., Alshirah, A., Saad, M., Ibrahim, N. and Mohammed, F. (2021). Influences of the environmental factors on the intention to adopt cloud based accounting information system among smes in Jordan, *Accounting* **7**(3): 645–654.

Alsmadi, I. and Hoon, G. K. (2019a). Review of short-text classification, *International Journal of Web Information Systems* **15**(2): 155–182.

Alsmadi, I. and Hoon, G. K. (2019b). Term weighting scheme for short-text classification: Twitter corpuses, *Neural Computing and Applications* **31**(8): 3819–3831.

Asatiani, A., Apte, U., Penttinen, E., Rönkkö, M. and Saarinen, T. (2019). Impact of accounting process characteristics on accounting outsourcing-comparison of users and non-users of cloud-based accounting information systems, *International Journal of Accounting Information Systems* **34**: 100419.

Asatiani, A. and Penttinen, E. (2015). Managing the move to the cloud–analyzing the risks and opportunities of cloud-based accounting information systems, *Journal of Information Technology Teaching Cases* **5**(1): 27–34.

Baldwin, T., De Marneffe, M.-C., Han, B., Kim, Y.-B., Ritter, A. and Xu, W. (2015). Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition, *Proceedings of the Workshop on Noisy User-generated Text*, pp. 126–135.

Bansal, N., Blum, A. and Chawla, S. (2004). Correlation clustering, *Machine Learning* **56**(1): 89–113.

Bardelli, C., Rondinelli, A., Vecchio, R. and Figini, S. (2020). Automatic electronic invoice classification using machine learning models, *Machine Learning and Knowledge Extraction* **2**(4): 617–629.

Beliga, S., Meštrović, A. and Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches, *Journal of Information and Organizational Sciences* **39**(1): 1–20.

Bellingeri, M., Bevacqua, D., Scotognella, F., Alfieri, R. and Cassi, D. (2020). A comparative analysis of link removal strategies in real complex weighted networks, *Scientific Reports* **10**(1): 1–15.

Beļskis, Z., Zirne, M. and Pinnis, M. (2020). Features and methods for automatic posting account classification, *International Baltic Conference on Databases and Information Systems*, Springer, pp. 68–81.

Bengtsson, H. and Jansson, J. (2015). *Using classification algorithms for smart suggestions in accounting systems*, Master's thesis.

Bergdorf, J. (2018). Machine learning and rule induction in invoice processing: Comparing machine learning methods in their ability to assign account codes in the bookkeeping process.

Bird, S. (2006). Nltk: the natural language toolkit, *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pp. 69–72.

Biswas, K., Muthukkumarasamy, V. and Sithirasenan, E. (2013). Maximal clique based clustering scheme for wireless sensor networks, *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, IEEE, pp. 237–241.

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent dirichlet allocation, *Journal of Machine Learning Research* **3**(Jan): 993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E. (2008). Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10): P10008.

Böcker, S. and Baumbach, J. (2013). Cluster editing, *Conference on Computability in Europe*, pp. 33–44.

Böcker, S., Briesemeister, S. and Klau, G. W. (2011). Exact algorithms for cluster editing: Evaluation and experiments, *Algorithmica* **60**(2): 316–334.

Brandes, U. (2005). *Network analysis: Methodological foundations*, Vol. 3418, Springer Science & Business Media.

Brandstädt, A., Le, V. B. and Spinrad, J. P. (1999). *Graph classes: A survey*, SIAM.

Cai, L. (1996). Fixed-parameter tractability of graph modification problems for hereditary properties, *Information Processing Letters* **58**(4): 171–176.

Cedillo, P., García, A., Cárdenas, J. D. and Bermeo, A. (2018). A systematic literature review of electronic invoicing, platforms and notification systems, *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*, IEEE, pp. 150–157.

Cevahir, A. and Murakami, K. (2016). Large-scale multi-class and hierarchical product categorization for an e-commerce giant, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 525–535.

Charikar, M., Guruswami, V. and Wirth, A. (2005). Clustering with qualitative information, *Journal of Computer and System Sciences* **71**(3): 360–383.

Chen, M., Jin, X. and Shen, D. (2011). Short text classification improved by learning multi-granularity topics, *Twenty-second International Joint Conference on Artificial Intelligence*.

Choi, J. and Hwang, Y.-S. (2014). Patent keyword network analysis for improving technology development efficiency, *Technological Forecasting and Social Change* **83**: 170–182.

Christauskas, C. and Miseviciene, R. (2012). Cloud–computing based accounting for small to medium sized business, *Engineering Economics* **23**(1): 14–21.

Chua, C. E. H., Storey, V. C., Li, X. and Kaul, M. (2019). Developing insights from social media using semantic lexical chains to mine short text structures, *Decision Support Systems* **127**: 113142.

Clauset, A., Newman, M. E. and Moore, C. (2004). Finding community structure in very large networks, *Physical Review E* **70**(6): 066111.

Cleary, P. and Quinn, M. (2016). Intellectual capital and business performance: An exploratory study of the impact of cloud-based accounting and finance infrastructure, *Journal of Intellectual Capital* **17**(2): 255–278.

Corneil, D. G., Perl, Y. and Stewart, L. K. (1985). A linear recognition algorithm for cographs, *SIAM Journal on Computing* **14**(4): 926–934.

Crespelle, C. (2021). Linear-time minimal cograph editing, *International Symposium on Fundamentals of Computation Theory*, Springer, pp. 176–189.

Davis, J. A. and Leinhardt, S. (1967). The structure of positive interpersonal relations in small groups.

Dimitriu, O., Matei, M. et al. (2014). The expansion of accounting to the cloud, *SEA-Practical Application of Science* **4**(2): 237–240.

Dondi, R., Lafond, M. and El-Mabrouk, N. (2017). Approximating the correction of weighted and unweighted orthology and paralogy relations, *Algorithms for Molecular Biology* **12**(1): 1–15.

El-Mallah, E. S. and Colbourn, C. J. (1988). The complexity of some edge deletion problems, *IEEE Transactions on Circuits and Systems* **35**(3): 354–362.

Eldalabeeh, A. R., Al-Shabil, M. O., Almuiet, M. Z., Bany Baker, M. and E'lemiat, D. (2021). Cloud-based accounting adoption in jordanian financial sector, *The Journal of Asian Finance, Economics and Business* **8**(2): 833–849.

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

Furao, S. and Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning, *Neural networks* **19**(1): 90–106.

Gao, Y., Hare, D. R. and Nastos, J. (2013). The cluster deletion problem for cographs, *Discrete Mathematics* **313**(23): 2763–2771.

González, P. C. and Velásquez, J. D. (2013). Characterization and detection of taxpayers with false invoices using data mining techniques, *Expert Systems with Applications* **40**(5): 1427–1436.

Greco, F. and Polli, A. (2020). Emotional text mining: Customer profiling in brand management, *International Journal of Information Management* **51**: 101934.

Grewal, D., Krishnan, R., Baker, J. and Borin, N. (1998). The effect of store name, brand name and price discounts on consumers' evaluations and purchase intentions, *Journal of Retailing* **74**(3): 331–352.

Hadar, Y. and Shmueli, E. (2021). Categorizing items with short and noisy descriptions using ensembled transferred embeddings, *arXiv preprint* .

Hagberg, A., Swart, P. and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx, *Technical report*, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Hamza, H., Belaïd, Y. and Belaïd, A. (2007). Case-based reasoning for invoice analysis and recognition, *International conference on case-based reasoning*, Springer, pp. 404–418.

Hedberg, N. (2020). Automated invoice processing with machine learning: Benefits, risks and technical feasibility.

Hellmuth, M. and Wieseke, N. (2018). On tree representations of relations and graphs: Symbolic ultrametrics and cograph edge decompositions, *Journal of Combinatorial Optimization* **36**(2): 591–616.

Hellmuth, M., Wieseke, N., Lechner, M., Lenhof, H.-P., Middendorf, M. and Stadler, P. F. (2015). Phylogenomics with paralogs, *Proceedings of the National Academy of Sciences* **112**(7): 2058–2063.

Hempstalk, K. (2017). BTD10: Machine Learning at Xero. [Online]. [Accessed: Feb-28-2023].
URL: *https://rb.gy/ho7ln4*

Homans, G. C., Hare, A. P. and Polley, R. B. (2017). *The human group*, Routledge.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Inches, G., Carman, M. J. and Crestani, F. (2010). Statistics of online user-generated short documents, *European Conference on Information Retrieval*, Springer, pp. 649–652.

Kaiser, M. (2008). Mean clustering coefficients: the role of isolated nodes and leafs on clustering measures for small-world networks, *New Journal of Physics* **10**(8): 083042.

Kariyawasam, A. (2019). Analysing the impact of cloud-based accounting on business performance of smes, *The Business & Management Review* **10**(4): 37–44.

Kaur, R. and Kumar, M. (2018). Domain ontology graph approach using markov clustering algorithm for text classification, *International Conference on Intelligent Computing and Applications*, Springer, pp. 515–531.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D. (2019). Text classification algorithms: A survey, *Information* **10**(4): 150.

Křivánek, M. and Morávek, J. (1986). Np-hard problems in hierarchical-tree clustering, *Acta Informatica* **23**(3): 311–323.

Kühnl, F. (2014). orthodeprime: A tool for heuristic cograph editing on estimated orthology graphs. bachelor's thesis.

Lesner, C., Ran, A., Rukonic, M. and Wang, W. (2019). Large scale personalized categorization of financial transactions, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 9365–9372.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, Vol. 10, Soviet Union, pp. 707–710.

Liu, J., Pei, L., Sun, Y., Simpson, H., Lu, J. and Ho, N. (2021). Categorization of financial transactions in quickbooks, *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3299–3307.

Liu, Y., Jiang, C. and Zhao, H. (2019). Assessing product competitive advantages from the perspective of customers by mining user-generated content on social media, *Decision Support Systems* **123**: 113079.

Liu, Y., Wang, J., Guo, J. and Chen, J. (2012). Complexity and parameterized algorithms for cograph editing, *Theoretical Computer Science* **461**: 45–54.

Ma, D., Fisher, R. and Nesbit, T. (2021). Cloud-based client accounting and small and medium accounting practices: Adoption and impact, *International Journal of Accounting Information Systems* **41**: 100513.

McAssey, M. P. and Bijma, F. (2015). A clustering coefficient for complete weighted networks, *Network Science* **3**(2): 183–195.

Mishra, N., Schreiber, R., Stanton, I. and Tarjan, R. E. (2007). Clustering social networks, *International Workshop on Algorithms and Models for the Web-Graph*, Springer, pp. 56–67.

Munoz, J., Jalili, M. and Tafakori, L. (2022). Hierarchical classification for account code suggestion, *Knowledge-Based Systems* p. 109302.

Nastos, J. and Gao, Y. (2013). Familial groups in social networks, *Social Networks* **35**(3): 439–450.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks, *Physical Review E* **69**(2): 026113.

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M. and Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: a systematic review of current approaches, *Systematic Reviews* **4**(1): 1–22.

Schaeffer, S. E. (2007). Graph clustering, *Computer Science Review* **1**(1): 27–64.

