A FRAMEWORK FOR DYNAMIC HETEROGENEOUS INFORMATION NETWORKS CHANGE DISCOVERY BASED ON KNOWLEDGE ENGINEERING AND DATA MINING METHODS

SAFWAN UMER

School of Science, Engineering and Environment

University of Salford, Manchester, UK

Submitted in Partial Fulfilment of the Requirements of the Degree of

Doctor of Philosophy

2021

Table of Contents

List of Figures

Abstract

Information Networks are collections of data structures that are used to model interactions in social and living phenomena. They can be either homogeneous or heterogeneous and static or dynamic depending upon the type and nature of relations between the network entities. Static, homogeneous and heterogenous networks have been widely studied in data mining but recently, there has been renewed interest in dynamic heterogeneous information networks (DHIN) analysis because the rich temporal, structural and semantic information is hidden in this kind of network. The heterogeneity and dynamicity of the real-time networks offer plenty of prospects as well as a lot of challenges for data mining. There has been substantial research undertaken on the exploration of entities and their link identification in heterogeneous networks. However, the work on the formal construction and change mining of heterogeneous information networks is still infant due to its complex structure and rich semantics. Researchers have used clusters-based methods and frequent pattern-mining techniques in the past for change discovery in dynamic heterogeneous networks. These methods only work on small datasets, only provide the structural change discovery and fail to consider the quick and parallel process on big data. The problem with these methods is also that cluster-based approaches provide the structural changes while the pattern-mining provide semantic characteristics of changes in a dynamic network. Another interesting but challenging problem that has not been considered by past studies is to extract knowledge from these semantically richer networks based on the user-specific constraint.

This study aims to develop a new change mining system ChaMining to investigate dynamic heterogeneous network data, using knowledge engineering with semantic web technologies and data mining to overcome the problems of previous techniques, this system and approach are important in academia as well as real-life applications to support decision-making based on temporal network data patterns. This research has designed a novel framework "ChaMining" (i) to find relational patterns in dynamic networks locally and globally by employing domain ontologies (ii) extract knowledge from these semantically richer networks based on the user-specific (meta-paths) constraints (iii) Cluster the relational data patterns based on structural properties of nodes in the dynamic network (iv) Develop a hybrid approach using knowledge engineering, temporal rule mining and clustering to detect changes in the dynamic heterogeneous networks.

The evidence is presented in this research shows that the proposed framework and methods work very efficiently on the benchmark big dynamic heterogeneous datasets. The empirical results can contribute to a better understanding of the rich semantics of DHIN and how to mine them using the proposed hybrid approach. The proposed framework has been evaluated with the previous six dynamic change detection algorithms or frameworks and it performs very well to detect microscopic as well as macroscopic human-understandable changes. The number of change patterns extracted in this approach was higher than the previous approaches which help to reduce the information loss.

Keywords: Networks; Ontologies; Static Data Mining; Clustering; Frequent Pattern Mining; Dynamic Heterogeneous Information Networks;

Acknowledgements

I am immensely thankful to my supervisor, Prof. Mo Saraee, who has guided me patiently throughout the final process of my PhD journey. May all the glory be to God that Prof. Mo Saraee took over the project at the time he did, and I acknowledge the opportunity he has given to me to complete this thesis under his short period of guidance. I would have never been able to finish my theses without his invaluable support, encouragement, and important suggestions.

I would also like to express my great appreciation to Prof. Farid Meziane for providing me with fruitful guidance for my PhD. I also acknowledge the groundwork and proofreading made by him. I would have never been able to complete my research without his guidance, corrections and help at all stages of my PhD. research.

I express my thanks to my beloved wife, Asia, and my child Urwa Umer for their love, and for always being a source of inspiration for me during this time of my PhD studies. Finally, I am also thankful to my parents, family and especially my elder brother Shabbir Ahmad who were always supporting me morally and encouraging me with best wishes.

Declaration

I, Safwan Umer declare that this Thesis is the result of my independent work. No part of this research work has been submitted to any other institution for another degree. Moreover, I have acknowledged, where appropriate the content of the others work in my research.

Chapter 1 Introduction

1.1 Introduction

The area of heterogenous network analysis is not confined to computer science; it has grown extensive attention from researchers in transportation, neuroscience, biology, physics and various other scientific and social disciplines. Most real-life systems comprise a hefty amount of interrelating, multi-typed objects which connect based on some relationship between them and we can call them information networks without loss of generalization(C. Shi, Li, Zhang, Sun, & Philip, 2016). The concepts of Heterogeneous Information Networks (HINs) have been introduced by Sun, Han, et al. dating back to 2009(Y. Sun, Han, et al., 2009). It has been a decade that HINs analysis has become a very hot topic in the fields of databases, information extraction and data mining.

Heterogenous information networks also have temporal data and we construct dynamic heterogeneous information networks (DHINs) considering the effect of the time factor (X. Cao, Zheng, Shi, Li, & Wu, 2016). Before the introduction of dynamic heterogeneous information networks, static and dynamic networks structures were employed as a common model to show the associations among different entities, their temporal evolution and evolving aspects of the data. Researchers have made valuable efforts in the development of different techniques to examine and extract network change information from static data networks. The studies by (Asai, Abe, Kawasoe, Sakamoto, & Arikawa, 2001; Dillon, 1983; Inokuchi, Washio, & Motoda, 2000a; Zaki, 2002) and (Brin & Page, 2012) used different approaches to extract and analyze information from static networks. Furthermore, (Berkhin, 2006) has given a list of various clustering techniques for static networks analysis. In comparison to static networks, the dynamic networks consist of the temporal snapshot of the network nodes, and each snapshot represents a logical timestamp of the correlation between single-type nodes of the network system.

The dynamic network change analysis has less developed methods available to extract change point information from dynamic networks compared to static networks. Nonetheless, a few pieces of research have been made in this field recently using graph theory to find information patterns by (Borgwardt, Kriegel, & Wackersreuther, 2006); to distinguish network evolution rules by (Berlingerio, Bonchi, Bringmann, & Gionis, 2009); clustering graphs by (Chakrabarti, Kumar, & Tomkins, 2006); searching subgraph subsequences by (Inokuchi & Washio, 2010), and finding related cliques by (Cerf, Nguyen, & Boulicaut, 2009) and (Robardet, 2009) in dynamic data networks. Earlier change detection techniques in the dynamic networks have employed graph theory(Shoubridge, Kraetzl, Wallis, & Bunke, 2002; Showbridge, Kraetzl, & Ray, 1999), statistical procedures(Bunke, Münger, & Jiang, 1999; Jiang, Munger, & Bunke, 2001), and pattern mining approaches(Messmer & Bunke, 1998; Wallis, Shoubridge, Kraetz, & Ray, 2001) to detect changes in dynamic networks. Network Change discovery has received the most attention in recent years and can be divided into three groups supervised, unsupervised and rule-based (pattern recognition) methods.

The supervised network change discovery methods use the extracted feature and historical information of the dynamic network to understand the change dynamics of the network(Ze Li, Sun, Zhu, & Lin, 2017). Although different classification methods(Ze Li et al., 2017; Sliva, Subrahmanian, Martinez, & Simari, 2009; Subrahmanian, Mannes, Sliva, Shakarian, & Dickerson, 2012; Xue, Wang, & Zhang, 2011) have been exercised to detect changes in dynamic networks, these methods usually ignore the heterogeneity of the DHINs. In contrast, unsupervised methods require no prior information about the network to detect changes. Cluster-based techniques are an important unsupervised method that has been very significant to detect changes in different types of dynamic networks. In various scientific fields, data comes from heterogeneous sources directly or indirectly, clustering received great attention due to its ability to explore information by links between the nodes in the network. Since most of the real-world complex networks are dynamic, it becomes more challenging to explore these data structures. The most common type of dynamic networks is mobile communication data, scientific research publication data, and data on human communication via the Internet (Hartmann, Kappes, & Wagner, 2014). To analyze dynamic network graphs, the cluster-based solution has been used, which depends on the recognition of the variations in the characteristics features of the network globally. These solutions work on the natural summary for recognizing the network configuration and the natural modifications during the growth process using the graphs of the network states (Aggarwal & Wang, 2010). Aggarwal and Philip (2005) have used the clustering-based method in a social network context to discover changes. Moreover, these methods have been used by (J. Sun, Faloutsos, Papadimitriou, & Yu, 2007) in discovering communities using clusters, (J. Sun, Tao, & Faloutsos, 2006) and (Tong, Papadimitriou, Sun, Yu, & Faloutsos, 2008) in large static and dynamic graphs, (Berlingerio, Coscia, Giannotti, Monreale, & Pedreschi, 2013; Ferlez, Faloutsos, Leskovec, Mladenic, & Grobelnik, 2008) tensor analysis, (Desikan & Srivastava, 2006) and (R. Ahmed & Karypis, 2012) have used graph-based clustering to detect changes in the network structures.

The rule-based system in DHIN employed inductive logic to make the connection between objects and detect changes. Loglisci et. al. (2015) used a rule-based method to detect changes employing extracted patterns from small temporal snapshots of the single network. The proposed method faces a classification problem hence not suitable for large and multiple DHINs change detection.

Change discovery in DHINs is very important because, in reality, changes in networks and their behaviours translate to substantial events for the organization using these networks (Ze Li et al., 2017). Common types of changes in DHINs include gradual change over time, recurring, abrupt, local (microscopic), global (macroscopic), and community-level changes. Detection of these changes help to answer the questions like; how does the network evolve? and how can we find the suspicious activities in a network (Yu Wang, Chakrabarti, Sivakoff, & Parthasarathy, 2017)?

The dynamic networks used in the past have only considered single typed objects and their relationships, but in real network systems, the objects are multi-typed, heterogeneous and have complex schemas. For example, the most famous dynamic multi-type heterogeneous networks are social media websites (e.g., Facebook[1], Twitter[2] and LinkedIn[3]), citation networks Digital Bibliography & Library Project (DBLP[4]), protein-protein connection network systems and email systems (R. Ahmed & Karypis, 2012). Moreover, in real-life social media networks, there is a continuous evolution cycle. Nodes in these networks connect with new nodes and evolve with time. Similarly, new users connect with the network and old users create new relationships with existing users, they also create and like pages, posts and new multimedia items such as videos, infographics texts which make the network interactions dynamic and heterogeneous (Jure Leskovec, Kleinberg, & Faloutsos, 2005).

DHINs are dynamic, multi-typed and heterogeneous. In this type of network, the multi-typed objects and their links carry different semantic meanings. The literature reviewed in this study has revealed that none of the studies employed semantic web ontologies to detect changes in DHINs. This research explores the potential of using knowledge engineering processes(Kendal

[1] https://www.facebook.com/
[2] https://twitter.com/
[3] https://www.linkedin.com/
[4] https://dblp.org/

& Creen, 2007) in DHINs. A novel framework is proposed that embeds the abstract knowledge-base which support the designing and generation of knowledge graphs after interpreting the specified dynamic heterogeneous network data. This thesis proposes a new DHINs semantics-based changes mining system framework called “ChaMining” using knowledge-engineering and data mining methods. The proposed framework has been developed based on knowledge engineering and data mining methods to detect and visualize local, global and community-level changes in DHINs. The thesis also presents an empirical evaluation of the developed system. According to the best of the author’s knowledge, there is no known system that can detect changes in DHINs using knowledge engineering and data mining methods.

1.2 Motivation and Challenges

In real systems, network data is ubiquitous, multi-typed, complex, dynamic, heterogeneous and asymmetrical. As the real network data become more complex, heterogeneous and temporal, we need to design more powerful and dynamic heterogeneous networks, which provide more challenges for data mining methods(C. Shi & Philip, 2017). Many types of networked data, for example, Resource Description Framework (RDF) data cannot be easily modelled with DHINs, as the RDF data is schema-rich, it becomes difficult to detect changes with various meta-paths and multiple objects with interconnected relations(J. Gao et al., 2010; Chenguang Wang, Sun, et al., 2016). The studies of dynamic network change detection have mostly considered that the network is single typed and neglects the fact of heterogeneity and schema-richness. Rule-based and supervised approaches to network analysis have mostly examined graph-based methods to detect changes by attachment and removal of the (edges) nodes in the network. In comparison, cluster-based methods give the changes of entities and pattern-based methods give the characterization of changes in the dynamic network. The dynamic and ubiquitous networks require a novel class of knowledge engineering processes and data mining techniques to solve the problems in the current approaches because graph theory explores the structural characteristics in the homogenous networks and shows less effectiveness to the dynamic and heterogeneous networks data which evolve with time (Loglisci, Ceci, & Malerba, 2015).

The goal of this study is to devise a new methodology to construct DHINs, extract association-based patterns from real-time temporal data, cluster these relational patterns and detect local, global and community level changes using these clusters of relational data. This research will lead to the following broad questions that need to be addressed:

1. How well do existing algorithms or frameworks for dynamic network change detection perform?
2. Can the development of DHINs employing a novel framework embedded with knowledge engineering processes through Web Ontology Language (OWL[5]) and data mining methods work best to detect stable and change patterns in DHINs?
3. How does the performance of the developed framework "ChaMining" compare with other existing methods and frameworks?