Seinsche, D. (1974). On a property of the class of n-colorable graphs, *Journal of Combinatorial Theory, Series B* **16**(2): 191–193.

Serrano, M. Á., Boguñá, M. and Pastor-Satorras, R. (2006). Correlations in weighted networks, *Physical Review E* **74**(5): 055101.

Shamir, R., Sharan, R. and Tsur, D. (2004). Cluster graph modification problems, *Discrete Applied Mathematics* **144**(1-2): 173–182.

Škrlj, B., Martinc, M., Kralj, J., Lavrač, N. and Pollak, S. (2021). tax2vec: Constructing interpretable features from taxonomies for short text classification, *Computer Speech & Language* **65**: 101104.

Song, Y., Wang, H., Wang, Z., Li, H. and Chen, W. (2011). Short text conceptualization using a probabilistic knowledgebase, *Twenty-second International Joint Conference on Artificial Intelligence*.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H. and Demirbas, M. (2010). Short text classification in twitter to improve information filtering, *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 841–842.

Syed, S. and Spruit, M. (2017). Full-text or abstract? Examining topic coherence scores using Latent Dirichlet Allocation, *2017 IEEE International conference on data science and advanced analytics (DSAA)*, IEEE, pp. 165–174.

Thomas, J., McNaught, J. and Ananiadou, S. (2011). Applications of text mining within systematic reviews, *Research Synthesis Methods* **2**(1): 1–14.

Trivedi, N., Asamoah, D. A. and Doran, D. (2018). Keep the conversations going: engagement-based customer segmentation on online social service platforms, *Information Systems Frontiers* **20**(2): 239–257.

Wahid, D. F., Ezzeldin, M., Hassini, E. and El-Dakhakhni, W. W. (2022). Common-knowledge networks for university strategic research planning, *Decision Analytics Journal* **2**: 100027.

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C.-L. and Hao, H. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification, *Neurocomputing* **174**: 806–814.

Wang, W., Lesner, C., Ran, A., Rukonic, M., Xue, J. and Shiu, E. (2020). Using small business banking data for explainable credit risk scoring, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 13396–13401.

Yoo, S., Jang, S., Byun, S. W. and Park, S. (2019). Exploring human resource development research themes: A keyword network analysis, *Human Resource Development Quarterly* **30**(2): 155–174.

Yujian, L. and Bo, L. (2007). A normalized levenshtein distance metric, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6): 1091–1095.

Zhang, H. and Zhong, G. (2016). Improving short text classification by learning vector representations of both words and hidden topics, *Knowledge-Based Systems* **102**: 76–86.

Zhu, D., Lappas, T. and Zhang, J. (2018). Unsupervised tip-mining from customer reviews, *Decision Support Systems* **107**: 116–124.

Chapter 5

Hybrid Fraud Detection Framework in Invoicing Platforms using an Augmented AI Approach

The content of this chapter is a revision of the manuscript text submitted for publication under the following title:

Wahid, D. F., Hassini, E.. Hybrid Fraud Detection Framework in Invoicing Platforms using an Augmented AI Approach.

Hybrid Fraud Detection Framework in Invoicing Platforms using an Augmented AI Approach

Dewan F. Wahid
School of Computational Science and Engineering
McMaster University, Hamilton, ON, Canada
Email: wahidd@mcmaste.ca

Elkafi Hassini
DeGroote School of Business
McMaster University, Hamilton, ON, Canada
Email: hassini@mcmaster.ca

Abstract

In this era of e-commerce, many companies are moving towards subscription-based invoicing platforms to maintain their electronic invoices. Unfortunately, fraudsters are using these platforms for different types of malicious activities. Identifying fraudsters is often challenging for many companies due to the limitation of time and other resources. On the other hand, a fully automated fraud detection model also creates a risk of false-positive identification. This paper proposed a Hybrid Fraud Detection Framework when only a small set of labelled (fraud/non-fraud) data is available, and human input is required in the final decision-making step. This framework used a combination of unsupervised and supervised machine learning, red-flag prioritization, and an augmented AI approach containing a human-in-the-loop process. We also proposed a weighted center based on the feature importance scores for the fraud risk cluster and used it in the red-flag prioritization process. Finally, we outlined a case study of this hybrid framework in a weekly segmented structure to identify fraudulent users in an invoicing platform. Our hybrid framework showed promising results in identifying fraudulent users and improving human performance when human input is required to make the final decision.

Keywords: hybrid fraud detection; unsupervised clustering; artificial neural network; red-flag prioritization; human-in-the-loop; augmented AI.

5.1 Introduction

The rapid adaptation of online business platforms in recent years has led to the use of electronic invoices in every industrial sector (Cedillo et al.; 2018). Companies are generally using different electronic accounting and invoicing services (EAIS) systems to maintain their accounting and bookkeeping. Besides maintaining accounting, many small and medium-sized enterprises (SMEs) are shifting towards EAIS systems to generate their business invoices, send these invoices electronically to their clients, and receive payments directly to their banks (Asatiani et al.; 2019). These platforms are efficient and reliable, and they provide real-time access to business data (Christauskas and Miseviciene; 2012). Several studies indicate that using EAIS infrastructures positively impacts SMEs' efficiency, human, and relational capital (Cleary and Quinn; 2016; Kariyawasam; 2019). Furthermore, SMEs can avoid several red-tapes and mitigate cash flow issues using electronic invoices for their services or products (Lee; 2016; Guerar et al.; 2020). However, maintaining an 'in-house' IT department to run an EAIS is often very expensive for many SMEs (Asatiani and Penttinen; 2015). Therefore, following the increasing demand, many Cloud-based Accounting and Invoicing Service (CB-AIS) companies are coming to markets targeting SMEs to undertake their electronics business accounts and invoices in the cloud instead of hiring external accountants or IT personnel (Eldalabeeh et al.; 2021; Ma et al.; 2021). CB-AIS companies are operating as subscription-based software-as-a-service (SaaS) and providing cost-saving accounting and invoicing platforms services to SMEs (Asatiani and Penttinen; 2015).

With the growing number of invoicing platforms, invoice-related fraudulent activities using these platforms are also increasing (Guerar et al.; 2020). Opening a trial account for a certain period in any of the popular CB-AIS's invoicing platforms is easy; you often only need an email address (Popivniak; 2019; Trialopedia; 2021). Fraudsters use trial accounts to carry out invoice-related malicious activities. Businesses and customers are falling for these fraudulent activities because fraudsters can generate professional-looking invoices during the trial period (Xie et al.; 2019; Guerar et al.; 2020). Studies show that SMEs are more vulnerable to fraud and suffer disproportionate losses than large enterprises (Kramer; 2015). According to another study, businesses in the UK lost an estimated 9 billion pounds to invoice-related frauds (White; 2017). Even the tech-sophisticated company Amazon has lost $19 million due to a fraudulent invoicing scheme targeting their vendor system (U.S. Attorneyś Office; 2020).

In general, all payments in the invoicing platform go through payment gateway services (PGS) providers (Chan et al.; 2022). As the invoicing platform service provider, CB-AIS companies usually pay a fixed fee per user to PGS providers for carrying risks and securing transactions (Guerar et al.; 2020). Moreover, based on our experience, the fixed fee per user is generally higher if an invoicing platform is prone to fraud. Therefore, a fraudulent user is spamming other customers, businesses, or individuals and, as the service provider, also causing financial harm to the CB-AIS company. Besides creating financial burdens, fraudulent activities in an invoicing platform draw bad reviews and loss of business integrity (Guerar et al.; 2020). Moreover, identifying fraudulent users in invoice platforms also helps to mitigate more significant financial crimes such as money laundering and tax evasion (Cassara; 2015; Dejong; 2018). A recent report from McKinsey & Company emphasized that advanced machine learning-based fraudulent client risk detection algorithms would be the essential tool to fight against money laundering (Kumar et al.; 2022).

Fraud is a well-studied topic in financial sectors, especially for credit card and insurance-related frauds (Abdallah et al.; 2016; Bhattacharyya et al.; 2011; Al-Hashedi and Magalingam; 2021). In recent years, several studies have appeared in the literature to deal with invoice-related fraud (Guerar et al.; 2020). Most of these studies followed supervised machine learning approaches with accurately labelled training data (Pai et al.; 2011), or smart-contract-based blockchain technology approaches to prevent double financing (Gong et al.; 2022; Xie et al.; 2019). Furthermore, labelling the training data is not always feasible, most often expensive and time-consuming, as it requires human annotators sometimes with specific domain expertise (Hady and Schwenker; 2013). Again, in many practical applications, especially for detecting financial fraud, sometimes it is required to have a human-in-the-loop (HiTL) to make the final decision (Baader and Krcmar; 2018; Balayan et al.; 2020; Maadi et al.; 2021; Karim et al.; 2022). Such approaches have been referred to as augmented Artificial Intelligence (AI) in the literature. Many studies showed that the augmented AI approach performs better than classical machine learning approaches (Gopinath et al.; 2016; Agnisarman et al.; 2019; Reddy et al.; 2021). To the best of our knowledge, only a handful of studies have used augmented AI in invoice fraud detection structures (e.g., Kim et al. (2022), Chan et al. (2022) and Hamelers (2021)).

In this paper, we proposed a hybrid fraud detection framework (HFDF) for invoicing platforms that uses a small set of labelled processes. A combination of unsupervised, supervised machine learning and cluster association of the small labelled data set was used to identify the fraud risk cluster (FRC) and the subsequent model training process.

We also proposed a weighted center for the FRC based on the feature importance score and utilized it in the red-flag (RF) prioritization and augmented AI processes. Furthermore, this paper presented a case study of the proposed HFDF for a CB-AIS company to identify fraudulent users in their invoicing platform. We implemented a weekly segmented structure of the proposed HFDFs to detect fraudulent users as early as possible after signing up on the platform.

The main motivation for designing a framework for identifying fraudulent users in invoicing platforms came from practical implementations requiring human intervention in the final decision-making process. Also, we wanted to develop this model when only a small labelled data set is available for training. The contributions of this paper are as follows:

i. An HFDF for invoicing platforms using small labelled data.

ii. A combination of a red-flag and an augmented AI approach-based design for empirical settings where human review is required in the final decision-making process.

iii. A weighted center based on the feature importance scores for the fraud risk cluster in the red-flag prioritization process.

iv. A case study of identifying fraudulent users in a CB-AIS's invoicing platform.

v. A weekly segmented structure of HFDF was designed to identify fraudulent users as soon as possible after signing up on the CB-AIS's invoicing platform.

The rest of this paper is organized as follows. Section 5.2 outlines the background and related works on this topic. Section 5.3 presents a high-level architecture for the HFDF for invoicing platforms as well as detailed discussions on different components of our proposed framework. Section 5.4 illustrates a case study of the proposed framework in an empirical setting. Finally, we give concluding remarks, limitations and future works in Section 5.5

5.2 Background and Related Work

5.2.1 Fraud in Invoicing Platforms

Any fraudulent activity related to invoices can be classified as invoice fraud. It is one of the latest methods fraudsters have been using in recent years due to the surge of cloud-based invoicing platforms (Popivniak; 2019; Trialopedia; 2021).

Several types of fraudulent activities can be observed in invoicing platforms, such as sending fake invoices posing as legitimate suppliers or pretending to be a team member (GrantThornton; 2021; Barclays; 2022). Fig. 5.1 presents a classification of fraudulent activities observed in our case study's invoicing platform. The fraudulent activities in typical invoicing platforms can be divided into the following two categories: short-term and long-term frauds. Short-term frauds include: sending fake invoices, self-invoicing with stolen credit cards, and posing as team members.

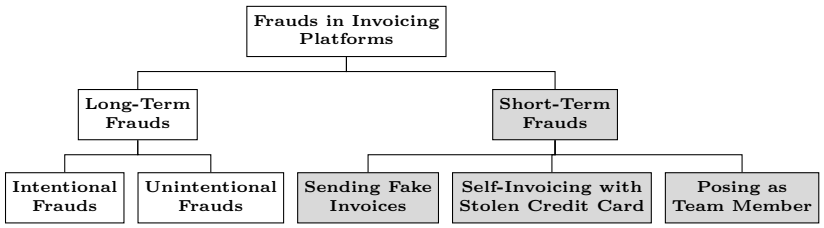


FIGURE 5.1: Typical fraudulent activities' classification in invoicing platforms.

Sending fake invoicing is the most common type of invoice-related fraud (Kramer; 2015; Xie et al.; 2019). Generally, a fraudster claiming to be a legitimate supplier sends invoices with small bill amounts (Kearse; 2020). Fake invoices in which payments are made for fictitious products or services may be undetected for up to 24 months before any in-depth audits (Stamler et al.; 2014). Using stolen or lost credit cards is one of the leading causes and accounts for 12% of all fraudulent activities (Pavía et al.; 2012). Self-invoicing with stolen credit cards is another type of short-term fraud that can be performed through the invoicing platform. In this case, fraudsters open two user accounts in the invoicing platform and then pay invoices from one fraud account to another by using stolen credit cards. Finally, posing as a team member and then asking to pay invoices is the most common fraud in invoicing platforms. In this case, fraudsters discover some details of business expenses types, amount, and regularity and ask to pay expense invoices utilizing that information (GrantThornton; 2021).

On the other hand, long-term fraud can be divided into two main categories: intentional and unintentional fraud cases. Intentional frauds are when a fraudster uses a business to do malicious activities with long-term planning. Unintentional fraud occurs when a legitimate business fails to deliver products after receiving an invoice payment due to unexpected financial losses or other uncontrollable reasons. Both intentional and unintentional frauds occur over a long period and are not in this paper's scope.

5.2.2 Hybrid Machine Learning Framework

Supervised machine learning (SML) is generally very efficient in different application sectors when labelled training data with high accuracy is available (Raghavan and El Gayar; 2019). However, if accurately labelled data is available, but the size is small compared to the unlabelled data set, the induced predictive model using SML shows a deficient performance (Forestier and Wemmert; 2016). In addition, labelling training data is costly and may be affected by human bias (Pise and Kulkarni; 2008; Chakraborty et al.; 2021). Regarding unsupervised machine learning (UML), several challenges related to pattern or cluster interpretations have been discussed in the literature (Li et al.; 2021). Most often, interpreting results from UML is challenging since it identifies patterns or clusters without using any label as a semantic reference (Wang and Biljecki; 2022). Some researchers interpreted clusters from UML by identifying the most representative features using manually summarizing corresponding clusters' features (Ferrara et al.; 2017) or taking help from subject matter experts (Kruber et al.; 2018). However, these approaches do not consider statistically well-defined, and human professionals are still required for final interpretation, which creates windows for human biases (Wang and Biljecki; 2022).

As we discussed, both UML and SML have advantages and disadvantages. Li et al. (2018) suggested combining UML and SML in a hybrid setting to overcome their shortcomings. Therefore, hybrid machine learning frameworks use a combination of UML and SML models in a specific structure (Khayyam et al.; 2020). Several approaches of UML and SML combination frameworks can be observed in the literature (e.g., hierarchical structure combination (Al-Mohair et al.; 2015)). A popular combination of UML and SML in a hybrid machine learning (HML) structure is to use UML to aggregate data and then use SML for classification (Best et al.; 2022). The application of the HML framework that combines unsupervised learning and a supervised classifier can be observed in solving many practical problems where the levelled data is unavailable or limited (Best et al.; 2022). For example, Al-Mohair et al. (2015) and (Samrin and Vasumathi; 2018) used a combination of Artificial Neural Networks (ANN) with K-Mean clustering-based hybrid frameworks for detecting human skin and designing intrusion anomaly detection systems, respectively.

5.2.3 Red-Flag and Augmented AI Approaches

The RF generally refers to marking specific behaviours (most often fraudulent or irregular behaviours) using a manual, or an automated process (Sittig and Singh; 2013). It is

a well-established fraud detection technique recommended by most auditing standards (Albrecht et al.; 2018; Kramer; 2015).

Augmented AI refers to a process of having a human agent in the workflow (also known as 'Human-in-the-Loop') to review, intervene or make the final decision (Cranor; 2008). Many machine-learning models have been designed with an augmented AI process to solve practical problems in recent years (Wu et al.; 2022). Some of these studies refer to this approach as 'Augmented AI' since it is seen as augmenting/improving human performance in terms of accuracy and speed (Zheng et al.; 2017; Sorantin et al.; 2021). Many cloud service companies like Amazon are developing their platform to support these models in their platform (Amazon; 2021).

A combination of RF and Augmented AI works sequentially in a workflow and is recommended in many practical applications (Baader and Krcmar; 2018; Kim et al.; 2022). Several studies showed that this approach significantly outperforms other state-of-the-art machine learning approaches, especially for fraud detection (Gopinath et al.; 2016; Agnisarman et al.; 2019; Chai et al.; 2020; Reddy et al.; 2021; Chan et al.; 2022).

5.3 Proposed Hybrid Fraud Detection Framework

The proposed hybrid framework for identifying fraudulent users in invoicing platforms has three main phases: model building, testing and model optimization, and production. A high-level architecture of this framework is presented in Fig. 5.2. In the model-building phase, we used historical users' data from the invoicing platform and a set of small labelled data to develop the model. The following subsections discuss the different components of these three phases.

5.3.1 Model Building Phase

Feature Selection

Feature selection is a procedure of reducing dimensionality in the machine learning process. Generally, users' data from invoicing platforms come with a lot of features and considering all features may be redundant, noise-dominated and computationally infeasible (Handl and Knowles; 2006). Many SML feature selection approaches have been proposed in the literature (Chandrashekar and Sahin; 2014). Examples of such approaches include embedded (e.g., random forest (Breiman; 2001)), wrapper method (Kohavi and John; 1997), and filter methods (e.g., genetic algorithm (Hilda and Rajalaxmi; 2015)) (Guyon and Elisseeff; 2003; Cai et al.; 2018). In order to select features, we proposed a

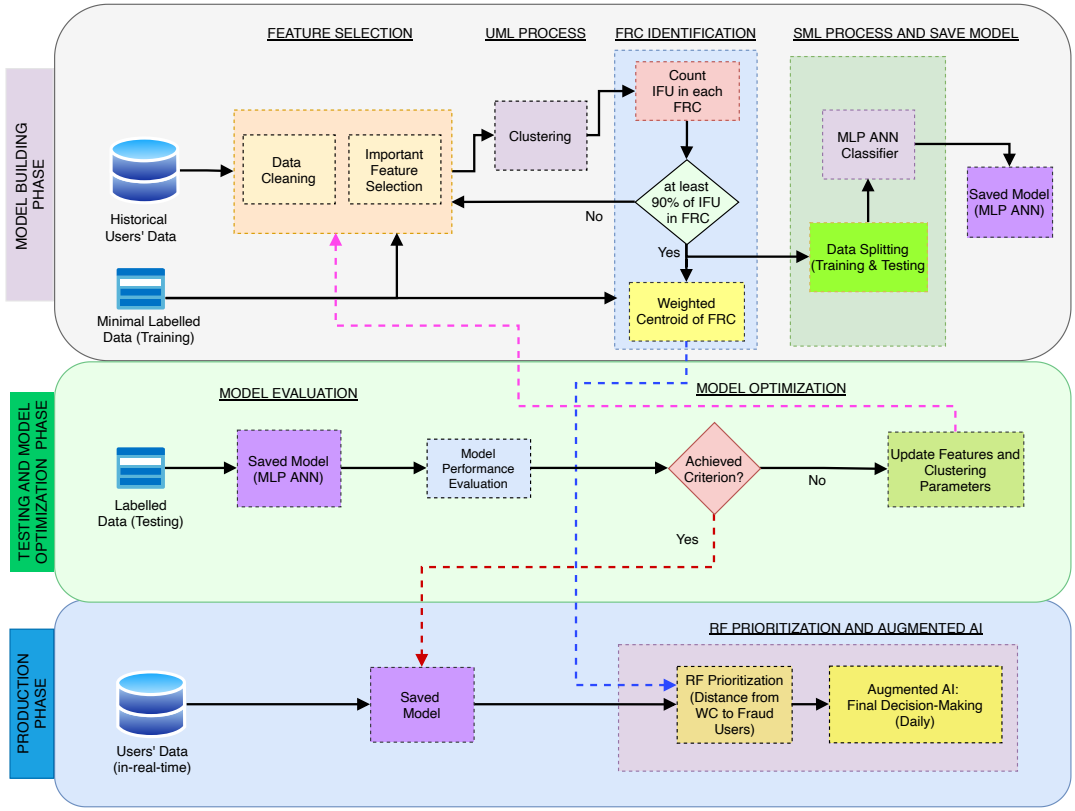


FIGURE 5.2: A high-level architecture of the proposed hybrid fraud detection framework for invoicing platform.

small set of labelled data that were generated through internal investigations prompted by affected customers' feedback and payment gateway notifications.

Generally, in SML, the feature selection problem can be formulated as an optimization problem as follows (Song et al.; 2007; Cai et al.; 2018; Taylor et al.; 2022). Let $X_t = \{X_t^{ifu} \cup X_t^{infu}\}$, $X_t^{ifu} \cap X_t^{infu} = \emptyset$ be the available small fraud/non-fraud labelled training data set where X_t^{ifu} is the Identified Fraud Users (IFU) set and X_t^{infu} is the Identified Non-Fraud Users (INFU). Also, let $\mathcal{S}$ be the corresponding full feature set in the initial step. For each $x_t \in X_t$, a corresponding fraud/non-fraud label $y_t \in \mathcal{Y}_t$ is also available. Selecting a feature set $\mathcal{F}$ from $\mathcal{S}$ can be formulated as an optimization problem (Song et al.; 2007; Kumar and Minz; 2014):

$$\arg\max_{\mathcal{F}\subseteq\mathcal{S}} \ \mathcal{Q}(\mathcal{F}, y_t), \tag{5.1}$$

$$\text{subject to:} \qquad |\mathcal{F}| \leq l \leq |\mathcal{S}|, \tag{5.2}$$

where $\mathcal{Q}(\mathcal{F}, y_t)$ estimates the performance of selected feature set $\mathcal{F}$ with respect to predicting target variable y_t and l is the upper bound on the number of selected features in $\mathcal{F}$.