1.3 Research Aim and Objectives

This study aims to develop a framework using knowledge engineering and data mining methods to construct dynamic heterogeneous information networks and detect, local, global and community-level changes in DHINs. Given this aim, the research objectives and main research question are given below:

1. To review the current frameworks and algorithmic methods for change discovery of dynamic heterogeneous information networks.
2. To propose a novel framework that can construct DHINs containing multiple objects, rich semantics and relationships.
3. To explore and assess the use of knowledge engineering processes and data mining methods with existing methods.
4. To detect local, global and community-level changes in DHINs using the developed new framework.
5. To evaluate the framework by applying it on domain-specific DHINs data and determine the outcomes in terms of system accuracy concerning existing approaches for change discovery.

The main research question of this research is that, can the use of knowledge engineering processes help to construct dynamic heterogeneous information networks' knowledge-base which can be employed to detect changes in the temporal, multi-typed heterogenous knowledge graphs by applying data mining methods?

[5] https://www.w3.org/OWL/

1.4 Research Methodology

The purpose of the research is to discover knowledge and answers to the questions using scientific methods. Kothari (Kothari, 2004) has divided the research into eight basic types these include descriptive, analytical, applied, pure, quantitative, qualitative, conceptual and experimental. The comparative summaries of these research types are as follows:

1.4.1 Descriptive and analytical research comparison.

Descriptive research explains the characteristics of the data or phenomenon being studied. Descriptive research includes comparative and correlation methods, research questions-based all types of surveys and fact-finding investigations of different types. This approach can be used in various ways and it is widely employed in the field of social and business management studies. In this method, the researcher has no control over the variables, and he can only use this method on the current and past happenings. The researchers use this approach to analyse such items as, for instance, shopping frequency, which Box Office movie has the highest number of viewers, likings of people, or similar facts and figures. In Analytical research, on the contrary, the researchers make critical evaluations based on the facts or information already available(Kothari, 2004).

1.4.2 Applied and Pure research comparison.

Applied research aims to find solutions for a pressing practical problem, while fundamental (pure) research is concerned with the formulation of a theory. Applied research is designed to solve immediate problems facing a society or a business organisation. The knowledge obtained from applied research has a certain objective in the form of a product, process or package. Marketing research and research to recognize political, economic or social trends that may affect a specific institution are some examples of applied research. Collecting knowledge for knowledge’s sake and research regarding some natural phenomena are examples of fundamental research. Therefore, the main objective of applied research is to solve an immediate problem, on the other hand, fundamental research works on the formulation of the theories, which have wide-ranging applications and contribute to the existing body of knowledge(Kothari, 2004).

1.4.3 Quantitative and qualitative research comparison.

Research can also be either quantitative research or qualitative research. The former research method is based on measurement and deals with numbers and calculations. Quantitative research also empirically investigates a quantitative measurement using statistical methods. In comparison, qualitative research is concerned with the phenomenon related to use in-depth interviews aims at a qualitative phenomenon to determine the underlying motivations and desires(Kothari, 2004).

1.4.4 Conceptual and Empirical research comparison.

Empirical or experimental research is data-dependent research where hypotheses are developed to be verified by experiment. In this research, the experimenter has control over the selection of the variables being used in the research. There are two stages in empirical research, data collection and experiment design stage. Data in experiment research is gathered from its source and the researcher is required to have a working hypothesis to test. The experimenter has the responsibility to select data in favour of or against their hypothesis. In contrast to the empirical research method, conceptual research which is popular among philosophers and thinkers is based on theories and abstract ideas(Kothari, 2004).

Apart from the above-mentioned research methodologies researchers have also used scientific methodology and experimental methodology to research in the field of system and algorithm development(Franklin, 1971; Mitchell, 2006). The scientific methodology by employing statistical methods and experimental researches also help to test a hypothesis and validate the outcomes of a research experiment. This research belongs to the fields of data mining and knowledge engineering and most researchers have used experimental or empirical, research methodologies in these fields(Franklin, 1971; Mitchell, 2006). Hence, this study will use the empirical methodology to address the research questions and objectives. The following research steps will be adopted in this research to achieve the desired goals of this study:

1. Performing a comprehensive literature review on the present frameworks and algorithms to solve the problem.
2. Reviewing ontologies from the perspective of data mining methods, which will help to understand how ontologies are supporting data mining methods to perform better.

3. Constructing relations from structured, unstructured, and semi-structured dynamic heterogeneous network data to develop knowledge-base.
4. Surveying similarity measuring methods can be used to find the changes in the relational patterns and clusters.
5. Developing a framework based on knowledge engineering and data mining methods.
6. Using the similarity measuring methods in step-4 to find stable and change patterns in the temporal knowledge graphs.
7. Testing the outcomes of the research questions is based on using knowledge engineering and data mining methods to develop a system to detect local, global and community-level changes in dynamic heterogeneous information networks.
8. Evaluating the system which has been developed on real dynamic heterogeneous information networks for example DBLP[6], Internet Movies Database (IMDB[7]), Enron-Email[8] and High-energy physics data[9] .

1.5 Main Contributions

This research is versatile and combines theoretical understanding from computer science and data can be employed from social media, computer networks, biomedical or any other domain which holds dynamic heterogenous information graph structures. This thesis presents a framework embedded with knowledge-engineering processes and data mining methods to facilitate the effective local, global and community level change discovery of dynamic heterogeneous information networks. The proposed approach addresses various limitations of the existing methods and produces outstanding outcomes. The core contributions of this research could be summarised as below:

1.5.1 Proposed novel DHINs change mining framework:

This research proposes a general framework for DHINs construction. This framework consists of certain core components which have relationships with other components. It provides machine-centric change detections without losing the underlying DHINs semantics of the knowledge-base, based on the idea of graph classification theory. The proposed framework is

[6] https://dblp.org/
[7] https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset
[8] https://www.cs.cmu.edu/~./enron/
[9] https://snap.stanford.edu/data/cit-HepPh.html

developed based on knowledge engineering and data mining concepts to detect local, global and community-level changes.

1.5.2 Meta-paths based temporal knowledge graphs:

Another main contribution of this research is to implement an algorithm to map DHINs data to the temporal knowledge graphs on user-specific meta-paths and predicate logic.

1.5.3 A hybrid change mining methodology:

Once we have the knowledge base of DHINs then the proposed approach divides the knowledge graph into a sequence of temporal graphs based on meta-paths and nodes of the knowledge graph. This division of knowledge graph into reduced graphs will help to detect changes at the local as well as community level.

1.5.4 Application of framework on different DHINs.

The proposed approach and framework have been implemented on four different benchmark dynamic heterogeneous information networks DBLP, IMDB, Enron Email and High-energy physics citation.

1.6 Thesis outline

An overview of the thesis structure is presented in Figure 1.1. which shows each chapter and its link with other chapters. Dotted arrows show a weak link and solid arrows represents a strong link between two chapters.

Chapter 2. Heterogeneous Information Networks Background: This chapter starts by introducing HINs basic definitions and gives an in-depth overview of different data mining methods to analyse HINs.

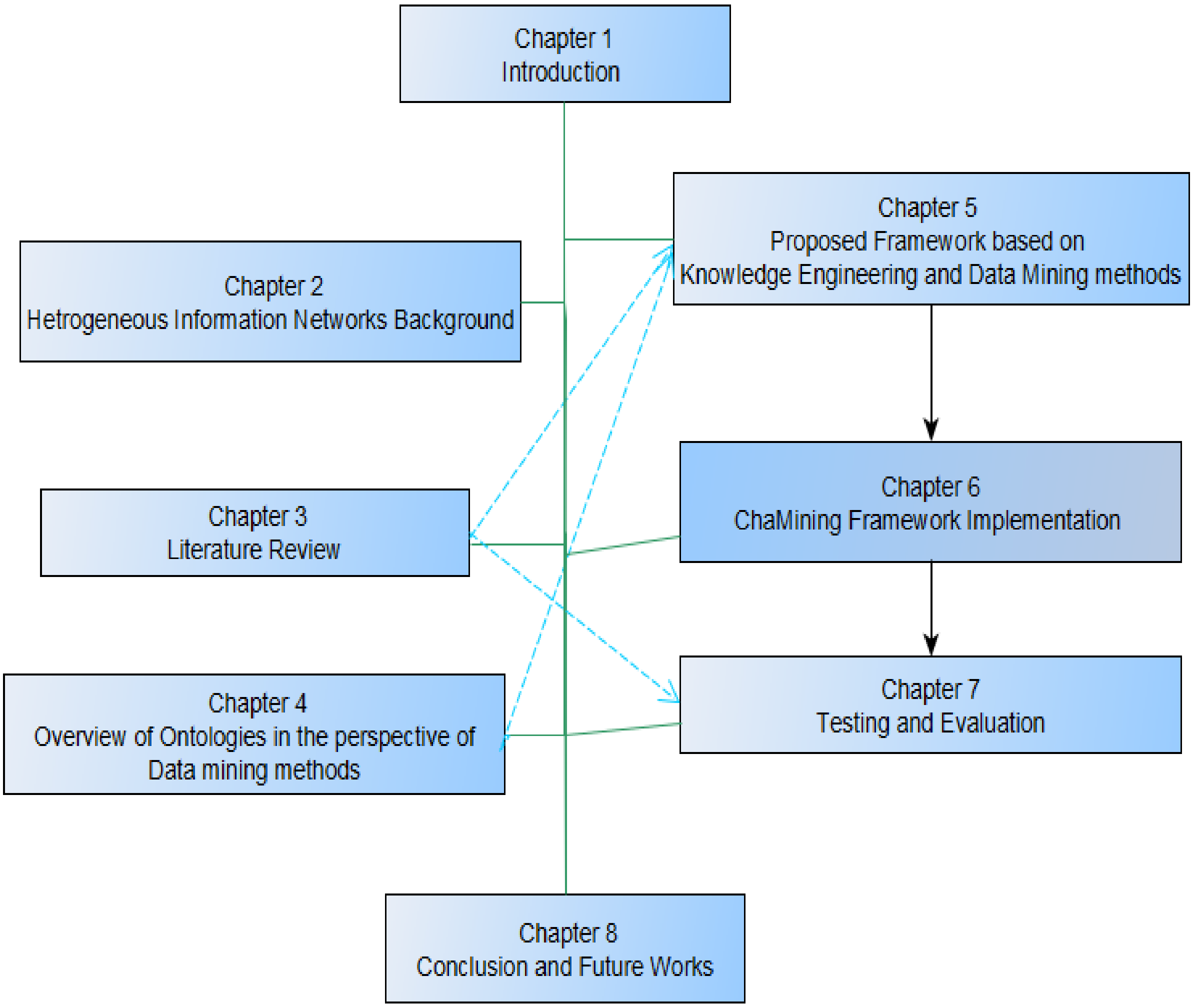

Chapter 1
Introduction
Chapter 2
Hetrogeneous Information Networks Background
Chapter 3
Literature Review
Chapter 4
Overview of Ontologies in the perspective of Data mining methods
Chapter 5
Proposed Framework based on Knowledge Engineering and Data Mining methods
Chapter 6
ChaMining Framework Implementation
Chapter 7
Testing and Evaluation
Chapter 8
Conclusion and Future Works

Figure 1 overview of the thesis structure

Chapter 3 Literature Review: Particularly, this Chapter focuses on dynamic heterogeneous information networks. It elaborates on the difference between dynamic and DHIN networks and network schema. This chapter also explains why we do change detection. After a brief history of different types of networks and their applications, details of different types of change detection frameworks and algorithms are reviewed, and their pros and cons are discussed.

Chapter 4 An overview of Ontologies from the perspective of data mining: This Chapter introduces an overview of Semantic Web architecture. It explores what are the terminologies that can be used to develop an ontology. It also outlines the significance of ontologies in the data mining field.

Chapter 5 A new framework for DHIN Change detection: This Chapter introduce the proposed framework. This Chapter presents the different components of the framework and their relationships with other components. Some relevant examples are also presented to explain the concepts.

Chapter 6 ChaMining Framework Implementation: This Chapter outlines the details on the implementation of the framework.

Chapter 7 Testing and Evaluation: This Chapter presents the comparisons of the present methods and algorithms with the method developed through this study.

Chapter 8 Conclusions and Future Research: This Chapter contains the concluding remarks and discusses the possible future directions for this research.

Chapter 2 Background Heterogeneous Network Analysis

2.1 Introduction

As described in the introduction chapter, this research discovers the use of semantic web technologies and data mining techniques for developing a new framework for change discovery in dynamic heterogeneous networks. This Chapter presents the details on the background of heterogeneous information networks (HINs) analysis and data mining techniques for HINs analysis. The objective of this Chapter is to introduce the conceptual overview of HINs and different kinds of heterogeneous networks. The literature reviewed is rationally organised into the supervised, and unsupervised learning categories that support an understanding of their data mining tasks. Section 2.1 introduces the background of heterogeneous information networks, briefly explain basic concepts in this field, compare the heterogeneous information network with other related concepts, and give some HIN examples. This section also presents a detailed background of the heterogeneous information networks, basic definitions in the building blocks of HIN. Section 2.2 gives an overview of supervised data mining techniques to analyse HINs. Section 2.3 describe unsupervised data mining methods to classify unlabelled HINs data. Finally, section 2.4 gives a brief overview of other data mining tasks to analyse HINs.

2.2 Preliminaries

In this section, basic definitions and preliminaries of the background knowledge including the concepts of different types of information networks, meta-path and dynamic change mining are introduced.

2.2.1 Heterogeneous vs homogeneous networks

Information network represents an abstract picture of the objects and the relations among these objects (Y. Sun & Han, 2013). The Heterogeneous Information Network is defined by a function $G = (V, E, W; A, \psi)$ where:

- V is the Nodes set, $E \subseteq V \times V$ is the nodes links set
- $W: E \rightarrow R +$ is a defined weight function
- A is an alphabetical table denoting names of different types of vertices,
- $\psi: V \rightarrow A$ is the mapping from each vertex to its type

If the number of types |A| > 1, G is called a heterogeneous information network; otherwise, G is a homogeneous information network. For example, Figure 2.1 shows the HIN instance in the domain of Cyber Security. This HIN consists of clients, domains, IP addresses and there can be six types of relations as shown in Figure 2. The relation between the HIN nodes can be of six types client-query-domain, client-segment-client, domain-resolve-IP, IP-domain-IP, domain-cname-domain, and domain-similar-domain. HINs contains multiple types of nodes and links, it is a very powerful information modelling method. For example, Figure 2.3 part (c) presents a HIN based bibliographic information network containing five types of entities. For each paper, it has a set of links to authors, venue organization and terms.

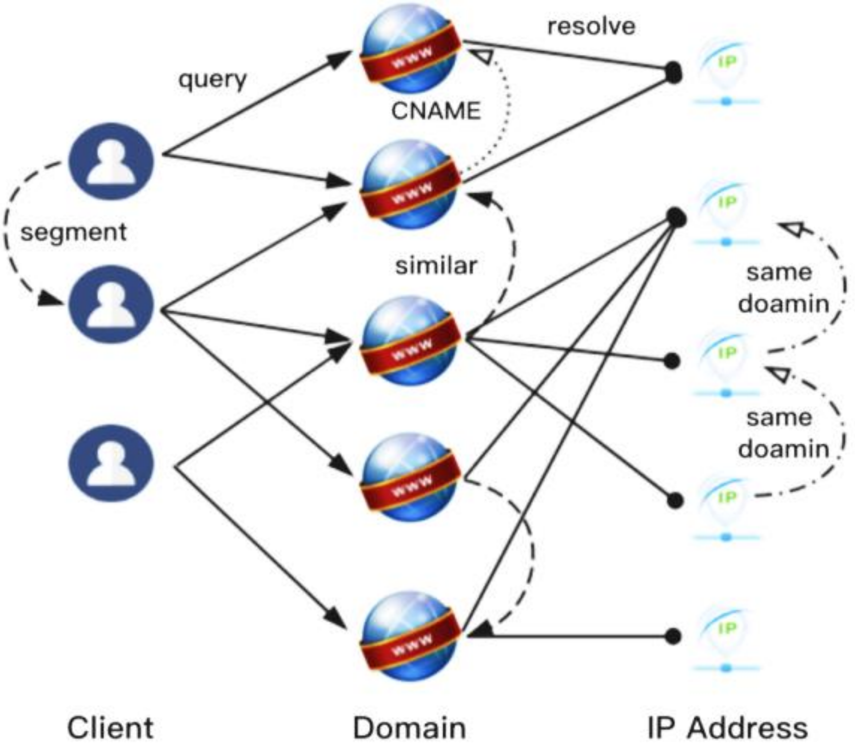


Figure 2 Heterogeneous information network instance in Cyber Security domain(X. Sun, Tong, Yang, Xinran, & Heng, 2019)

2.2.2 HIN Network Schema

To understand the network object and link types better in a ubiquitous heterogeneous network it is mandatory to provide network schema. A network schema(Y. Sun & Han, 2013) of a given information network G= (V, E), is defined by a function T_G = (A, R) and it also has the following functions associated with it:

- Object type mapping function φ: V→A
- Link-type mapping ψ: E→ R, which is a directed graph defined over object types A, with edges as relations from R

Figure 3 gives the network schema of the HIN network presented in Figure 2.

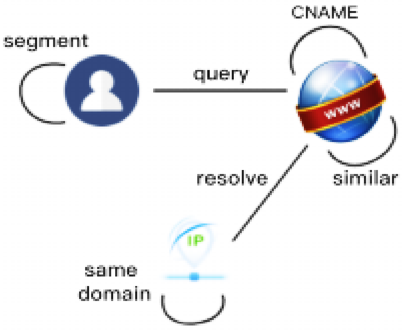


Figure 3 **Network schema of the Cyber Security HIN (X. Sun et al., 2019)**

2.2.3 Meta-path

The meta-structure of the network is also necessary to understand the relations between network nodes. Therefore the concept of network meta-path is proposed to describe the relational meta structure of a network. Meta-path is a path defined on the network schema T_G = (A, R) graph, and a meta-path P using a relation composition operator ◦ is used to define composite relationship R between objects A_1….. $A_{L+1.}$ P is presented as A_1 A_2 A_3 ….. A_{L+1} Where L is the length of the meta-path(X. Sun et al., 2019). For example, Figure 4 parts (d) and (e) elaborate the concepts of meta-paths and meta-graphs respectively for bibliographic HIN(Y. Sun, Han, Yan, Yu, & Wu, 2011).

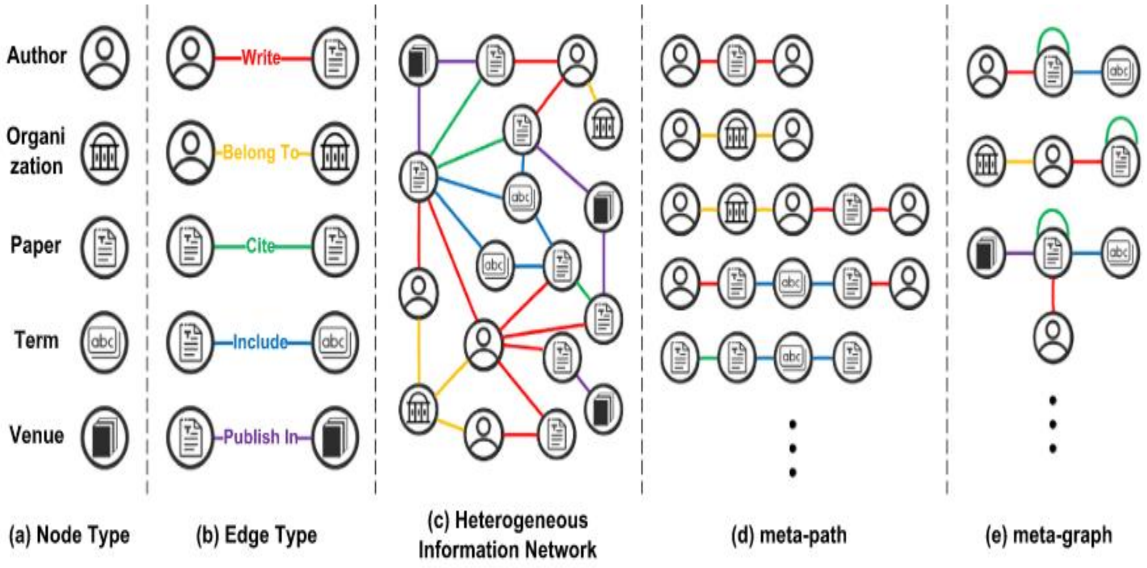


Figure 4 DBLP example of Heterogeneous information networks with five types of nodes author, paper, venue, organization, and the term(X. Gao, Chen, Zhan, & Yang, 2020).

2.2.4 Multi-relational Network

A network with V set of vertices and E set of edges is called multi-relational M=(V, E) if the vertices and edges have a relationship function defined as $E_k \subseteq$. Form example kinship and co-start networks are types of multi-relational networks. This type of network has been used in many fields from cognitive, social science and artificial intelligence(Bollen, Rodriguez, Van de Sompel, Balakireva, & Hagberg, 2007; Rodriguez, Bollen, & Van de Sompel, 2007; Sowa, 2014; Wasserman & Faust, 1994). These types of networks are the core data structure of the emerging WWW data(T Berners-Lee, J Hendler, & O Lassila, 2001). These networks can be used to construct very complex systems than a single relation network. The problem with these types of networks is that there are very few algorithms to analyse these types of multi-relational networks and most of the researchers have used single relational networks to develop algorithms(Rodriguez & Shinavier, 2010). Recently, Victor et. Al. (StröEle, ZimbrãO, & Souza, 2013) used this method to predict links using link analysis.

2.2.5 Multi-dimensional Network

The concept of multi-dimensional networks was introduced by Tang et al.(Tang, Liu, Zhang, & Nazeri, 2008) This is a very significant type of multilayer network. These are networks with multiple kinds of relations. This is a special case of heterogeneous networks and these networks have only one type of object and more than one type of relationship between objects.

2.3 Data mining methods for heterogenous network analysis

Data mining is a method to turn raw data into useful information(Han, Pei, & Kamber, 2011). This section overviews supervised and unsupervised data mining methods to analyse heterogeneous information networks.

2.3.1 Supervised learning methods

Supervised learning is a machine learning technique. This technique allows us to collect and produce a data output from previous experience or examples. The support vector machines (T Berners-Lee et al., 2001), linear regression (Montgomery, Peck, & Vining, 2012), logistic regression (Kleinbaum, Dietz, Gail, Klein, & Klein, 2002), Naive Bayes (Rish, 2001), linear discriminant analysis (Izenman, 2013), decision trees (Rokach & Maimon, 2005), k-nearest neighbour algorithm (Kataria & Singh, 2013), neural networks (Hramov et al., 2018) similarity learning (D. Chen, Yuan, Chen, & Zheng, 2016) are most widely used supervised algorithms or methods.

2.3.2.1 Similarity measure based HIN analysis

Similarity measure based HIN is a supervised machine learning technique in artificial intelligence. This technique is closely related to regression and classification, but the primary objective is to evaluate the similarity of objects. This technique is the basis of many data mining tasks, such as web search, clustering, classification and product recommendation (C. Shi et al., 2016). The feature-based approaches and link-based approaches can be used to measure the similarity of objects based on their feature values using the following similarity measures:

- Cosine Similarity Measure(Broumi & Smarandache, 2014)
- Jaccard coefficient (Ivchenko & Honov, 1998)

- Euclidean distance(L. Wang, Zhang, & Feng, 2005)
- Minkowski distance (Merigó & Gil-Lafuente, 2008)
- Manhattan distance(Pandit & Gupta, 2011)

Similarity measure on heterogeneous information networks considers structure similarity of two objects but also takes the meta path linking these two objects into account. For example, two different meta paths link two objects, and these meta paths contain different semantic meanings, which may lead to different similarities. Hence, the similarity measure of HIN is a meta path constraint(C. Shi & Philip, 2017). PathSim(Y. Sun et al., 2011) is the first measure introduced that evaluates the similarity of same-typed objects based on symmetric paths. Baily et al. (J. He, Bailey, & Zhang, 2014) and Hou et al. (Yao & Mak, 2014)extended the PathSim and included the transitive similarity, temporal dynamics and supportive attributes which adds richer information about objects. Xiong, Zhu and Phillip (Xiong, Zhu, & Philip, 2014) proposed a path-based similarity join method that returns the top k similar pairs of objects based on user-specified join paths. Relation similarity search in schema-rich heterogeneous information networks (RelSim) measure proposed (Chenguang Wang, Sun, et al., 2016). In the information retrieval community, Lao and Cohen (Lao & Cohen, 2010) propose a Path Constrained Random Walk (PCRW) model to measure the entity proximity in a labelled directed graph constructed by the rich metadata of scientific literature. Since, existing meta-structure constructed similarity measures only use one meta-structure and lead to loss of accuracy Li and Wang(Zhaochen Li & Wang, 2018) propose a weighted method to challenge this issue. Yang et al. (C. Yang et al., 2018) introduced weighted heterogeneous information networks (WHIN) to develop attractive recommendations which attach attribute values to nodes in the network. Zhou et al. (Y. Zhou et al., 2018) presented stratified meta structure-based similarity measure to automatically capture rich semantics of HINs. Current important researches on similarity measures in HIN analysis are HeteRank (M. Zhang, Wang, & Wang, 2018), RecurMS (Y. Zhou et al., 2019), GraphSim (X. Chen, Jiang, Wu, Wei, & Lu, 2020), and HowSim(Yue Wang et al., 2020).

2.3.2.2 HIN analysis employing Classification

Classification is one of the data mining techniques where a model or classifier is designed to predict class labels. The independent identically distribution (Jacobs, 1992) and link-based objects classification (Getoor, 2005) are different techniques of classification. However, the link-based object classification has received considerable attention due to links that exist among objects in many real-world datasets. The link-based classification in HIN can classify multiple types of objects simultaneously and label knowledge can evolve through various links among different types of objects. For example, Actors, movies, Venue and types of Movies are different types of objects that are interconnected by multitype links.