Besides selecting features, there are many well-practised approaches for calculating feature importance scores, such as the permutation importance algorithm and impurity-based importance in trees (Breiman; 1998, 2001; Geurts et al.; 2006). Let $w = \{w_1, w_2, ..., w_l\}$ be the feature importance score corresponding to each feature in $\mathcal{F} = \{f_1, f_2, ..., f_l\}$.

In this paper, we used an appropriate feature selection approach, and an importance score calculator based on the available small labelled fraud/non-fraud labelled data.

UML Process: Clustering

Clustering is an unsupervised machine-learning process that identifies the input data's natural groups (clusters). In this model-building step, we use a UML clustering process to identify clusters from the unlabelled user data from the invoicing platform. Therefore, the clustering problem for the users' data from an invoicing platform can be defined more formally as follows (Breaban and Luchian; 2011):

Definition 5.1 *Let X_U be the set of m users' data from the invoicing platform with l features. We merge data set $\mathcal{X} = \{X_U \cup X_t\}$ by removing target variable y_t (labels) from X_t. Each user's data $x_i \in X$ has l numerical features (attributes). Therefore: $X = \{x_1, x_2, ..., x_m\}$ where $x_i = \{x_{i1}, x_{i2}, ..., x_{il}\} \in \mathbb{R}^m$. Also consider, $\mathcal{C} = \{C_1, C_2, ..., C_k\}$, such that $\bigcup_{p=1}^{k} C_p = \mathcal{X}$ and $C_p \cap C_q = \emptyset, \forall p, q = 1, 2, .., k; q \neq q; k \in \{1, 2, ..., card(\mathcal{C})\}$, is a possible partition (set of clusters) in the input data. The clustering problem identifies an optimal partition $\mathcal{C}^*$ by solving the following problem:*

$$\mathcal{C}^* = \arg\max_{\mathcal{C}\in\Omega} \mathcal{F}(\mathcal{C}) \tag{5.3}$$

where Ω is the set of all possible partitions in the input data $\mathcal{X}$, and $\mathcal{F}$ is a function that measures the quality of each partition $\mathcal{C} \in \Omega$.

This framework employs a classical data clustering model based on users' data from the invoicing platform.

Identifying Fraud Risk Cluster and Labelling

In this step, our proposed framework identifies the 'fraud risk cluster(s) (FRC)' from the previous UML's outputs. An FRC cluster is a possible fraudulent users' cluster that can be recognized using the cluster association of the small set of labelled data. Regarding platform economy, here we label a cluster(s) obtained from the UML process as FRC instead of fixing them as 'just' fraud. Since labelling as 'just' fraud will enforce to ban of the associated users from the platform without any final decision-making by a human. This process may irritate some users and drive them away from the platform, ultimately affecting the platform's revenue. With this consideration, we identify and label them as 'fraud risk cluster(s) (FRC)'; then, a human's final decision is made through RF-based prioritization and the augmented AI process. After this point, we refer to a user as an IFU only if it comes through RF-prioritization and an augmented AI process.

We propose Algorithm 5.1 to identify FRC by using the cluster association of the small labelled data set. Let $\mathcal{C}^* = \{C_1^*, C_2^*, ..., C_k^*\}$ be the set of clusters obtained from the previous UML clustering process that uses $\mathcal{X}$ as the input data. An $C_i^* \in \mathcal{C}^*$ is only considered as FRC if it contains at least 90% of users from X_t^{ifu}. Also, in order to avoid any infeasibility, we allow a maximum of 10% of users from X_t^{infu} in an FRC.

After identifying $\mathcal{C}^*$, users not belonging to that cluster are labelled as 'non-fraud'. Therefore, for $\mathcal{X}$, the corresponding target variables (labels) set is $\mathcal{Y}$, which has only two types of labels: fraud and non-fraud.

Weighted Center of Fraud Risk Cluster

This step calculates a weighted center of the identified FRC obtained from the previous step. We consider the IFU data set and feature importance scores (from Section 5.3.1) as the input to determine a weighted-euclidean distance (also called Mahalanobis distance (Mahalanobis; 1936))-based simplified weighted center of the FRC. The important features have more influence on the cluster center, and thus, we use the feature importance scores as input to the weights. We define the Weighted Center (WC) of FRC as follows:

Definition 5.2 *Let* ${}_cX_t^{ifu} = \{x_1, x_2, ...x_m\} \subseteq X_t^{ifu}$ *such that for every* $x_i \in {}_cX_t^{ifu}$ *implies that* $x_i \in C^*$*, i.e.,* ${}_cX_t^{ifu}$ *is the set of IFU that contains* C^**,* $|{}_cX_t^{ifu}| = m$*, and* C^* *is the identified FRC from the previous step. As discussed earlier,* $x_i \in {}_cX_t^{ifu}$ *can be*

Algorithm 5.1 FRC using the small labelled data set

procedure FRC-IDENTIFICATION($\mathcal{C}, X_t = \{X_t^{ifu} \cup X_t^{infu}\}$)

    // count fraud users contained in each cluster
    $a_i \leftarrow 0$ ▷ fraud users count for cluster $C_i \in \mathcal{C}$

    for $x_t \in X_t$ **do**:
        for $C_i \in \mathcal{C}$ **do**:
            if $x_t \in C_i$ **then**:
                $a_i = a_i + 1$ ▷ update fraud users count for cluster C_i

    // maximum fraud users contained in cluster FRC
    $\mathcal{C} =^*, a^* = \arg\max_{\forall C_i \in \mathcal{C}} a_i$ ▷ cluster with max fraud users number

    if $a_i \geq |X_t^{ifu}| * 0.9 \quad \& \quad a_i \leq |X_t^{infu}| * 0.1$ **then**:
        $y_{\mathcal{C}^*} \leftarrow$ fraud risk ▷ label $\mathcal{C}^*$ as a fraud risk cluster
        return $\mathcal{C}^*$
    else:
        Go back to the feature selection step.
        Update clustering model.

represented as a row vector that has l dimension corresponding to the selected features set $\mathcal{F} = \{f_1, f_2, ..., f_l\}$ with respective features importance scores $w = \{w_1, w_2, ..., w_l\}$, $0 \leq w_j \leq 1$. Consequently, ${}_cX_t^{ifu} = (x_{ij}), i = 1, ..., m;\ j = 1, ..., l$. Therefore, an IFU $x_c^* \in {}_cX_t^{ifu}$ is the WC of C^*, if it satisfies

$$d(x_c, x_i) = \sum_{i=1}^{m} \sum_{j=1}^{l} (1 - w_j)(x_{cj} - x_{ij})^2, \tag{5.4}$$

$$x_c^* = \underset{\forall x_i \in {}_cX_t^{ifu}}{\arg\min}\ d(x_c, x_i), \tag{5.5}$$

where $d(x_c, x_i)$, defined in Eq. 5.4, is the modified weighted distance between x_c and $x_i \in {}_cX_t^{ifu}$.

Since this model is designed for invoicing platforms where the size of the available IFU labelled data set is small, a brute-force approach can be used to find the WC for the FRC. We use the identified WC in the RF prioritization process in the production phase of this framework (see Section 5.3.3).

SML Process: Classifier Training

In this step of the proposed framework, we have a set of labelled data $\mathcal{X}$ with a corresponding target variable set $\mathcal{Y}$. Before training a classifier using SML, we have to split the labelled data into two datasets: training and testing. Let $\{\mathcal{X}_{tr}, \mathcal{Y}_{tr}\}$ and $\{\mathcal{X}_{te}, \mathcal{Y}_{te}\}$ be the training and testing datasets for the target variables sets, respectively. SML creates a learning mapping between input $\mathcal{X}_{tr}$ and the target variable $\mathcal{Y}_{tr}$ and applies this mapping to predict fraud/non-fraud classes for unseen dataset $\{\mathcal{X}_{te}, \mathcal{Y}_{te}\}$ (Cunningham et al.; 2008).

This framework employs a classical feed-forward multi-layer perceptron-based Artificial Neural Network (MLP ANN) trained with the back-propagation algorithm to predict intrusion (Bishop et al.; 1995). This MLP ANN holds an input layer, an output layer with two nodes (corresponding to labels 'fraud' and 'non-fraud'), and two hidden layers. The number of nodes in the hidden layers is determined in the testing and model optimization phase (Section 5.3.2). After optimizing and training MLP ANN, this framework saves this model for future prediction.

Note that, according to this framework definition, any new user from the invoicing platform identified as 'fraud' by this model is not an 'identified fraud user (IFU)'. This 'fraud' labelled user can only be labelled as IFU if it goes through the RF prioritization and augmented AI final decision-making process. Then the human agent can also ban this IFU user from the platform or initiate any other steps, according to the existing organizational fraud policy.

5.3.2 Testing and Model Optimization Phase

In the testing phase, this framework uses the labelled data set $\mathcal{X}_{tr}$, corresponding with target variables set $\mathcal{Y}_{tr}$ obtained from the previous step, to optimize the SML model. There are two objectives in this optimization process of MLP ANN: (a) determining the best combination of feature space, $\mathcal{F} \subseteq \mathcal{S}$, and (b) determining the optimal number of nodes n (perceptron) in the hidden layer of the MLP ANN.

To evaluate the framework's performance, the prediction accuracy, i.e., the total number of correct predictions, may not be adequate when the data set is imbalanced (Chawla; 2009). Therefore, we adopt the well-known measures for information retrieval: precision (P), recall (R), and F_1-measure (Manning; 2008).

The outer layer of MLP has two nodes (labels): fraud and non-fraud. Therefore, in terms of prediction correctness, true positive (TP), true negative (TN), false positive

(FP), and false negative (FN) can be defined as follows: TP: Model predicts a fraud user for a user who was a true fraud user; TN: Model predicts non-fraud user for a user who was a non-fraud user; FP: Model predicts a fraud user for a user that was a non-fraud user; FN: Model predicts a non-fraud user for a user who was a fraud user. Precision (P), recall (R), and F_1-measure metrics of the trained MLP ANN model can be calculated as follows (Powers; 2011):

$$P = \frac{TP}{TP + FP}, \tag{5.6}$$

$$R = \frac{TP}{TP + FN}, \tag{5.7}$$

$$F_1 - \text{measure} = \frac{2PR}{P + R}. \tag{5.8}$$

Suppose the stopping conditions of evaluation metrics are achieved for the MLP ANN model. In that case, we save this model for predicting fraud/non-fraud labels for any new user signing up on the invoicing platform. The stopping conditions may differ based on the specifics of the industry and organizational preferences.

In addition to the framework's accuracy during the development period, we propose another level of post-production benchmarking (see Section 5.4.8) between the classical Payment Gateway's Red Flag (PGRF) approach and our proposed HFDF with RF prioritization and the augmented AI approach.