A very significant approach in the field of machine learning and data mining is classification(Phyu, 2009). The idea of classification of the graph has obtained great consideration in both industry and academia to classify relationships of the entities in a graph. Two important classifications of graph-based technique are (1) Label propagation and (2) Graph classification. In the label propagation method, labels are associated with a subset of nodes in the graph while in the graph classification; labels are given to a subset of graphs in graph data. Studies have not most commonly used these methods in network analysis. However, Taskar, Abbeel et al. (Taskar, Abbeel, & Koller, 2002) developed a framework that builds on Markov networks and solves two limitations of Probabilistic Relational models and relational version of the Bayesian network. They used undirected models because these models do not impose the acyclicity constraint and stop the display of many significant relational dependencies in the directed model. Secondly, the undirected models are also good for discriminative training, which improves classification accuracy. Zhou, Bousquet et al. (D. Zhou, Bousquet, Lal, Weston, & Schölkopf, 2004) in learning local and global consistency have used the label propagation approach. They used a principled approach to semi-supervised learning to develop a classification function. This function used a simple algorithm and was significantly smooth concerning the intrinsic structure specified by labelled and unlabelled points.

After a few years, Zhou, Huang et al. (D. Zhou, Huang, & Schölkopf, 2005)proposed a framework for directed graphs. They used this method to classify graphs based on labelled and unlabelled data graphs. Their algorithm was better than the previously used techniques and results were nearly linear. They used this technique on real data of the World Wide Web with splendid results. While, Saigo, Nowozin et al. (Saigo, Nowozin, Kadowaki, Kudo, & Tsuda, 2009)used this method for mathematical programming model and Zaki (Zaki, 2005) for an effective classifier of XML data used on graph classification technique. Chuang, Lee, Liu, Lee, & Ideker (Chuang, Lee, Liu, Lee, & Ideker, 2007) used a network-based classification technique to classify breast cancer metastasis as shown in Figure 5.

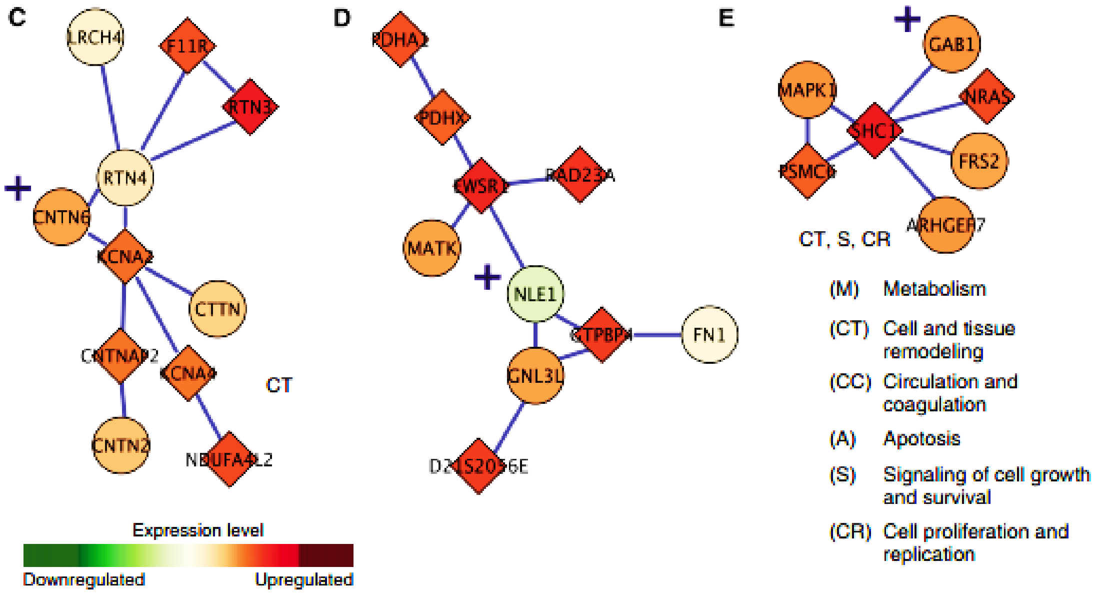


Figure 5 classification technique to classify breast cancer metastasis (Chuang et al., 2007)

Zhang (D. Zhang, Dai, Li, & Zhang, 2015) developed a new method to measure the influence in the dynamic social networks using the user-content bipartite graph model. They also proposed an algorithm to compute scores of the 'user reach' and 'influence boards'. Using these techniques, we can easily discover the most influential user on social networks and popular broads.

The classification is a method of knowledge dissemination throughout the network where the arrows (see Figure 6) represent possible knowledge flow. Figure 6 represents the classification process on the bibliographic heterogeneous network. As HINs are multi-typed the label of objects is decided by the effects of various typed objects and links between them. Researchers have also extended traditional classification methods to HINs classification and some studies use the transductive task to predict labels for unlabelled data. For instance, GNetMine proposed by Ji et al. (Ji, Sun, Danilevsky, Han, & Gao, 2010) classify the link structure using arbitrary network schema and number object types. Lou et al. (Luo, Guan, Wang, & Lin, 2014) introduced Het-pathMine to cluster the labelled data on HIN through meta-path selection model and Jacob et al.(Jacob, Denoyer, & Gallinari, 2014) gave a conceptual method of classification by calculating latent representation of a node. Some studies also employed inductive classification methods to develop a decision function for complete HIN data. For instance, Rossi et al. developed a novel algorithm IMBHN using this method(Rossi, de Paulo Faleiros, de Andrade Lopes, & Rezende, 2012). Multi-label classification is also being used for HIN, Angelova et al.(Angelova, Kasneci, & Weikum, 2012) proposed a method using random walks process to model mutual influence between nodes and Zhou et al.(Y. Zhou & Liu, 2014) also employed edge-centric multi-label classification method. Meta-paths based classification has also been used by various studies and recently used this method for text classification in HINs(Chenguang Wang, Song, Li, Zhang, & Han, 2016). The classification process has also been integrated used with other data mining methods to classify HIN(Jendoubi, Martin, Liétard, & Yaghlane, 2014; Ji, Han, & Danilevsky, 2011).

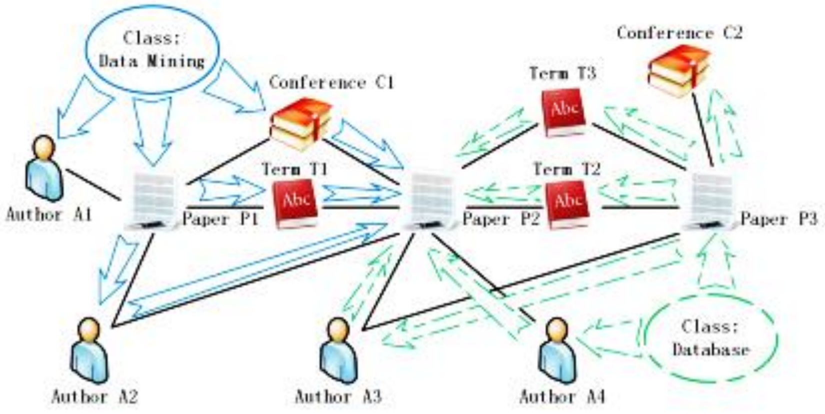


Figure 6 Classification process on DBLP HIN (Ji et al., 2010)

In HIN several meta-paths associating nodes of the target type, on which classification is done, make the classification problematic. To resolve this problem Gupta et al. (Mukul Gupta, Kumar, & Bhasker, 2017)proposed a HeteClass meta-path based framework for translative classification of target type nodes. Their method discovers the network schema and can also use the knowledge of the domain expert to develop a set of meta-paths. Recent HIN classification tasks include deep collective classification (Y. Zhang et al., 2018); Meta-path based classification (J. Zhang, Jiang, & Li, 2018); classification employing multi-semantic (Du, Guo, Liu, & Yao, 2019); and context-aware meta-path based classification of HINs (X. Li, Ding, Kao, Sun, & Mamoulis, 2020).

2.3.2 Unsupervised learning methods

Unsupervised data mining methods does not have predetermined class or target variables and find relation and structures that are hidden in the data. This learning technique is also a synonym for clustering. In this type of learning method, the input examples are not class labelled. Usually, clustering is used to discover classes in the data(Han, Pei, et al., 2011).

2.3.2.1 HIN analysis based on Clustering

Clustering is a significant method to analyze and evaluate large networks. Various studies used Cluster-based algorithms to extract information on a global scale in a dynamic network environment using the graph of the network. The clustering algorithms have an important application in different graph plans. For example, it has been used previously for XML data integration, facility location and congestion detection. Clustering is a data mining method of grouping similar objects into distinct groups or more specifically clustering is dividing a set of data objects into a set of clusters. Similar objects are classified in a cluster and dissimilar objects in other clusters. The two most important approaches to cluster graphs are Node Clustering and Graph clustering (M. L. Lee, Yang, Hsu, & Yang, 2002).

2.3.2.2 Node Based Graph Clustering

In node-based graph clustering, there is only one enormous graph and underlying nodes of the graph are clustered using the distance between edges. It assigns a particular numerical distance value is assigned to each edge in the graph is assigned value and then used to create a cluster of nodes. For instance, in a specific situation, the absence of the edge is assigned with a value of 0, while the presence of an edge is given a value of 1. Table 2.1 shows the studies using this technique in the domain of network study. Nettleton (2013) has presented a detailed survey of different node-based techniques used for social networks based on the graph structure. He has divided the survey into four major variations: graph theory, social network, online social networks and graph mining. While in the second part, he has emphasized five major themes of the social network: communities, influence and recommendation, models metrics and dynamics, behaviour and relationships and information diffusion.

Aissa and Belghith (2014) used the node clustering idea, using node degree as an important weight metric in the cluster head selection process to improve the weighted clustering algorithm and other similar algorithms. Furthermore, they proposed the idea of finding critical graph nodes of structural properties of a network to analyse it effectively.

A very recent advancement employing this technique is by Boobalan, Lopez, & Gao (2016), they have introduced a novel graph clustering algorithm kNAS to detect overlapping community structure, in a large graph by joining the attribute similarity and topological values into m clusters, based on intra-cluster and low inter-cluster similarity. They have used the core node based on the local outlier factor and developed a new function for faster convergence of the proposed algorithm. Furthermore, they used a density and Tanimoto coefficient to validate the effectiveness and quality of the proposed method. Figure 7 gives an idea of node-based clustering to understand community structure. Table 2.1 also gives a comparative analysis of some studies using this approach.

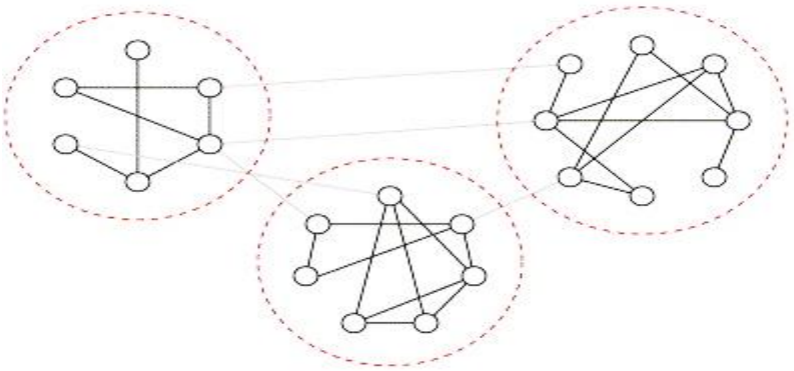


Figure 7 Node-based clustering to understand community structure (Newman & Girvan, 2004)

Study	**Features of the Study.**
Kernighan and Lin (1970)	1. Considered the issue of partitioning the nodes of a graph. 2. Introduced a heuristic strategy for dividing discretionary graphs, which is both powerful in finding ideal partitions, and sufficiently quick to be useful in taking care of huge issues.
Ahuja, Magnanti et al. (1988)	1. Tested a percentage of the major algorithmic thoughts created in the late years. 2. Facilitated and gave point-by-point knowledge into algorithmic conduct. 3. Their study found that an ideal opportunity to perform relabel operations (or developing the layered systems) takes in any event as much calculation time as that taken by enlargements and/or pushes
(Fjällström, 1998)	1. Provided a complete survey of graph partitioning using this approach used in the past.
Flake, Tarjan et al. (2004)	1. Introduced straightforward graph clustering strategies in light of minimum cuts inside of the graphs. 2. The researchers demonstrated that the nature of the delivered cluster is limited by solid minimum cut and extension criteria. They likewise added a system for hierarchical clustering and present applications to real-world data. 3. Concluded that clustering algorithms fulfil solid hypothetical criteria and perform well practically
Gibson, Kumar et al. (2005)	1. Gave algorithm for discovering vast, dense sub-graphs in gigantic graphs. 2. The algorithm depends on a recursive use of fingerprinting utilizing shingles and is greatly productive, capable of taking care of graphs with tens of billions of edges on a solitary machine with modest assets.
(Pei, Jiang, & Zhang, 2005)	1. Clusters found from mining protein interactions and gene interactions are important due to two reasons, the quality expression information and the protein information are extremely noisy.
Zeng, Wang et al. (2007)	1. They Concentrated on dense graph-based databases to mining closed quasi-cliques. 2. Proposed novel optimization techniques to prune redundant subgraphs.