5.3.3 Production Phase

In the production phase of this framework, we proposed a combination of the following RF prioritization and augmented AI process for the users identified as 'fraud' by the SML classifier (see Section 5.3.1).

RF Prioritization

This step generates a prioritized list P_f of size L based on the weighted distance $d(x_c, x_i)$, where x_c is the FRC C^*, $x_i \in X_f$ and X_f is the set of users predicted as 'fraud' by the trained SML classifier (Section 5.3.1). The threshold of size L of P_f is determined by the resource capacity in the augmented AI process in the invoicing platform's organization. After generating P_f, this framework marks these users with red flags and forwards them

to the following augmented AI process. This step uses Algorithm 5.2 to generate P_f from X_f.

Algorithm 5.2 RF Prioritization

procedure RF-PRIORITIZATION(x_c, X_f, L)

    // calculate distance using Eq. 5.4
    for $x_i \in X_f$ **do**:
        $d_i \leftarrow d(x_c, x_i)$ ▷ weighted dist. between x_c and x_i

    // ascending list of users based on d_i
    $P_f \leftarrow \{x_1, x_2, ..., x_L\}$;
     where $\forall x_i \in P_f \implies x_i \in X_f$,
     and $d(x_c, x_i) \leq d(x_c, x_j) \implies \forall x_i, x_j \in X_f,\ i \leq j \leq L$

return P_f

Augmented AI Process: Final Decision Making

The final step of this proposed framework uses an augmented AI process for final decision-making. In this process, a human generally goes through the RF prioritized list P_f and manually checks each suspected user's profile on a daily basis. Then based on their judgment, they label them as IFU or INFU. Our inspiration for this approach came from implementing this framework in an empirical setting where a human is needed to make the final decision at the end of the process (e.g., deactivating a user account, banning it from the platform, or initiating legal steps).

5.4 Fraud Detection in Invoicing Platforms: A Case Study

This section presents a case study of the proposed framework for identifying fraudulent users with our research partner's invoicing platform. The research partner is a CB-AIS company, and the majority of their customer are SMEs. The company applies a PGRF approach to identify fraud. They identify fraud cases through complaints from affected customers (Kranacher and Riley; 2019) or through the notification from payment-gateway service providers (Chan et al.; 2022). These processes allow the platform to maintain identified fraudulent user (labelled) data. Their PGRF process has two main steps. First, the customer support team creates red flags based on customer feedback (e.g., emails and calls) and repeated transaction declined alerts from payment gateway services. Later, the fraud detection team randomly goes through those

red-flagged users and labels the corresponding user accounts as either fraud/non-fraud based on their judgment.

Fraudulent users are spamming other customers, businesses, or individuals and costing the invoicing platform significant financial harm as well as the risk of losing reputation if the fraud news is publicized. Hence, the sooner we identify the fraudulent user accounts, the better for our partner to protect its financial and reputational integrity. So, our goal for this case study was to replace the manual RF prioritization steps and to design a framework for automating this process to effectively and efficiently identify fraudulent users.

The following subsections present a procedure to design an automated red flag-based prioritization process in our partner's invoicing platform by using the proposed framework (given in Fig. 5.2). This first automated RF prioritization and the next augmented AI approach complete the fraud detection process in our partner's invoice platform.

5.4.1 Initial Data Collection and Processing

We collected data on all users from our research partners' invoicing service platform who signed up between August 01, 2018, and July 30, 2019. Our decision to select this specific period was to avoid any influence on user activities due to the COVID-19 pandemic. During this period, 452,627 general user accounts signed up for this invoicing service platform. Each user data had 15 features, such as invoice creation count, client count, and the number of emails sent to clients.

We also collected historically labelled identified fraud users' data from our research partner. Over the years, they identified 305 fraud user accounts and labelled them accordingly. To create a set of fraud/non-fraud labelled data, we randomly collected 400 accounts that were created by legitimate users and labelled them as non-fraud. The choice of 400 is mainly dictated by the available human resources that will be needed in performing manual checks. We identified these non-fraud users by manually examining each account information individually (e.g., account activities, transactions, business information, website, and telephone). Therefore, we identified 305 fraud users (IFU), 400 non-fraud accounts (INFU), and $452,627$ general users (GUA). We used the unlabelled GUA data set for building the UML clustering model and small labelled datasets (IFU & INFU) for important feature selection (see Section 5.3.1) and training an SML classifier (see Section 5.4.6). Table 5.1 presents the number of IFU and INFU users in the training and testing data sets.

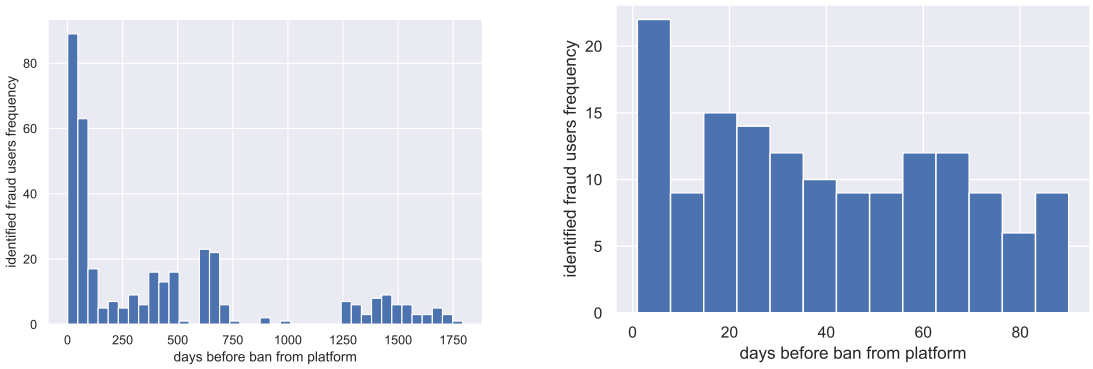
Dataset	IFU	INFU	Total
Unlabelled (UML Clustering)	–	–	452,627
Labelled (SML Training)	305	400	705

TABLE 5.1: The collected unlabelled and labelled data.

Therefore, the given small labelled training set contained 705 users (305 IFU and 400 INFU) in total. Furthermore, since we implemented weekly segmented structured HFDF, we processed 13 sets of weekly segmented (WS) structures. We will refer to these training data sets as WS-01, WS-02, ..., and WS-13. The training data sets include cumulative weekly data. For example, WS-01 contained the first week of 705 users' activities after the signup date. Similarly, WS-02 had the first two weeks of activities after the signup date, and so on.

5.4.2 Active Lifespan of Fraudulent Users

In order to fix the data range, first, we decided to investigate the active lifespan of fraudulent users. To understand the lifespan of the identified fraud users, we collected data on the number of days that they survived before being banned from the platform.



(A) (B)

FIGURE 5.3: Lifespan of IFU before being banned from the platform.

Fig. 5.3 presents the lifespan of the IFU before being banned from the platform after the signup date. Each column in the histogram plots in Fig. 5.3a and Fig. 5.3b represent a lifespan of 91 days and 7 days, respectively. Fig. 5.3a indicates that the highest number of fraud accounts banned was within 91 days after the signup date. This phenomenon can be explained by the fact that our partner CB-AIS company offers a

free trial account for the first 91 days after signup to their invoicing platform. Therefore, fraudsters are using free trial accounts to do illicit activities, which leads some of them to get banned from the platform. Further, we noticed in Fig. 5.3b, a more granular illustration of IFU's lifespan, that over the first 13 weeks after the signup date, the rate of IFUs getting banned from the platform tends to gradually decrease.

5.4.3 Weekly Segmented Structured HFDF

The primary goal of any fraud detection model is to identify fraud accounts on the platform as early as possible. Given that most fraud users were active within the first 13 weeks (91 days) after their signup date, we implemented a weekly segmented structure of the proposed HFDF for our partner's invoicing platform, as shown in Fig. 5.4. Another reason for the weekly segmented structured HFDF was as follows: if a user engaged in suspicious activities right after the signup date, we did not need to wait for another 12 weeks' data to predict their fraud/non-fraud status.

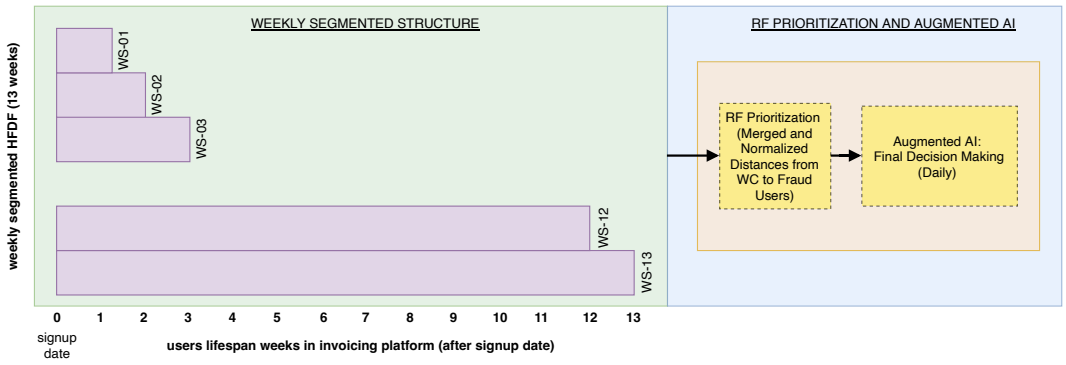


FIGURE 5.4: Implemented weekly segmented structure of the hybrid fraud detection framework.

In this weekly segmented structure, there were thirteen HFDFs (WS-01, WS-02, ..., WS-13). The i-th segment of the HFDF, WS-i, was designed for users whose lifespan after signup on the invoicing platform is more than $(i-1)$ weeks and less than i weeks. In addition, a user whose lifespan is less than 13 weeks on this platform will be tested on only one of the 13 weekly segmented HFDFs. So, for example, if the lifespan of users after signup is less than one week, then they will be tested by WS-01 HFDF. Similarly, when the lifespan is more than one week but less than two weeks, then the corresponding users will be tested by WS-02 HFDF.

In the production phase, these segmented structured HFDFs predict fraud users and merge to create a dataset X_f^{ws}, that is then forwarded to the RF prioritization and

augmented AI process. The RF prioritization step creates an ascending list P_f^{ws} of size L users based on a normalized distance between x_c and $x_i \in X_f^{ws}$.

5.4.4 Important Features Selection

Random forest-based feature selection is fast and can select a low-cost subset of informative features with better performance than any other state-of-art methods in real-world problems (Zhou et al.; 2016). We used the random forest (Breiman; 2001) approach to identify important features for the HFDF from the small labelled data set.

Fig. 5.5 presents the list of important selected features and importance scores for HFDF model building. There are 8 important features, and the top 5 features are (in descending order): 'email_sent' 'create_item', 'create_expense', 'invoice_count', and 'client_count'. All of these features are related to invoices and can only be accessed through the invoicing platform. On the other hand, two payment gateway features ('declined_electronic_payment' and 'enable_electronic_payment') are not at the top in terms of importance scores. Therefore, the classical PGRF model, which relies on payment gateway notifications, is not very effective in identifying fraudulent users.

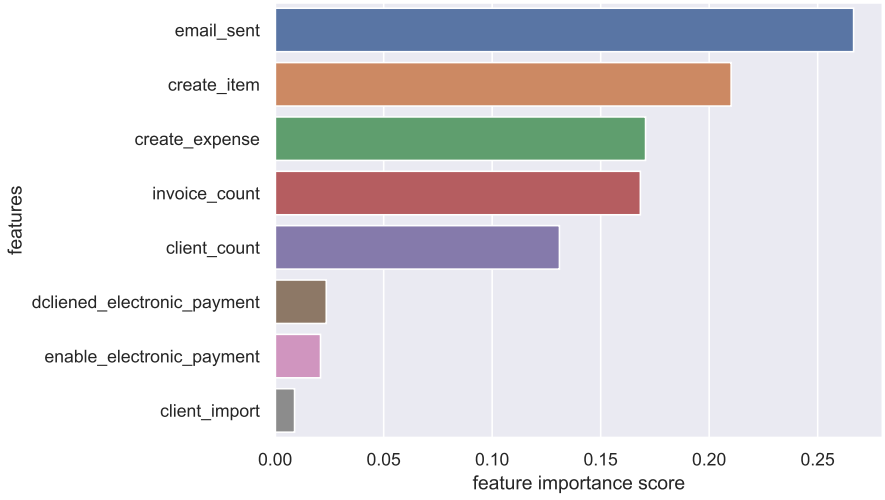


FIGURE 5.5: Important features with corresponding scores for HFDF model building.

5.4.5 Clustering and Identifying FRC

In this case study, we used Gaussian Mixture Model (GMM) (McLachlan and Basford; 1988; Bouman et al.; 1997) clustering in the UML process to identify clusters from the

WS-i's collected process data, $i = 1, \ldots, 13$. The GMM clustering algorithm has been extensively studied for its effectiveness and efficiency (Bouman et al.; 1997) and shows better clustering performance with complex multi-modal data (Zhang et al.; 2021). As a result, it is widely applied in diverse fields, such as image segmentation (Xie et al.; 2013).

Weekly Segment	**GMM Outputs Clusters**							
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
WS-01 (n=7)	198573	**17279** (IFU:305)	145174	1059	61113	573	14526	–
WS-02 (n=8)	2126	200777	8939	988	144948	59136	14597	**6787** (IFU:305)
WS-03 (n=6)	15580	144747	568	200942	59173	**17289** (IFU:305)	–	–
WS-04 (n=7)	200933	560	14492	144493	59436	**17292** (IFU:305)	1093	–
WS-05 (n=8)	6706	559	200944	14489	56914	**17297** (IFU:305)	1096	140331
WS-06 (n=7)	6713	568	201342	15580	56518	**17289** (IFU:305)	140289	–
WS-07 (n=6)	201253	**17852** (IFU:305)	14443	144352	59257	1142	–	–
WS-08 (n=8)	201568	17337 (IFU: 6)	1186	144527	59287	**11704** (IFU: 299)	14711	2245
WS-09 (n=6)	201641	**31286** (IFU:305)	15897	6723	56693	140325	–	–
WS-10 (n=8)	17389	201668	1239	144471	59243	14658	2205	**11692** (IFU:305)
WS-11 (n=6)	201865	**31286** (IFU: 305)	15897	58534	6775	138208	–	–
WS-12 (n=8)	201968	17289 (IFU:299)	15897	144465	**11709** (IFU:6)	58949	490	1798
WS-13 (n=7)	144435	**28831** (IFU:305)	201994	1269	58953	2455	14628	–

TABLE 5.2: GMM cluster sizes and the number of IFU (if any) for each of the weekly segments.

Table 5.2 presents the output clusters of the GGM clustering algorithm corresponding to each WS-i; $i = 1, 2, \ldots, 13$. The Elbow method was used to determine the number of clusters n corresponding to each segment's model (Thorndike; 1953; Goutte et al.; 1999). There were 215 IFUs in each WS-i training data. Algorithm 5.1 was used to identify FRC for each WS-i using the corresponding IFU cluster association from WS-i training data. The identified Fraud Risk Clusters (FRC) are presented in Table 5.2 in bold fonts. We note that with the exception of WS-8 and WS-12, each of the WS-i's had one cluster that contained all 305 IFUs. For WS-8 and WS-12, clusters C_6 and C_2 had 211 IFUs (more than 90% of 215 IFUs). Therefore, in each WS-i, we called the cluster that contained more than 90% of IFUs an FRC and labelled all users in these clusters as 'fraud'. Additionally, we labelled all users belonging to other clusters as 'non-fraud'.

At this step, we had 13 labelled (fraud/non-fraud) data sets $\mathcal{X}^i$; $i = 1, 2, ..., 13$ corresponding with the 13 weekly segments WS-i; $i = 1, 2, ..., 13$. Each $\mathcal{X}^i$ contained the first i-th weeks' activities (after the signup date) of all users who signed up between August 01, 2018, and July 30, 2019.

5.4.6 Classifier Model Training and Testing

To train and test an SML process, first, we split the labelled data $\mathcal{X}^i;\ i = 1, 2, ..., 13$ a training dataset $\mathcal{X}^i_{tr}; i = 1, 2, ..., 13$ and a testing dataset $\mathcal{X}^i_{te}; i = 1, 2, ..., 13$ corresponding to the 13 weekly segments WS-$i; i = 1, 2, ..., 13$.

In the SML process, we used a Multi-Layer Perceptron (MLP)-based Artificial Neural Network (ANN) to train the fraud/non-fraud classifier model using labelled data from the previous step. Due to the weekly segmented structure, we trained 13 MLP ANNs by using training datasets $\mathcal{X}^i_{tr};\ i = 1, 2, ..., 13$ for each corresponding WS-$i;\ i = 1, 2, ...13$. In the testing and optimization phase, we used testing datasets $\mathcal{X}^i_{te};\ i = 1, 2, ..., 13$ for each corresponding WS-$i;\ i = 1, 2, ...13$ to evaluate the model performance. To optimize each week's MLP ANN, the number of nodes in hidden layers was investigated using 3–16 nodes and 500 maximum iterations. We used an 85% threshold for each evaluation metric (precision, recall, F1-score, and accuracy) to optimize each MLP ANN.

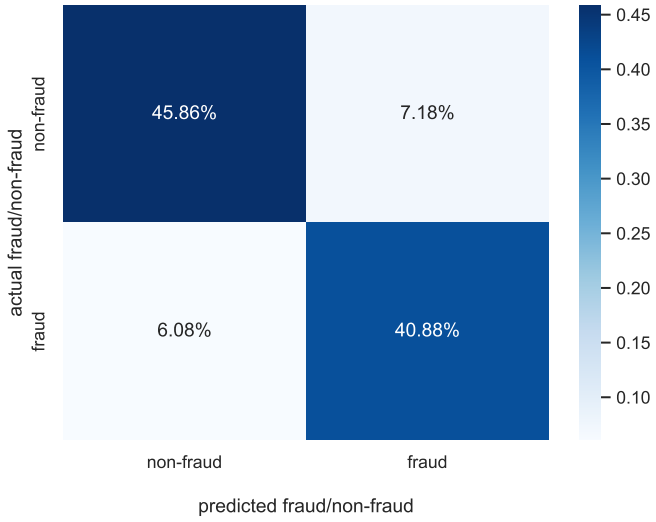


FIGURE 5.6: Confusion matrix for WS-01's MLP ANN based on the small amount of labelled testing data.

Label	**Precision (P)**	**Recall (R)**	**F1-Score**
non-fraud	0.88	0.86	0.87
fraud	0.85	0.87	0.86
Accuracy			88%

TABLE 5.3: Precision, recall, F1-Score, and accuracy for the WS-01's MLP ANN based on the small labelled testing data.

Fig. 5.6 shows the confusion matrix for WS-01's MLP ANN. Table 5.3 presents the evaluation metrics scores for WS-01's MLP ANN. All of the metrics achieved the 85% threshold for this case study.

Following a similar process, we tested and optimized the rest of the MLP ANN for WS-i; $i = 2, 3, ..., 13$. Finally, we saved these models for identifying fraudulent users in the invoicing platform.

5.4.7 Implementation

We implemented this weekly segment structure of HFDF by using Python with Scikit-learn (Pedregosa et al.; 2011).

5.4.8 Post-Production Benchmarking

This section presents the post-production benchmarking between the previously used PGRF by our partner organization and our proposed HFDF. Based on our historical analysis research partner's data, on average, about 20,000 declined payments, and the payment gateway flags 6,000 associate users in a day. The success rate of identifying fraudulent users by randomly selecting a user from the declined list is very low.

We teamed up with the fraud detection team of this organization to perform the augmented AI process in the proposed HFDF. We collected this team's performance data in the first 20 days using the HFDF framework. Table 5.4 presents a success rate comparison between PGRF and HFDF frameworks. In one year (roughly 260 work days), the fraud detection team detected 305 IFUs by using the PGRF framework. On average, their daily fraud detection success rate was 1.17 IFUs. On the other hand, while using the HFDF framework for the same period, they identified 122 IFUs in 20 days. Therefore, on average, their daily success rate increased to 6.10 IFUs, significantly improving their performance. This improvement is largely attributed to the predictability power of our proposed method.

Model	**Run Period (in days)**	**IFU**	**Success Rate (daily)**
PGRF	260	305	1.17
HFDF	20	122	6.10

TABLE 5.4: Comparison between PGRF and HFDF frameworks.

5.5 Concluding Remark, Limitation, and Future Work

This study proposed a hybrid framework for identifying fraudulent users in invoicing platforms. We used UML and SML processes that use a small labelled data set to develop the hybrid framework. In addition, a combination of weighted center-based RF prioritization and an augmented AI approach was also introduced in the final decision-making process. As a result, this framework can be implemented in an application setting where only a small labelled data set is available and a human agent is required in the final decision-making step. Furthermore, for the specific case study presented in this paper, we implemented the proposed HFDF in a weekly segmented structure to identify fraudulent users in our partner's invoicing platform. The post-production benchmarking between classical PGRF and HFDF showed that the organization's fraud detection performance had significantly improved after using this proposed framework. In other words, we can say that the proposed HFDF framework is working as an augmented tool for the fraud detection team to improve their performance.

The first limitation of this proposed framework is that it heavily depends on the available the given small amount of labelled data set to determine FRC and training MLP ANN models. If the given small labelled data set is biased or does not entirely represent the users in the platform, then the ultimate model would have inferior performance. Furthermore, though this model is specifically designed for an application setting where human input is required in the final decision-making step, it may also create bias in the process. Finally, in the post-production benchmarking, we did not control all variables (e.g., the same team members in the fraud detection team and the rate of new account signup) during the testing phase. As a result, team members might show improved performance under the newly designed fraud detection framework. In future, randomized control testing needs to be done in the post-production benchmarking between PGRF and HFDF frameworks for a longer period.

This research can be extended in several ways, first, by adding another reinforcement layer to this framework so that the model can learn from the augmented AI inputs in real-time. Second, it is worth investigating other approaches for the FRC identification step, such as redesigning it as a semi-supervised learning process. Finally, in the event of limited expert resources, it becomes important to develop a schedule for engaging the decision-makers depending on their availability and experience level.