	3. They also devised an effective conclusion checking plan to encourage the closed quasi-clique only.
Cooper, Kalsin and Radzik (2008)	1. Provided a new algorithm for assigning neighbours to vertices joining dynamic distributed networks. 2. Each vertex of the graph was assigned a constant number of tokens to the network and any vertex can use these tokens if it is making a connection to the donor vertex.
Zhang & Kumar (2009)	1. A new algorithm called E-Spring was introduced which eradicated node similarities in clustered directed acyclic graphs. 2. The algorithm node was displayed as dynamic elements with weight and the final is derived by adjusting the positions of the nodes according to a grouping value between nodes. They introduce a stopping condition for stable state nodes.
Caron, Datta et al. (2010)	1. Used a K-Clustering graph for a mobile ad-hoc dynamic network. 2. The k-clustering was a partition of nodes into disjoint sets called a cluster.
Qiu & Lin (2011)	1. Used data structure named Community tree to display the organizational structure of a network and provided a framework for explaining the organizational structure in a social network. 2. A tree learning algorithm was used as a scoring function to extract an evolving community tree.
Zheng, et al. (2012).	1. Presented a new platform using nodes based graph technique to detect community evaluation in dynamic networks. 2. They also introduced a new concept of meta-community which can be detected by matching algorithm.
(Y. Chen, Sanghavi, & Xu, 2014)	1. Developed and empirically tested a novel algorithm for the maximum likelihood of graph clustering. 2. The algorithm can handle the semi-random graphs, heterogenous degree distributions, planted cliques and unequal clusters.
(Parimala & Lopez, 2015)	1. Proposed a new graph clustering algorithm, Structural Attribute Neighborhood Similarity (SANS). 2. The algorithm partitions the graphs using structural similarity and clusters are developed using the degree of contribution of vertex attributes with the vertex in the partitions.

(Moradi & Rostami, 2015)	1. Proposed supervised filter-based method using graph clustering and ant colony optimisation for classification problems. 2. The proposed method works firstly by representing the entire feature set as a graph, secondly, the features are grouped into clusters and finally, ant colony optimisation is used to select the final subset of features.
(Yin, Benson, Leskovec, & Gleich, 2017)	1. Introduced a new class of local graph clustering methods to address the issues in the existing local graph clustering techniques. 2. Developed motif-based approximate personalised PageRank algorithm that creates clusters containing a seed node with minimal motifs conductance.
(Chun Wang, Pan, Long, Zhu, & Jiang, 2017)	1. Presented a new deep learning-based marginalised graph autoencoder (MGAE) unsupervised learning algorithm for node oriented graph clustering. 2. MGAE advances deep learning autoencoder to graph domain which enables graph representation learning to be carried out in an unsupervised setting by using structural and content information.
(Berahmand, Haghani, Rostami, & Li, 2020)	1. Proposed a new version of label propagation algorithm for attributed graphs, to solve the problems related to structural cohesiveness and homogeneity in the network communities. 2. In the proposed algorithm, the influence of each node is calculated using laplacian centrality which makes this method more efficient and effective.

Table 2.1: Studies using node-based clustering approach

2.3.2.3 Complete graph clustering-based approach

In this type of algorithms, a large number of graphs are analyzed using their structural behaviour. These types of algorithms are discussed in both classical graph data and semi-structured data. There are two types of algorithms in this scenario structural distance-based and structural summary based. These classical graph approaches are quite similar to XML data because when the XML data is organized structurally, it is similar to graph data. This approach is further subdivided into structural and summery based approaches (Aggarwal and Wang 2010).

In the structural approach algorithms, the structural distance between XML documents is computed to create clusters of documents. Chawathe (1999) introduced a new algorithm that used auxiliary memory to compute the minimum cost between two labelled trees. This algorithm was able to make powerful utilization of RAM and lessen the I/O cost. Lee, Yang et al. (2002) presented XClust, a novel coordination methodology that includes the clustering of DTDs. A matching algorithm based on semantics, quick descendants and leaf-context similarity of DTD elements is produced. Their experimental results show that real-world DTDs exhibit the viability of the XClust approach. Lian, Cheung et al. (2004) proposed an algorithm (S-GRACE). This algorithm was capable of clustering XML documents in large sizes. This proposed algorithm is supporting the reduction of the high computational cost related to the processing of these documents using separation metrics.

In the summary based approach, summaries of the underlying XLM documents are used to create clusters. (Dalamagas, Cheng, Winkel, & Sellis, 2005) used this approach to recommend the utilization of tree structure and it helped to keep up or even enhance its quality. Furthermore, (Aggarwal, Ta, Wang, Feng, & Zaki, 2007) propose a viable algorithm for XML. This algorithm used the subparts of the document to extract information about the hidden structures. They proposed better approaches for using numerous XML documents in the form of substructures. Moreover, clustering massive graph approaches have been used in many evolving network research articles. The most important research on clustering evolving networks was started in the 20th century. Table 2.2 gives a deep overview of research studies graph-based clustering approach. Moreover, (Riedy, Meyerhenke, Ediger, & Bader, 2012) recommended a method to attain reasonable parallel scalability without losing sequential operational complexity. They took an agglomerative method alike to Clauset, Newman, and Moore’s sequential algorithm, integrating pairs of associated middle subgraphs to improve different graph properties. Stanton (Stanton & Kliot, 2012) offered natural, heuristics and relate their significance to hashing and METIS, a fast, offline heuristic. They characterized on a huge collection of graph datasets that their heuristics are an important enhancement, with an average improvement of 76%.

(Görke, Maillard, Schumm, Staudt, & Wagner, 2013) provided progressive and scalable dynamization of the presently fastest and extensive static heuristics and plan a heuristic dynamization of static algorithm. Their algorithms proficiently sustain a modularity-based clustering of a graph for which dynamic changes occur as a stream. Furthermore,(Held & Kruse, 2013) offered a conception to display the varying actions of clusters in dynamic networks. They improved the MONIC framework to trail the clusters over time. They tracked relations during the lifetime of a cluster birth, death, growth, contraction, splitting, and merging. In addition to this, (C.-D. Wang, Lai, & Yu, 2013) and (C. Lee & Cunningham, 2013) has also used this method in community detection. Moreover, (Hartmann et al., 2014) have given a list of different approaches available to detect community in evolving networks.

Maslennikov, (Maslennikov & Nekorkin, 2015) studied transient sequential dynamics of evolving dynamical networks. They display that such networks can create structures of Meta-stable cluster states. They also found the method how the structures produced by such networks can be vigorous against background noise, small perturbations of initial conditions, and parameter detuning, and at the same time, can be delicate to input. Zhang, (D. Zhang et al., 2015) has used a cluster-based approach to introduce a new model for bipartite evolving networks.

Study	A critical review of the Study.
(C. H. Ding, He, Zha, Gu, & Simon, 2001)	1. Developed min-max cut algorithm for graph partitioning. 2. They proposed an algorithm for graph partition with a target function that follows the min-max clustering standard. The min-max cut algorithm tested on the newsgroup data set is found to out-perform other current prominent clustering methods.
(Condon & Karp, 2001)	1. Gave algorithm for the graph l-segment issue and they break down it on an irregular "planted l-segment" model. 2. The model represents n nodes of a graph partitioned into l groups, each of size n/l; two nodes in the same group are associated by an edge with some likelihood p, and two nodes in various gatherings are associated by an edge with some likelihood r<p.
Girvan & Newman (2002)	Highlighted structure of communities in which the nodes of the network are present in the form of weave groups.
Newman (2003)	Surveyed improvements in this field, including ideas as little world impact, degree circulations, clustering, network relationships, irregular graph models, models of system development and particular connection, and dynamical procedures occurring on systems.
Vázquez (2003)	Demonstrated that the local models offer a clarification for different properties, such as the clustering chain of command and degree connections.
Yu and Shi (2003)	Presented principled record on multiclass spectral clustering and gave a discrete grouping definition to tackle a casual constant enhancement issue by Eigen decomposition.
Chakrabarti (2004)	1. A novel approach without parameters to gather nodes, utilizing information-theoretic standards. 2. They also proposed novel algorithms that use this node group structure to get further insight into the information, by discovering outliers and calculating distances between groups

Flake, Tarjan et al. (2004)	Framework for various hierarchical clustering and graph clustering strategies in light of minimum cuts within the graph.
Gloor & Zhao (2004) (Hopcroft, Khan, Kulis, & Selman, 2004)	Methodology for organizational upgrade and optimization of communication streams and Temporal Communication Flow Visualizer to produce motion picture of communication stream among people. Specified stable communities in the large linked network, which remain unchanged after various cluster executions.
Leskovec, Kleinberg et al. (2005)	Introduced a new method based on the forest fire model. This method was able to show the complete properties of a graph using a smaller number of parameters.
(Moody, McFarland, & Bender-deMoll, 2005)	The idea of the temporal representations of social networks. Solved How to best link network change to changes in the graphical representation.
Palla, et al. (2005)	The idea is to uncover the overlapping community structure of complex networks.
Falkowski, et al. (2006)	Introduced two systems to examine communities, the primary system includes statistical analyses and perceptions. The second strategy is intended for the discovery of groups in a domain with exceedingly fluctuating individuals.
Palla, Barabási et al. (2007)	A novel algorithm was introduced. It was able to detect the time dependency of the communities in large data sets and it also uncovers the associations between communities during the evolution process.
(U. N. Raghavan, Albert, & Kumara, 2007)	Used and iterative process using label propagation method. The nodes were given a unique label at every step iteratively for mining communities.
(Tong et al., 2008)	Provided a family of Colibri algorithms Colibri-S and Colibri-D to deal: 1. Low-rank approximations of the adjacency matrix of a graph to find patterns and detect anomalies. 2. Track low-rank structures as the graph evolve within limited storage to deal with the thousands and millions of nodes of the sparse dynamic evolving graphs.

(Bagrow, 2008)	Devised a method that is unambiguous and specific to local community findings.
Asur, Parthasarathy et al. (2009)	Proposed a new model to analyse the interesting properties of graphs and behaviours of individuals in the communities.
Lancichinetti and Fortunato (2009)	They presented an idea to generate directed and weighted network graphs using a built-in community structure.
Meyerhenke, Monien et al. (2009)	Proposed TruncCons Faster method for the improvement of the partitioning of graphs. The method is based on a different diffusive process, have a high degree of parallelism and is related to the local area properties of graphs.
Nicosia, Mangioni et al. (2009)	Evaluated the goodness of network community decompositions, and extended it to the more general case of directed graphs using modularity function. Method for finding overlapping communities.
Aggarwal, et al. (2010)	Gave a new framework to solve the problem of clustering gigantic graph streams.
Fortunato (2010)	Introduced clustering methods based on minimum cuts within the graph. Developed a framework for hierarchical clustering.
Bogdanov, et al. (2011)	Devised an efficient approach for large graph instances that evolve over long periods. The subgraph of a dynamic network with the highest score based on degree was also solved using this technique.
Doll, et al. (2011)	Proposed a novel method for hierarchical clustering.
Görke, et al. (2011)	The solution to finding clustering of graphs that simultaneously guarantee good inter-cluster quality.
Nguyen, et al. (2011) Yang, et al. (2011)	A novel method to find communities in a dynamic network environment. They used social network data to analyse their empirical results.
(Agarwal, Ramamritham, & Bhide, 2012)	1. Methods for microblog message streams to discover the important messages in the steams. 2. Presented the problem in the form of dense clusters graphs. A novel technique to detect communities in dynamic graphs based on real-time data.

	3. Short cycle property was introduced to use all events in the clusters.
Angel, et al. (2012)	Presented effective preservation of condensed subgraphs under streaming edge weight updates. Suggested a novel algorithm, DYNDNS, which outpaces variations of existing techniques to this setting, and produces significant results.
(Campello, Moulavi, & Sander, 2013)	Developed an improved density-based clustering method providing a clustering hierarchy from which a tree of important clusters can be constructed.
(Mina & Guzzi, 2014)	Presented AlignMCL algorithm based on alignment graph and Markov clustering. They applied their approach to the protein behaviour dataset at the network level and the proposed algorithm performed better than other states of the art algorithms.
(G. Ma et al., 2016)	Proposed multi-graph clustering method (MGCT) based on the interior node topology of graphs to investigate the problem of clustering multiple graphs.
(Kamis, Chiclana, & Levesley, 2018)	This study used the concepts of social network analysis and consensus-based decision making to analyse social structures and patterns of network relationships.
(C. He, Liu, Zhang, & Zheng, 2019)	A novel method for graph partitioning based on fuzzy clustering was introduced, the proposed approach uses SimRank algorithm to develop a fuzzy similarity matrix then it deduce the corresponding fuzzy equivalent matrix using fuzzy set theory.
(C. Wu, Gu, & Yu, 2019)	Density Peak-based Structural Clustering Algorithm for Networks(DPSCAN) was developed and employed in real and synthetic graphs to demonstrate that this approach works better than its counterparts in detecting meaningful clusters, hubs and outliers.

Table 2.2; Studies using the graph-based clustering approach

The traditional clustering process is based on the features of objects, such as k-means(Jain, 2010). Some research studies(Y. Zhou, Cheng, & Yu, 2009) consider the link structure of objects and attribute information to improve the accuracy of clustering. Recent research has focused on the clustering of heterogeneous networks data. In contrast to homogeneous networks, the heterogeneous networks integrate multi-typed objects and links. There are various challenges associated with heterogeneous and multi-type clustering which leads to new clustering standards. Same topics and multi-type objects links in a network can be incorporated in clusters or sub-network clusters. An IMDB heterogeneous network is one of the examples that divide the network into different levels of the cluster. For example, a cluster of movies consists of a set of movies, venues, actors, and genres. Similarly, the bibliographic heterogeneous network can also be grouped based on authors, papers and domain of the papers. Figure 8 is an example of clustering based on DBLP data. In the multi-level clustering process, clustering in HIN keeps richer information, but it also creates more challenges. However, the benefits of preserving the abundant information in HIN makes it more convenient to integrate additional information for clustering (C. Shi & Philip, 2017).

Further, the literature in this study also considers integrating supplementary information about the attribute of the variable in clustering. This additional information about the objects' attribute added into clustering analysis on the heterogeneous information support the creation of a composed group of communities. To extract compressed descriptions of the underlying community Aggarwal et al. (2012) used the local succinctness property which supports in creation of a composed group of communities in a HIN. Sun et al.(Y. Sun, Aggarwal, & Han, 2012) and Qi et al. (Qi, Aggarwal, & Huang, 2012) also developed two clustering algorithms using structural information in the links of HIN. Former clustering algorithm uses relational strength of the links incompleteness while later used outlier connections and random fields to develop clusters. Deng et al.(Deng, Han, Zhao, Yu, & Lin, 2011) introduced a new approach of clustering by combining HIN with topic modelling. This approach also employed a probability-based topic modelling method to simultaneously model multi-type nodes of a HIN. A topic modelling approach was further extended and optimized in the researches (Deng, Zhao, & Han, 2011 and Q. Wang et al., 2015) for clustering of multi-type networks.

Typically, clustering is an unsupervised data mining task, but it can be combined with other data mining methods to increase the clustering effectiveness and efficiency some studies have employed rank-based clustering and semi-supervised clustering methods to group HIN objects. Semi-supervise learning is a method to partition unlabelled data using domain knowledge expressed as pairwise user-guided constraints among instances (Basu, Banerjee, & Mooney, 2002). SemiRPClus(Luo, Pang, & Wang, 2014) and PathSelClus(Y. Sun et al., 2013) are Semi-supervised clustering learning algorithms in HIN. PathSelClus produce different clustering outputs using path based on user input and SemiRPClus use the concept of relation-path to measure the similarity between same-typed nodes to develop HIN clusters. Different algorithms such as Rank-Clus(Y. Sun, Han, et al., 2009) Netclus (Y. Sun, Yu, & Han, 2009) comClus(R. Wang, Shi, Philip, & Wu, 2013) HEProjI(R. Wang et al., 2013) have employed ranking based clustering.

Community detection and outlier detection is also hot topic in HIN analysis. Both tasks are related to each other, but their aims are different. Community detection is a method of finding a group with similar characteristics (Y. Sun, Tang, Han, Gupta, & Zhao, 2010) while outlier detection is the method of finding data elements that are different from expectations (Qi et al., 2012). Some notable researches in the field of community detection are OcdRank(Qiu, Chen, Wang, & Lei, 2015) to detect overlapping communities and EVRA(Huang et al., 2018) detects overlapping communities. Recently, Chunaev (Chunaev, 2020) have surveyed community detection algorithms in detail to elaborate on the concepts of community detection and state of the art methods available to detect communities in a multi-typed network. Gao et al. (J. Gao et al., 2010) introduce the concept of community outlier detection by modelling network data as a mixture model composed o multiple normal communities and a group of randomly created outliers. Gupta et al. (Manish Gupta, Gao, & Han, 2013) propose the concept of CDOutliers for HINs. They used an iterative two-stage method of pattern discovery and outlier detection in a tightly integrated system using non-negative matrix factorization. Kuck et al. (Kuck, Zhuang, Yan, Cam, & Han, 2015) proposed a query language that enables a user to extract query-based outliers in HINs. Furthermore, Liu and Wang proposed the meta-path-based outlier detection method (MPOutliers) (L. Liu & Wang, 2020). Very recent, research development of HIN in clustering tasks include coupled semi-supervised clustering(J. Zhao, Xiao, Hu, & Shi, 2019); spectral clustering (X. Li, Kao, Ren, & Yin, 2019); extended star-schema based clustering (Mei, Lv, Yang, & Li, 2019); weighted meta-path embedded clustering (Y. Zhang, Yang, Wang, & Li, 2020), and mutual clustering on comparative text via HINs (J. Cao et al., 2020).

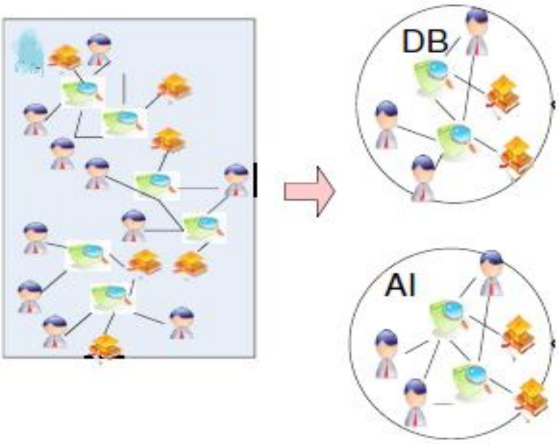


Figure 8 Clustering authors, papers based on domain(Y. Sun et al., 2010)

2.4 Other Data Mining tasks to analyse HINs

Besides the supervised and unsupervised data mining tasks to analyse HINs, there are various other data mining tasks to analyse HINs, such as link prediction and ranking (Y. Sun & Han, 2013).

2.4.1 Link prediction

Based on observed links and the attributes of nodes, link prediction attempts to estimate the likelihood of the existence of a link between two nodes. This is a significant research topic for recent years, which has the following characteristics (C. Shi & Philip, 2017):

1. The links to be predicted are of different types as nodes in HIN relate to different types of links.
2. There are dependencies present among multiple types of links.

2.4.2 Ranking

The ranking evaluates network object importance based on ranking functions. This is also an important data mining task for HIN analysis. Although, ranking is a meaningful task in HIN analysis but has some challenges for example (C. Shi et al., 2016):

1. HIN are multi-type and multi-relational, treating all objects equally will mix different types of objects.
2. The semantic meaning of the objects affects the ranking results

Due to the scope of this research, we have only considered supervised and unsupervised learning methods to analyze HINs.

2.5 Summary

In this chapter, a detailed background overview of heterogeneous information networks (HINs) was presented for explaining the basic building blocks of a HIN. The described background of the heterogeneous information networks included the preliminaries of HINs and a comparison of different types of networks. Furthermore, data mining supervised and unsupervised learning methods to analyse HINs were critically overviewed with their characteristics and drawbacks. Finally, other data mining methods link prediction and ranking were briefly elaborated to conclude the chapter.

In the next chapter of the thesis literature review of dynamic heterogeneous information, networks will be presented. Preliminaries of DHINs will be defined and explained. Moreover, the research gap will also be discussed after completing the change discovery related works survey.

Chapter 3 Literature Review

3.1 Introduction

Dynamic heterogeneous information networks (DHINs) are the networks that evolve with time. In these types of networks nodes and linkage, structures change over time. It is a hot research topic for a decade, and it is intriguing for various researchers due to the capacity of dynamic systems to represent social, multi-typed and complex systems. In DHIN change detection we divide a large DHIN network into a series of temporal snapshots of the same DHIN network, and then apply graph theory, feature extraction, generative methods, pattern mining and clustering methods to extract the changes in different time snapshots of the DHIN network. These changes in the network can be local, global and community level which is also a research question of this research. This chapter is divided into 12 subsections and presents a detailed literature review of DHIN and its applications it also overviews different dynamic change detection algorithms and frameworks.

3.1.1 Dynamic Network

Definition: A dynamic network at a specific timestamp is defined as $G_t = (V_t, E_t)$ where V_t and E_t are, respectively, the set of nodes and the set of links existing in the network at the timestamp *t*(T. Zhu, Li, Yu, Chen, & Chen, 2020).

3.1.2 Dynamic Heterogeneous information networks

Definition: Dynamic heterogeneous information network is a network with a function of five tuples G_D=(V, E, φ, ψ, D) (Jia, Wang, Jin, Zhao, & Cheng, 2017) where:

- V is the set of nodes or vertices such that each vertex $v \in V$ is associated with dynamic information $d \in D$ where D is dynamic information set.
- E is the set of edges and set of triples (u, v, r) for $u, v \in V$ and $r \in R$ such that each edge is attached with some dynamic information $d \in D$.
- $\varphi: V \rightarrow A$ is a vertex type mapping function on the vertex set such that each vertex $v \in V$ is assigned a single type $\varphi(v) \in A$. $\psi: E \rightarrow R$ is a relation type mapping function such that each pair of vertices is assigned at most |R| relations

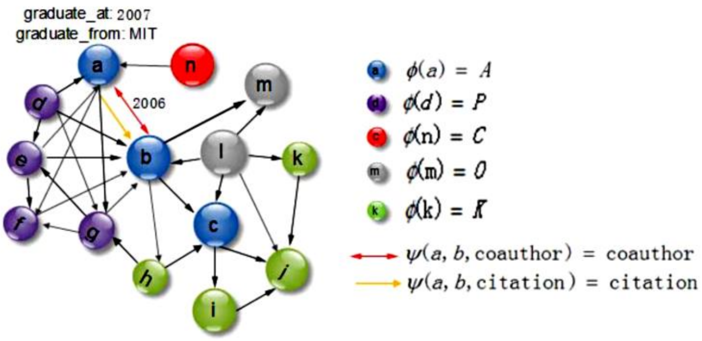


Figure 9 (a) An example of DHIN(Jia et al., 2017)

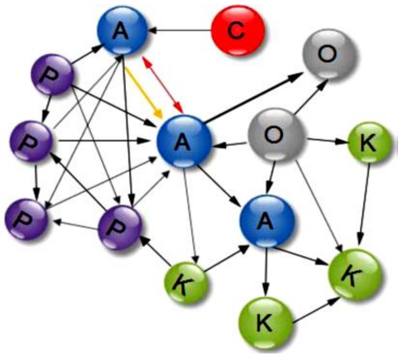


Figure 10 (b) DHIN Schema(Jia et al., 2017)

For example, Figure 9 (a) and Figure 10 (b) are examples of an academia DHIN and network schema respectively, where nodes are of five different types: authors(A), conferences(C), organizations(O), paper (P) and key terms (K). The edges of this DHIN can denote the co-author relationship between authors at a specific time. We can see dynamic information of node "*a*" in figure 9 (a) which shows that vertex "*a*" graduated from MIT in 2007 and he also has co-author relationship with node "*b*" in the year 2006. Figure 10 (b) represent the domain-specific network schema of academic dynamic heterogeneous information network and gives the conceptual and relational representation of the vertices in the network.

3.2 Why DHINs change detection

Since, Sun, Han et al. introduced the concepts of Heterogeneous information networks in 2009 and meta path in 2011 respectively, Heterogeneous information network analysis has become a very important topic rapidly in the fields of data mining, databases and information retrieval (C. Shi et al., 2016). Various articles have been written on the topic of change detection and mining but the analysis of this problem in a dynamic heterogeneous information network environment is very recent (Loglisci et al., 2015). There are many data mining methods available in the literature for homogenous networks. Nevertheless, due to unique characteristics (e.g., rich semantic, time evolution, the fusion of more information) of DHINs, most techniques of homogeneous networks cannot be directly applied to DHINs (C. Shi & Philip, 2017).

Following are the benefits of dynamic heterogeneous information networks change discovery:

3.2.1 It helps to extract rich semantics

Mining interesting patterns with semantic information is a unique issue in DHINs. In real-world applications, patterns and relations in networks are dynamic, multi-typed, heterogeneous and evolve with time. In this type of network, the multi-typed objects and their links carry different semantic meanings(C. Shi & Philip, 2017). For example, IMDB movie heterogeneous network contains actor, movie and venue object types. The relationship type "Actor-Movie" means actors working in a movie, while the relation type "Movie-Venue-Time" movie is released in a specific venue and time. So the extraction of this semantic and temporal information through change detection will lead to more elusive knowledge discovery. For instance, in IMDB movie data if someone wants to find actors who performed in movies with "Tom Cruise" in different years, can do that by using knowledge extraction and semantic reasoning of the temporal data.

3.2.2 A tool to fuse more temporal information

DHINs, as compared to homogenous, static and heterogeneous information networks, can fuse more information from multiple data sources, objects and their temporal interactions. For instance, people can use many social network platforms (e.g., Twitter, Snapchat, WeChat, Facebook) and use various services provided by these platforms such as marketing, messaging, networking and shopping etcetera. So using DHINs change detection we can fuse more temporal information related to people behaviours across multiple services of social network platforms(C. Shi et al., 2016).

3.2.3 A novel development of data mining

With the beginning of WWW in late 1990, data mining researchers started to study the links among data objects and constructed homogenous relational objects. But in recent years, due to the abundance of social networks, many different types of objects are being interconnected and it is very difficult to model these relational objects as homogenous networks. Naturally, these objects can be displayed as DHINs. Specifically, with the rapid generation of online content, big data is a new yet important task to be studied. For XML (semi-structured) representation, DHINs can be an effective tool to model complex relational and temporal objects in big data(C. Shi & Philip, 2017).