Funding

We acknowledge support from the Natural Sciences and Engineering Research Council (NSERC) Discovery (Award Number: RGPIN-2020-06792) and Mitacs Accelerate Fellowship Program (Award Number: IT16025) programs for their support of this project.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

Chapter References

Abdallah, A., Maarof, M. A. and Zainal, A. (2016). Fraud detection system: A survey, *Journal of Network and Computer Applications* **68**: 90–113.

Agnisarman, S., Lopes, S., Madathil, K. C., Piratla, K. and Gramopadhye, A. (2019). A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection, *Automation in Construction* **97**: 52–76.

Al-Hashedi, K. G. and Magalingam, P. (2021). Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019, *Computer Science Review* **40**: 100402.

Al-Mohair, H. K., Saleh, J. M. and Suandi, S. A. (2015). Hybrid human skin detection using neural network and k-means clustering technique, *Applied Soft Computing* **33**: 337–347.

Albrecht, W. S., Albrecht, C. O., Albrecht, C. C. and Zimbelman, M. F. (2018). *Fraud examination*, Cengage Learning.

Amazon (2021). Cyber defence in the age of AI, Smart Societies and Augmented Humanity. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/ruvuj5*

Asatiani, A., Apte, U., Penttinen, E., Rönkkö, M. and Saarinen, T. (2019). Impact of accounting process characteristics on accounting outsourcing-comparison of users and non-users of cloud-based accounting information systems, *International Journal of Accounting Information Systems* **34**: 100419.

Asatiani, A. and Penttinen, E. (2015). Managing the move to the cloud–analyzing the risks and opportunities of cloud-based accounting information systems, *Journal of Information Technology Teaching Cases* **5**(1): 27–34.

Baader, G. and Krcmar, H. (2018). Reducing false positives in fraud detection: Combining the red flag approach with process mining, *International Journal of Accounting Information Systems* **31**(July 2016): 1–16.

Balayan, V., Saleiro, P., Belém, C., Krippahl, L. and Bizarro, P. (2020). Teaching the machine to explain itself using domain knowledge, *arXiv preprint* .

Barclays (2022). Invoice Fraud: How to protect your organisation from fraudsters. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/ktdncj*

Best, L., Foo, E. and Tian, H. (2022). Utilising k-means clustering and naive bayes for IoT anomaly detection: A hybrid approach, *Secure and Trusted Cyber Physical Systems*, pp. 177–214.

Bhattacharyya, S., Jha, S., Tharakunnel, K. and Westland, J. C. (2011). Data mining for credit card fraud: A comparative study, *Decision Support Systems* **50**(3): 602–613.

Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*, Oxford university press.

Bouman, C. A., Shapiro, M., Cook, G., Atkins, C. B. and Cheng, H. (1997). Cluster: An unsupervised algorithm for modeling gaussian mixtures.

Breaban, M. and Luchian, H. (2011). A unifying criterion for unsupervised clustering and feature selection, *Pattern Recognition* **44**(4): 854–865.

Breiman, L. (1998). Rejoinder: Arcing classifiers, *The Annals of Statistics* **26**(3): 841–849.

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

Cai, J., Luo, J., Wang, S. and Yang, S. (2018). Feature selection in machine learning: A new perspective, *Neurocomputing* **300**: 70–79.

Cassara, J. A. (2015). *Trade-based money laundering: the next frontier in international money laundering enforcement*, John Wiley & Sons.

Cedillo, P., García, A., Cárdenas, J. D. and Bermeo, A. (2018). A systematic literature review of electronic invoicing, platforms and notification systems, *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*, IEEE, pp. 150–157.

Chai, C., Cao, L., Li, G., Li, J., Luo, Y. and Madden, S. (2020). Human-in-the-loop outlier detection, *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 19–33.

Chakraborty, J., Tu, H., Majumder, S. and Menzies, T. (2021). Can we achieve fairness using semi-supervised learning?, *arXiv preprint* .

Chan, L., Hogaboam, L. and Cao, R. (2022). Artificial intelligence in accounting and auditing, *Applied Artificial Intelligence in Business*, Springer, pp. 119–137.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods, *Computers & Electrical Engineering* **40**(1): 16–28.

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview, *Data Mining and Knowledge Discovery Handbook* pp. 875–886.

Christauskas, C. and Miseviciene, R. (2012). Cloud–computing based accounting for small to medium sized business, *Engineering Economics* **23**(1): 14–21.

Cleary, P. and Quinn, M. (2016). Intellectual capital and business performance: An exploratory study of the impact of cloud-based accounting and finance infrastructure, *Journal of Intellectual Capital* **17**(2): 255–278.

Cranor, L. F. (2008). A framework for reasoning about the human in the loop, *Proceedings of the 1st Conference on Usability, Psychology, and Security*, pp. 1–15.

Cunningham, P., Cord, M. and Delany, S. J. (2008). Supervised learning, *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval* pp. 21–49.

Dejong, M. (2018). Tax crimes: The fight goes digital, *Organisation for Economic Cooperation and Development. The OECD Observer* pp. 1–3.

Eldalabeeh, A. R., Al-Shabil, M. O., Almuiet, M. Z., Bany Baker, M. and E'lemiat, D. (2021). Cloud-based accounting adoption in jordanian financial sector, *The Journal of Asian Finance, Economics and Business* **8**(2): 833–849.

Ferrara, C., Carlucci, M., Grigoriadis, E., Corona, P. and Salvati, L. (2017). A comprehensive insight into the geography of forest cover in italy: Exploring the importance of socioeconomic local contexts, *Forest Policy and Economics* **75**: 12–22.

Forestier, G. and Wemmert, C. (2016). Semi-supervised learning using multiple clusterings with limited labeled data, *Information Sciences* **361**: 48–65.

Geurts, P., Ernst, D. and Wehenkel, L. (2006). Extremely randomized trees, *Machine Learning* **63**(1): 3–42.

Gong, Y., Zhang, Y. and Alharithi, M. (2022). Supply chain finance and blockchain in operations management: A literature review, *Sustainability* **14**(20): 13450.

Gopinath, D., Jain, S. and Argall, B. D. (2016). Human-in-the-loop optimization of shared autonomy in assistive robotics, *IEEE Robotics and Automation Letters* **2**(1): 247–254.

Goutte, C., Toft, P., Rostrup, E., Nielsen, F. Å. and Hansen, L. K. (1999). On Clustering fMRI Time Series, *NeuroImage* **9**(3): 298–310.

GrantThornton (2021). Invoice fraud: How it works and five ways to prevent it. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/hnaedj*

Guerar, M., Merlo, A., Migliardi, M., Palmieri, F. and Verderame, L. (2020). A fraud-resilient blockchain-based solution for invoice financing, *IEEE Transactions on Engineering Management* **67**(4): 1086–1098.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**(Mar): 1157–1182.

Hady, M. F. A. and Schwenker, F. (2013). Semi-supervised learning, *Handbook on Neural Information Processing* pp. 215–239.

Hamelers, L. (2021). *Detecting and explaining potential financial fraud cases in invoice data with machine learning*, Master's thesis, University of Twente.

Handl, J. and Knowles, J. (2006). Feature subset selection in unsupervised learning via multiobjective optimization, *International Journal of Computational Intelligence Research* **2**(3): 217–238.

Hilda, G. T. and Rajalaxmi, R. (2015). Effective feature selection for supervised learning using genetic algorithm, *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pp. 909–914.

Karim, M., Islam, T., Beyan, O., Lange, C., Cochez, M., Rebholz-Schuhmann, D., Decker, S. et al. (2022). Explainable ai for bioinformatics: Methods, tools, and applications, *arXiv preprint* .

Kariyawasam, A. (2019). Analysing the impact of cloud-based accounting on business performance of smes, *The Business & Management Review* **10**(4): 37–44.

Kearse, N. (2020). What is supplier invoice fraud and how do you prevent it? Hub. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/6fywno*

Khayyam, H., Jamali, A., Bab-Hadiashar, A., Esch, T., Ramakrishna, S., Jalili, M. and Naebe, M. (2020). A novel hybrid machine learning algorithm for limited and big data modeling with application in industry 4.0, *IEEE access* **8**: 111381–111393.

Kim, S., Mai, T.-D., Han, S., Park, S., Khanh, T. N. D., Soh, J., Singh, K. and Cha, M. (2022). Active learning for human-in-the-loop customs inspection, *IEEE Transactions on Knowledge and Data Engineering* pp. 1–1.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection, *Artificial Intelligence* **97**(1-2): 273–324.

Kramer, B. (2015). Trust, but verify: Fraud in small businesses, *Journal of Small Business and Enterprise Development* **22**(1): 4–20.

Kranacher, M.-J. and Riley, R. (2019). *Forensic accounting and fraud examination*, John Wiley & Sons.

Kruber, F., Wurst, J. and Botsch, M. (2018). An unsupervised random forest clustering technique for automatic traffic scenario categorization, *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2811–2818.

Kumar, P., Murphy, A., Werner, S. and Rougeaux, C. (2022). The fight against money laundering: Machine learning is a Game Changer. McKinsey & Company. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/mn66cp*

Kumar, V. and Minz, S. (2014). Feature selection: a literature review, *SmartCR* **4**(3): 211–229.

Lee, H. C. (2016). Can electronic tax invoicing improve tax compliance? A case study of the Republic of Korea's electronic tax invoicing for value-added tax, *World Bank Policy Research Working Paper* (7592).

Li, N., Martin, A. and Estival, R. (2018). Combination of supervised learning and unsupervised learning based on object association for land cover classification, *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8.

Li, T., Kou, G., Peng, Y. and Philip, S. Y. (2021). An integrated cluster detection, optimization, and interpretation approach for financial data, *IEEE Transactions on Cybernetics* **52**(12): 13848–13861.

Ma, D., Fisher, R. and Nesbit, T. (2021). Cloud-based client accounting and small and medium accounting practices: Adoption and impact, *International Journal of Accounting Information Systems* **41**: 100513.

Maadi, M., Akbarzadeh Khorshidi, H. and Aickelin, U. (2021). A review on human–ai interaction in machine learning and insights for medical applications, *International Journal of Environmental Research and Public Health* **18**(4): 2121.

Mahalanobis, P. C. (1936). On the generalised distance in statistics, *Proceedings of the National Institute of Science of India*, Vol. 12, pp. 49–55.

Manning, C. D. (2008). *Introduction to information retrieval*, Syngress Publishing.

McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*, Vol. 38, M. Dekker New York.

Pai, P.-F., Hsu, M.-F. and Wang, M.-C. (2011). A support vector machine-based model for detecting top management fraud, *Knowledge-Based Systems* **24**(2): 314–321.

Pavía, J. M., Veres-Ferrer, E. J. and Foix-Escura, G. (2012). Credit card incidents and control systems, *International Journal of Information Management* **32**(6): 501–503.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011). Scikit-learn: Machine learning in python, *The Journal of Machine Learning Research* **12**: 2825–2830.