3.3 Pattern Mining Dynamic Evolving networks

The network nodes are descriptions of the entities of a network system and the relationship between these nodes is called edge. Connection of the nodes can either be direct or indirect and dynamic for example in the case of co-authorship networks where the relations are associated with the co-author indirectly and temporal while if we consider the network of teachers marking a student it is considered a directed relation and teacher can only mark student. Networks are also expressed with the help of labels to express the importance of nodes; these labels are called the features of the nodes. In many scenarios, nodes that represent the importance of the corresponding relations are also given special significance. Furthermore, networks can also express either simple or complex nodes and edges that are based on the labels assigned to the network (R. Ahmed & Karypis, 2015a).

Extensive research is dedicated to evolving algorithms capable of finding patterns in graphs and networks. Though most of these methods have been developed for mining databases comprising relatively small graphs, algorithms have also been developed to identify subgraphs with a large number of embedding in a single large graph such as GREW, a heuristic algorithm designed by (Kuramochi & Karypis, 2004c). Similarly, later in 2005, Kuramochi and Karypis presented algorithms, based on both horizontal and vertical pattern discovery paradigms. Many other studies also give different variations on evolving networks, Such as induced subgraphs (Inokuchi, Washio, & Motoda, 2000b), shortest paths finding (De Raedt & Kramer, 2001; Han et al., 2001), trees structure in networks,(Zaki, 2002); arbitrarily connected subgraphs (Lian, Cheung, Mamoulis, & Yiu, 2004) and numerous kinds of cliques (Zeng, Wang, Zhou, & Karypis, 2007).

Yoshida & Motoda (Yoshida & Motoda, 1995)used heuristics to select relevant patterns, based on minimum description length criteria other than occurrence frequency. They presented the Subdue system that discovers substructures to compress the original data and represent structural concepts in the data by replacing previously discovered substructures in the data. Subdue uses a computationally bounded inexact graph match that identifies similar *r,* but not identical, instances of a substructure and finds an approximate measure of closeness of two substructures when under computational constrictions.

Pei and et al.(Han et al., 2001) formulated PrefixSpan, an innovative sequential pattern mining method that explores prefix projection in sequential pattern mining. PrefixSpan outperformed both the apriori-based GSP algorithm and FreeSpan, in mining large sequence graph databases. TREEMINER, a unique algorithm was offered by Zaki(Zaki, 2002) to discover all recurrent subtrees in a forest, using a new data structure called scope-list, while mining (embedded) subtrees in a forest of rooted, labelled, and well-arranged trees. Raedt and Kramer (De Raedt & Kramer, 2001) proposed a simple and abstract model for inductive databases by defining the basic formalism, a simple but accurately powerful inductive query language and some basics of reasoning for query optimization along with memory organization and associated application issues. Inokuchi (Inokuchi et al., 2000a) proposed a new approach called AcGM, which achieves the complete search of regularly, connected (induced) subgraphs in a massive labelled graph dataset within a highly practical time. AcGM derives its powers from the algebraic illustration of graphs, its supplementary operations and efficient constraints to limit the search space proficiently.

In 2003, Han, and Afshar(Yan, Han, & Afshar, 2003) presented an efficient algorithm CloSpan for mining the closed sequential patterns based on cropping method called occurrence checking for the initial detection of closed chronological patterns during the mining process. Huan et al. (Huan, Wang, Prins, & Yang, 2004) developed an effective index structure, ADI (for adjacency index), to support mining numerous graph patterns over large databases unable to be stored in main memory which proved faster when compared to gSpan in performance. In the same year, (Kuramochi & Karypis, 2004c) designed GREW, a heuristic algorithm to overcome the limitations of existing complete or heuristic frequent subgraph discovery algorithms to operate on a large graph and to find patterns corresponding to connected subgraphs having a large number of vertex-disjoint embeddings.

The following year, Kuramochi and Karypis (Kuramochi & Karypis, 2005) presented algorithms, based on both horizontal and vertical pattern discovery paradigms to find the associated subgraphs having a sufficient number of edge-disjoint embeddings in a single large undirected labelled sparse graph using three different methods of finding the number of edge-disjoint embeddings of a subgraph. They employed different algorithms for contender generation and frequency counting, which finally allowed the operation on datasets with different characteristics to quickly prune unpromising subgraphs. The issues of mining closed frequent graphs with connectivity constraints in massive relational graphs were investigated by (Yan, Zhou, & Han, 2005) through the adoption of edge connectivity concept by applying the results from graph theory, to accelerate the mining process. Zeng (Zeng et al., 2007) proposed optimization techniques that can prune the unpromising and redundant sub-search spaces effectively by devising an efficient closure checking scheme to facilitate the discovery of closed quasi-cliques.

To Trace and identify certain features of emerging communities in social networks has been a great investigating task for many studies. Evolutionary clustering, one of the most studied problems in this field to discover clusters in a dynamic network using temporal snapshots of network data (Chi, Song, Zhou, Hino, & Tseng, 2007; Tang et al., 2008). Some of the others closely related problems of this area are, tracing the vicinity of two entities in the network (Tong et al., 2008). Moreover, this topic is investigated in the detection of some important events such as the development and splitting of current societies, similarly, the creation of new ones, and finding constant or determined communities along with their long-lasting members (Berger-Wolf & Saia, 2006). Various studies show the use of dynamic network structures, Kossinets & Watts(Kossinets & Watts, 2006) used an email dynamic network between students, faculty and staff of a university to understand the dynamic interaction of the relations. Katsaros et al. (Katsaros et al., 2009) have later examined the structure of ad-hoc wireless networks. Moreover, there are different approaches developed to solve various issues in a wide range of applications. These issues range from disease modelling (Eubank et al., 2004) to cultural and info transmission(Kempe, Kleinberg, & Tardos, 2003), intelligence and investigation (S. Li et al., 2004), preservative biology and interactive ecology (Lusseau & Newman, 2004). Moreover, recently Dakiche et al. (Dakiche, Tayeb, Slimani, & Benatchba, 2019) describe a detailed survey of community evolution in social dynamic networks.

Many businesses are using social media networks for tasks such as group buying, social marketing and social media marketplace. The social media network systems used for e-commerce have a large amount of time-dependent relational data. Liu and Yang (X. Liu & Yang, 2012) employing a pattern mining approach develop a framework of business intelligence to support dynamic data analysis for such relational social media systems. Gupta et al. (A. Gupta, Thakur, Jain, & Garg, 2015) introduced a framework for mining patterns of a dynamic network containing directed, undirected, weighted-directed and unweighted-undirected graphs. Later in 2016, they proposed another technique(A. Gupta, Thakur, Garg, & Garg, 2016) of mining periodic patterns in weighted-directed dynamic networks. The social dynamic networks are a source for event identifications as they produce a large amount of temporal networks information data representing posts and discussions of the users. Alkhamees and Fasli(Alkhamees & Fasli, 2016) developed a method for event detection in the streams of social network data. They extended their work (Alkhamees & Fasli, 2019) and introduced

Dynamic-FPM to identify temporal events from data streams using a frequent pattering mining method. Kostakis and Gionis (Kostakis & Gionis, 2017) studied the problem of mining duration-based interactions in dynamic graphs. Periodic interactions have an important meaning in dynamic graphs and sub-graphs. Halder et al. (Halder, Samiullah, & Lee, 2017) proposed a single pass super graph-based periodic pattern mining SPPMiner method to identify such periodic interactions. Though pattern mining approaches offer an attractive awareness about the evolution of dynamic networks, however, its ever-evolving nature suggests a limitless space for upcoming researchers.

3.4 Measures of Dynamic Network

Network properties elaborate on various network measures. We analyse these network measures to understand the change in the graphical properties of a network over time. We use these properties to present that the networks are not trivial and evolve, and to stimulate the analysis randomly. Other than, the properties based on graph theory like shortest path and degree, researchers have also developed centrality measures to realize the importance of a node in the network (Wasserman & Faust, 1994). Different mathematicians and physicists have introduced different properties of dynamic network graphs such as connectivity, cluster coefficient, adjacency matrix, degree distribution, transitivity etc.(Albert & Barabási, 2000; Erdős & Rényi, 1959; Newman, 2003). Different computer scientists have also proposed various measures of dynamic networks, for example, page rank introduced by (Langville, Meyer, & FernÁndez, 2008); tensor analysis factorization presented by (Asur, Parthasarathy, & Ucar, 2009b) algorithmic properties of the network such as routing by (J. Kleinberg, 2000). There are different property measures of a dynamic network some important measures are: (1) The connectivity is an important measure used to find the shortest path between nodes of the network, (2) Density of local and worldwide network represent the connection between the virtual numbers of nodes are links between them and (3) Spectral measures of the dynamic graph represent the matrix of a network. The details about different measures of dynamic networks are given below:

3.4.1 Connectivity

The connectivity in the network is an important aspect of network data because it is used to get the shortest length distribution between different nodes for all directed networks and undirected networks as well. Latora & Marchiori (Latora & Marchiori, 2001) have given a competence measure to the shortest path between the node and other researchers (Jure Leskovec et al., 2005; Jure Leskovec, Kleinberg, & Faloutsos, 2007) have used the notion of the conservative graph-theoretical background of network diameter and 90th percentile to calculate the shortest path length distribution.

3.4.2 Density and Clustering Coefficient

Bienenstock, Bonachich and Oliver (Bienenstock, Bonacich, & Oliver, 1990) state that, It is very important to count the comparative number of nodes and connections between them to get a better sense of the availability of nodes in the network, locally or globally regardless of the other properties network graph. Some network measures are not very important to measure in a network but the clustering ratio (Density) is very important to calculate to understand the structure of the network. Watts and Strogatz(Watts & Strogatz, 1998) introduced the clustering coefficient of the density of network graphs. Moreover, the clustering coefficient describes the average to cluster across all peaks of the network.

The clustering coefficient of a node correlates with the degree of the node because nodes with a high degree will have valuable edges between them and will have a small clustering coefficient(Soffer & Vazquez, 2005). Densification power law (DPL) has obtained an important curiosity in the field of computer science. Leskovec et al.(Jurij Leskovec, Chakrabarti, Kleinberg, & Faloutsos, 2005; Jure Leskovec et al., 2007) gave the idea of DPL in two papers Then different researcher worked in this field (Dikaiakos, Katsaros, Mehra, Pallis, & Vakali, 2009; Grus, Shi, & Zhang, 2007; Latapy, Magnien, & Del Vecchio, 2008; Menezes, Ziviani, Laender, & Almeida, 2009). The DPL follows equation (3). Where E represents edges and V represents vertices or nodes and α is densification exponent.

$$E(t) = k \cdot V\ (t)\ \alpha \qquad (1)$$

Where $1 < \alpha < 2$ is called the densification exponent.

3.4.3 Adjacency matrix

Conceivably, the easiest way to show the network is utilizing an adjacency matrix (Newman, 2004). Let us suppose that there are n vertices in the network that are joined to each other in some manner, via m edges. Furthermore, let these edges be directed or undirected. We can also show the linking structure using the following matrix formation.

A_ij=1 if there is an edge joining vertices i,j: 0 otherwise

This style of illustration is not just limited to the case of undirected graphs; however, it stretches to various sorts of networks. Regardless of its comparative ease, the adjacency matrix is fairly an influential tool while analysing networks, in such a sense that a lot of information encrypts in it.

3.4.4 Spectral properties

The algebraic structure of the graph matrix is used to develop spectral properties, it depends on the events and evidence about the construction of the graph (Biggs, 1993; Chung, 1997). The spectral properties of a network graph answer different questions related to eigenvalues associated with the nodes of the network. The page ranks is an example of the spectral properties of a graph (Brin, Motwani, Page, & Winograd, 1998).

3.4.5 Degree distribution

One of the utmost significance and broadly considered metrics of network structure is the degree distribution (Newman, Watts, & Strogatz, 2002). As deliberated, the degree ***k_i*** is the number of edges associated with any node **‘*i*'**. If we are to contemplate the network as a whole, then a supplementary practical method is to look at the segment of vertices of a graph network having k degree. We call this the degree distribution ***P_k***. Instead, ***P_k*** represents the likelihood of an arbitrarily selected vertex to have precisely **‘*k*’** edges associated with it. The degree distribution is beneficial for several motives. From an applied stance, most mathematical models built on it are comparatively straightforward and one can make exact calculations. In rapports of personifying networks in the actual world, the simplest thing to measure is the degree of a vertex, and from that one can simply create a histogram to characterize the degree distribution. Possibly most prominently, the degree distribution is an exceptional indicator or monitors a network’s characteristics together in a relation to its topology and its dynamics as well(Newman, 2003).

3.4.6 Transitivity

Transitivity is also a very significant structural property in dynamic network analysis it refers to the extent to which two nodes connected by an edge are transitive(Wasserman & Faust, 1994).