Pise, N. N. and Kulkarni, P. (2008). A survey of semi-supervised learning methods, *2008 International conference on computational intelligence and security*, Vol. 2, pp. 30–34.

Popivniak, Y. (2019). Cloud-based accounting software: choice options in the light of modern international tendencies, *Baltic Journal of Economic Studies* **5**(3): 170–177.

Powers, D. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation, *Journal of Machine Learning Technologies* **2**(1): 37–63.

Raghavan, P. and El Gayar, N. (2019). Fraud detection using machine learning and deep learning, *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 334–339.

Reddy, S., Dragan, A. and Levine, S. (2021). Pragmatic image compression for human-in-the-loop decision-making, *Advances in Neural Information Processing Systems* **34**: 26499–26510.

Samrin, R. and Vasumathi, D. (2018). Hybrid weighted k-means clustering and artificial neural network for an anomaly-based network intrusion detection system, *Journal of Intelligent Systems* **27**(2): 135–147.

Sittig, D. F. and Singh, H. (2013). A red-flag-based approach to risk management of ehr-related safety concerns, *Journal of Healthcare Risk Management* **33**(2): 21–26.

Song, L., Smola, A., Gretton, A., Borgwardt, K. M. and Bedo, J. (2007). Supervised feature selection via dependence estimation, *Proceedings of the 24th international conference on Machine Learning*, pp. 823–830.

Sorantin, E., Grasser, M. G., Hemmelmayr, A., Tschauner, S., Hrzic, F., Weiss, V., Lacekova, J. and Holzinger, A. (2021). The augmented radiologist: Artificial intelligence in the practice of radiology, *Pediatric Radiology* pp. 1–13.

Stamler, R. T., Marschdorf, H. J. and Possamai, M. (2014). *Fraud prevention and detection: warning signs and the red flag system*, CRC Press.

Taylor, P., Griffiths, N., Hall, V., Xu, Z. and Mouzakitis, A. (2022). Feature selection for supervised learning and compression, *Applied Artificial Intelligence* pp. 1–35.

Thorndike, R. L. (1953). Who Belongs in a Family?, *Psychometrica* (18): 267–276.

Trialopedia (2021). A list of accounting software offering a free trial. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/yvxlfc*

U.S. Attorneyś Office (2020). Four individuals charged with $19 million fraudulent invoicing scheme targeting Amazon's vendor system. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/dj6xqs*

Wang, J. and Biljecki, F. (2022). Unsupervised machine learning in urban studies: A systematic review of applications, *Cities* **129**: 103925.

White, A. H. (2017). 6 ways to spot and prevent invoice fraud. [Online]. [Accessed: 23-Feb-2023].
URL: *https://rb.gy/yccy8p*

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T. and He, L. (2022). A survey of human-in-the-loop for machine learning, *Future Generation Computer Systems* .

Xie, C.-H., Chang, J.-Y. and Liu, Y.-J. (2013). Estimating the number of components in gaussian mixture models adaptively for medical image, *Optik* **124**(23): 6216–6221.

Xie, R., Mao, W. and Shi, G. (2019). Electronic invoice authenticity verifying scheme based on signature recognition, *Journal of Physics: Conference Series*, Vol. 1213, IOP Publishing, p. 032019.

Zhang, Y., Li, M., Wang, S., Dai, S., Luo, L., Zhu, E., Xu, H., Zhu, X., Yao, C. and Zhou, H. (2021). Gaussian mixture model clustering with incomplete data, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **17**(1s): 1–14.

Zheng, N.-n., Liu, Z.-y., Ren, P.-j., Ma, Y.-q., Chen, S.-t., Yu, S.-y., Xue, J.-r., Chen, B.-d. and Wang, F.-y. (2017). Hybrid-augmented intelligence: collaboration and cognition, *Frontiers of Information Technology & Electronic Engineering* **18**(2): 153–179.

Zhou, Q., Zhou, H. and Li, T. (2016). Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features, *Knowledge-based Systems* **95**: 1–11.

Chapter 6

Conclusions and Recommendations

6.1 Concluding Remarks

This thesis drives substantial progress toward beginning a unified discussion of an important class of clustering problems and developing clustering frameworks for big empirical datasets with real-world applications. One of the primary driving forces of this thesis was to develop data-driven, action-oriented, implementable data clustering-based frameworks that deliver practical solutions to complex data clustering problems in real-world scenarios. In addition to designing these frameworks, this thesis demonstrates the applicability of these frameworks in diverse empirical settings. The overall contributions of this thesis span four different areas. The contributions' specific concluding remarks are presented in the following subsections.

6.1.1 Unified Discussion on the Correlation Clustering Problem

Correlation clustering is a multidisciplinary problem that identifies clusters in the presence of qualitative information about objects' mutual similarities or dissimilarities. It can be applied to many practical problems. Over the years, this problem appeared in the research literature in different scientific areas in various forms and names. The study in Chapter 2 presents a unified discussion, including a detailed discussion of mathematical formulations and solution approaches, to enhance the cross-fertilization of knowledge and mitigate gaps between disciplinary approaches to the Correlation clustering problem. In addition, a taxonomic development for several variant classes, solution procedures and applications is presented in this chapter. Furthermore, it provides a bibliometric-based

analysis to explore the collaborations, citation progressions, dominant research topics, and knowledge clusters in this area.

6.1.2 Defining and Identifying Common-Knowledge Network-based Research Communities

The study presented in Chapter 3 illustrates a common-knowledge network-based model to investigate the interdisciplinary research productivity among researchers over time. This study considers publication keywords as the knowledge elements that represent authors' areas of expertise and formulate a network based on the commonality of knowledge elements (keywords) among the researchers. In addition, a heuristic algorithm for clustering editing problems on weighted networks is presented to identify research communities from the formulated common-knowledge network. Furthermore, to illustrate the synergy and interaction among researchers in identified communities, several metrics or topics, such as collaboration and publication count, dominating research topic, top influential authors, and affiliated departments, are presented in this study, corresponding to each research community.

6.1.3 Line-Item Category Identification for Short-Text Classification

A process of identifying line-item categories (classifying labels) for short-text classification based on keyword (co-exiting) networks is outlined in Chapter 4. A Cograph Editing-based heuristic clustering algorithm using an integer linear programming formulation is proposed to identify keyword clusters from large-scale general-weighted networks. Furthermore, an application of the proposed framework to identify and classify invoices based on line-item categories is presented in this chapter.

6.1.4 A Hybrid Fraud Detection Framework Using Augmented AI

A hybrid framework for identifying fraudulent users on invoicing platforms where human review is required in the final decision-making process is presented in Chapter 5. This framework uses a combination of unsupervised clustering, supervised machine learning processes and a small amount of labelled data to identify fraud risk cluster(s) in the model training process. In addition, a weighted center for the fraud risk cluster based on the feature importance score is presented and used in the red-flag prioritization and augmented AI processes. Finally, a practical implementation of the hybrid framework in a weekly segmented structure is discussed in this chapter.

6.2 Recommendations for Future Research Directions

The research in this thesis can be extended in several ways related to the scalability of data clustering frameworks.

6.2.1 Correlation Clustering Problem: Scalability and Benchmarking against Machine Learning Algorithms

The Correlation Clustering problem, discussed in Chapter 2, is NP-hard and faces scalability issues with identifying clusters in large-scale real-world networks. Developing efficient solution algorithms for large-scale networks is a promising area for future research in the Correlation Clustering problem area. Recent developments using parallel structured algorithms for Correlation Clustering provide $(1 + \epsilon)$ approximation guarantees on complete networks. The future research directions of designing efficient parallel algorithms for the Correlation Clustering problem on the general-weighted networks with a theoretical guarantee are still open. One other possible research direction is to consider optimization decomposition methods such as Benders' decomposition Lukasik et al. (2020), and possibly with parallelization Keuper et al. (2019). Another possible direction could be designing an efficient algorithm for large-scale Correlation Clustering instances that use the problem's inherent structural properties. Finally, this thesis presents a little-explored but significant scope for future research, comparing the Correlation Clustering with well-known machine learning algorithms.

In machine learning, agnostic learning aims to find the best hypothesis for the target function from a given hypothesis class that contains the node clusters. Correlation Clustering can be viewed as a type of agnostic learning, where the link labels are the examples (either positive or negative), and it is only allowed to use partitioning as the hypothesis for the target function. Therefore, the question of comparison between the Correlation Clustering with other well-known machine learning algorithms should arise naturally. In recent years, very few discussions, such as Pozzi et al. (2005); Bressan et al. (2019), have been done in this area. Future research should investigate the performance comparison of Correlation Clustering with different machine learning algorithms.

6.2.2 Impacts of Common-Knowledge Network-based Research Communities

The proposed Common-Knowledge Network-based communities in Chapter 3 can be used to analyze research trends, collaboration, and the impact of strategic research investments on interdisciplinary research productively over time as well as perform inter-university comparative studies. Our proposed Common-Knowledge Network-based approach uses only publication data collected from Web of Science. One promising line of future research is combining Web of Science data with ORCiD, and a survey or interview of authors to support and validate the proposed hypotheses and incorporate other relevant factors such as research grants, infrastructure and graduate student activities.

6.2.3 Keyword Relative Position and Short-Text Ontology

The proposed user-generated short-text classification framework, in Chapter 4, utilized a keyword network-based clustering to identify line-item categories. The keyword formulation step used process keywords obtained from the short-texts regardless of their corresponding positions in the actual text. Adding additional features related to keywords' relative positions in the keyword network may be useful in the line-item identification process. Furthermore, studies showed that clustering-based algorithms could leverage knowledge from documents' taxonomic ontology. In future, besides the keywords clusters, another level of filtering (or features selection) can be added to this framework by using the given short texts ontology. A promising area of application is social media platforms such as Twitter or Facebook.

6.2.4 Extension of the Proposed Hybrid Fraud Detection Framework's Layers

The Hybrid Fraud Detection Framework framework, presented in Chapter 5, uses a small amount of labelled data in the fraud risk cluster(s) identification process. In future, the FRC identification step can be eliminated by a semi-supervised learning process to train the next-level supervised classifier. In addition, this framework also uses an augmented AI process in the final decision-making process, which opens the possibility of using a reinforcement learning layer to give feedback from human decisions.

Chapter References

Bressan, M., Cesa-Bianchi, N., Paudice, A. and Vitale, F. (2019). Correlation clustering with adaptive similarity queries, *Advances in Neural Information Processing Systems* **32**.

Keuper, M., Lukasik, J., Singh, M. and Yarkony, J. (2019). Massively parallel benders decomposition for correlation clustering, *arXiv preprint* .

Lukasik, J., Keuper, M., Singh, M. and Yarkony, J. (2020). A benders decomposition approach to correlation clustering, *2020 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments (MLHPC) and Workshop on Artificial Intelligence and Machine Learning for Scientific Applications (AI4S)*, IEEE, pp. 9–16.

Pozzi, S., Zoppis, I. and Mauri, G. (2005). Combinatorial and machine learning approaches in clustering microarray data, *Biological and Artificial Intelligence Environments*, Springer, pp. 63–71.