3.5 Applications of Dynamic Networks

Different applications from several domains use networks as basic prototypes to define relationships between different entities. Over the last two decades, a variety of networks has been studied by researchers. Cortes and Pregibon (Cortes & Pregibon, 1998) used a phone calling network, and it was the most frequently used network at that time. Brin and Page (Brin & Page, 1998) studied the structure of hypertextual search engines and this is an example of web page linking networks. Co-authorship and citation networks were used by Lawrence, Giles and Bollacker (Lawrence, Giles, & Bollacker, 1999) for digital libraries and autonomous citation indexing. The same co-authorship and citation networks were employed by Koren, North, and Volinsky (Koren, North, & Volinsky, 2007) for measuring and extracting proximity graphs in networks. Schwikowski, Uetz, and Fields (Schwikowski, Uetz, & Fields, 2000) used a structure of the protein to protein network connections, later after a few years in 2007, the same network was employed by Whorton et al. (Whorton et al., 2007). Pei et al. (Pei et al., 2004)used transcription regulatory networks in the yeast cell cycle. Diesner, et al, (Diesner, Frantz, & Carley, 2005) and Chapanond et al.(Chapanond, Krishnamoorthy, & Yener, 2005) exemplify the Enron email network through spectral based analysis and graph-theory methods on email corpus. Pattern mining infrequent dynamic subgraphs (Borgwardt et al., 2006) is also an example of the Enron email network. Park and Pennock (Park & Pennock, 2007) have applied a collaborative filtering method using a collaboration of nodes, for ranking and browsing of movies in the IMDB database efficiently. Lately, Boyd and Ellison (Boyd & Ellison, 2007) attempted to contextualize friends’ networks of MySpace, Facebook, Cyworld, and Bebo in their study.

Various application in different domains leads to the development of a variety of dynamic networks. These network groups and the underlying spheres assist as the foundations of dynamic networks to confirm the applicability moreover; they assess the performance of the proposed methods as well. The variety of these networks is based on their size, interactive density, and time steps to assist the appraisal of diverse features of suggested approaches (R. Ahmed & Karypis, 2012). Recently a survey by Shi et al. (C. Shi et al., 2016) describes the different applications of dynamic multi-typed network data. Following are a few applications of dynamic networks:

3.5.1 Social networks

Social communicative networks express the communication of information between different units of a set during a specific interval of time. The social-communicative networks include email networks, Internet traffic, and telephonic call networks. These networks represent different entities and their properties using nodes and labels. Moreover, directed edges indicate communication of information, between the units of networks. As far as, edge-labels are concerned they are the representative of the data (information) that is being transferred, (R. Ahmed & Karypis, 2015a). Holme, Edling, & Liljeros (Holme, Edling, & Liljeros, 2004) have used an online dating social network Pussokram to find the comprehensive antiquity of user connections. This network is rather infrequent since it encompasses complete chronological evidence about manifold methods of user communications. For instance, users can, specify friendship relations or can send and receive messages flirt with other users. All of these interactions will have different networks in structure. They concluded during the research of 400 to 500 days' timestamps that the result of all interaction starts from a single node with the increasing average degree of the network and rapidly the degree reached a constant value. The authors reported their results on the aggregate network of all interactions and friendship links alone since the other trends are very similar. They also calculated the clustering coefficient of the network and empirically reported that all the connections in the network show a decreasing (Albert & Barabási, 2002) clustering coefficient. This network of dating sites also did not show any long-term behaviour of the links between users. Hu and Wang (Hu & Wang, 2009) investigated the chronological progression of a dynamic social network in china called, Weblink. They used 10000 nodes and 2000000 edges dataset during the time stamp of 27 months. Their data set showed an S-shaped tendency of the features of the network, just like a logistic function between the number of nodes and edges over time.

3.5.2 Trading Networks

Friedman (Friedman, 2005)in the brief history of the twenty-first century, has presented an imaginary trading network between countries across the globe, which highlights the eminent spectacle of the invention due to globalization. He divided and shared his merchandising network into 17 diverse components and different states. These networks represent the trading and exchange of things between a group of different units or objects. According to their proposed model, the vertices represent the business units that may base on states, businesses, countries and individuals whereas the directed edges represent the transfer of goods (relations) from one of these units to the other. The label of the node is representative of the facts regarding several problems and solutions relative characteristics of the units as like governmental structure, gross domestic product and production group. On the other hand, the edge labels are indicative of various features defining the goods traded such as goods, their types and quantity. These complex networks are dynamic due to the ever-changing trading partners and the traded goods over time that leads to the creation of a series of dynamic network datasets each expressing the trading or exchange process over a specific time interval (R. Ahmed & Karypis, 2012).

3.5.3 Authorship Networks

Authorship Networks includes various inter-related links, developed during designing citation dataset. Such networks may also include many other networks like co-authorship, co-conference, and co-posting. In such networks, the node and edge labels are indicated through the information about the characteristics of the networks' units, about the co-authored content or publication sites, and include numerous dynamic networks such as, co-posting, co-authorship, and co-conference that have been formerly studied by machine learning and data mining. Liben-Nowell (Liben-Nowell & Kleinberg, 2007) while investigating the large co-authorship networks to answer their link-prediction problem, suggested that information about future interactions can be extracted from network topology alone and that objectively elusive measures for detecting node proximity can overtake more direct measures.

To analyze the concept of the growing network in the co-authorship graph of SIGMOD conference Nascimento (do Nascimento, 2003)proposed an approach to find the regular path lengths and largest length in the network. Various researchers show the use of authorship datasets to analyse changes between different time windows using structural and statistical properties. For example, Soffer et al.(Soffer & Vazquez, 2005) used a publication dataset of Neuroscience Journal of Mathematics. Recently, (R. Ahmed & Karypis, 2015b) used DBLP dataset containing relations from 1958 to 2012 based on CS publications.

3.5.4 Citation Networks

The networks founded on the references that appear in various publications are called reference (citation) networks. These networks are directed in nature. For example, the nodes of the network can refer to a single patent or multiple patents but the patent cannot refer back to the node. Barabasi et al. (Barabâsi et al., 2002) were the pioneer of finding direct path length in a declining degree network, and their proposed approach is against the research by (Newman, 2001). The research of Elmacioglu and lee (Elmacioglu & Lee, 2005)inspected the network of publications in the DBLP database to understand the community structure of the publication's settings. According to them the average shortest path between the nodes of the evolving publication network was one. Menezes et al. (Menezes et al., 2009)who used the citation network of faculty publications in Europe, Brazil and North America closely relate the results of this study to the research.

3.5.5 Relational Pattern-based change mining

We categorize relational networks into simple or complex relational networks based on the edge and node sets. Simple Relational networks are very small and can have only one vertex and edge. These kinds of small networks define comparatively static data sets of units and relations, while big networks define complex graph structures associated with complex relations. The big networks have massive properties, containing sets, vectors, and a combination of sets. Relational pattern mining based approach to mine changes focuses on the characterization of changes in a network at a local scale. In this approach, the evolution of the network is divided into temporal patterns of graphs, while the state of the network is vertices and transition among those vertices are change patterns (Loglisci et al., 2015).

The systems based on these networks differentiate networks as dynamic or static network systems. For a network to be dynamic, the entities and the relations between them need to be changing or evolving. The dynamic characteristics in complex network systems develop through the extensions of static network models, into a sequence of networks. For instance, (Borgwardt et al., 2006) have used this technique to mine frequent subgraphs and (Tang et al., 2008) has detected change in communities using multi-mode networks, where dynamic networks are representative of the changes of the entities and their relations with time. There are two variations of change mining using relational pattern-based approaches these are:

- change mining in static networks
- change mining in dynamic networks

3.5.5.1 Change Mining in Static Networks using a relational patterns-based approach

Many studies consider the concept of mining relational patterns in static networks. Inokuchi, Washio, & Motoda(Inokuchi et al., 2000b) discovered an approach named Apriori Graph Mining (AGM) to efficiently mine the association rules among the frequently appearing substructures in a given graph data set, represented by an adjacency matrix, where matrices were mined through the extended algorithm of the basket analysis. AGM approach begins with a single node subgraph and evolves recursively using a breadth-first approach. The breadth-first search (BFS) approach is also used by the Frequent Sub-graph (FSB) algorithm to find all recurrent and randomly connected subgraphs (Kuramochi & Karypis, 2004a, 2004b). FSG algorithm has attained a substantial improvement in runtime as compared to AGM through the introduction of an efficient canonical classification arrangement, competent pattern growth algorithm, and proficient frequency counting method. These approaches utilize high memory space for massive datasets. In comparison, BFS offers a solution for the complete graph in a very fast manner.

The gSpan (Huan et al., 2004) algorithm finds all recurrent and randomly associated subgroups of a graph by using a depth-first traversal (DFT), it also uses a well-organized and established tagging system constructed on (DFT). As like, gSpan, some other depth-first search (DFS) approaches are capable of deriving all frequent arbitrary connected subgraphs from graph databases, such as Efficient mining of frequent subgraphs in the presence of isomorphism (Huan, Wang, & Prins, 2003), mining maximal frequent subgraphs from graph databases (Huan et al., 2004), and frequent structure mining (Nijssen & Kok, 2004). Usually, these algorithms get motivations from different Dynamic Frequent Subgraphs (DFS) based arrangement and tree mining algorithms. The algorithm by Nijssen (2004) appears to be the most agile algorithm due to the fine segregation of the problem space.

Furthermore, a collection of algorithmic methods concentrates upon extracting frequent patterns in a large single graph using the data mining approach in such a way that each subgraph has t embedding in that large graph. SUBDUE by (Holder, 1994) appears to be the initial and renowned algorithm to find repeating 20 small graphs in a single large graph. Moreover, hSIGRAM and vSIGRAM (Kuramochi & Karypis, 2004a) methods extract almost all recurring subgraphs in a complex graph based on a sparse matrix.

3.5.5.2 Change Mining in dynamic Networks using a relational patterns-based approach

Dynamic networks are a newly introduced significant research area, so we have considerable research on describing important frequent structural patterns and development of algorithms for their documentation and classifications. Different researchers have used this method in detecting changes Table 3.1 Give a summary of some studies using the relational pattern-based approach to detect changes.

Study	Features of the study.
Agrawal, Imieliński, & Swami (1993)	They proposed an algorithm that produces important association rules between items. The Algorithm incorporated buffer management and novel estimation and pruning techniques.
Dong & Li (1999)	They proposed the idea of Emerging Patterns (EPs) for knowledge discovery from large-scale timestamped databases. Promoted the description of the large collection of datasets using their borders and introduced the EPs mining algorithm.
Zhang, Dong, & Kotagiri (2000)	Developed a constraint-based EP miner using two types of constraints external constraints and inherent constraints.
Li & Wong (2001)	They introduced Emerging Patterns(Eps) mining to develop accurate and efficient classifiers.
Lisi & Malerba, (2004)	1. A novel approach relies on hybrid language to find link-based rules with multiple levels of description granularity. 2. AL-Log language was used which is a treatment for relational as well as structural data.
Ceci, Appice et al. (2007)	They solved the problem of extracting EPs from spatial data. Their approach deal with the complexity of discovering emerging patterns from spatial data in two cases i) spatial properties and relations ii) data affected by autocorrelation.
Liu, Yu et al. (2008)	1. The technique to discover the subgraphs that show important changes in the evolving graphs. 2. They formalize the problem of changing regions related to actual changes and designed an algorithm to identify the changes in the subgraphs.
Berlingerio, et al. (2009)	They introduced graph-evolution rules, a novel type of frequency-based pattern that describe the evolution of large networks over time, at a local level.
(Bonchi, Castillo, Gionis, & Jaimes, 2011)	Consider key problems and techniques in social network analysis and mining from the perspective of business application. They discussed data acquisition, preparation, trust, expertise community structure, network dynamics and information propagation in particular. Moreover, gave an overview of the current techniques providing a critical perspective on business applications of social network analysis and mining.

(Birand, Zafer, Zussman, & Lee, 2011)	Investigated the change in dynamic evolving node mobility network structures of mobile graphs. The proposed approach used Levy Walk mobility to obtain interesting patterns from the evolving graphs.
(Lewis, Gonzalez, & Kaufman, 2012)	This study combined stochastic actor-based modelling and self-reported data to address the problem of social change selection and peer influence.
(Casteigts, Flocchini, Quattrociocchi, & Santoro, 2012)	Developed and used time-varying graphs (TVGs) framework to study the evolution of network properties.
(X. Wu, Zhu, Wu, & Ding, 2013)	This research proposed the HACE theorem to explore complex and evolving relational patterns among network Big datasets.
(Loglisci & Malerba, 2015)	This paper investigated the problem of periodic changes based on the notions of emerging patterns and repeating changes by capturing statistically evident changes and their evolutionary period.
(X. Ma & Dong, 2017)	The research proposed a semi-supervised evolutionary nonnegative matrix factorization framework to detect dynamic community patterns.
(Trivedi, Dai, Wang, & Song, 2017)	Presented Know-Evolve deep evolutionary network that learns non-linearly evolving entities relationships over time.
(Rossetti & Cazabet, 2018)	This research study is a literature survey to present features and problems of dynamic community discovery and propose future directions in the field of evolving network community detection.
(Z. Zhao, Li, Zhang, Chiclana, & Viedma, 2019)	This paper introduced an incremental method by employing subgraph-joins to detect relational communities patterns from social evolving networks.

Table 3.1 Change mining in dynamic networks using the relational pattern-based approach

Desikan & Srivastava, (2004), analyzed the importance of mining temporally evolving Web graphs. However, he did not present any algorithmic solution to perceive the stable patterns and their progression. Shoubridge, Kraetzl, WALLIS, & Bunke, (2002) presented two methods to identify regions of significant change. Moreover, they surveyed various sensitive graph distance measures to detect abnormal changes between two graph snapshots. (Besson, Robardet, Boulicaut, & Rome, 2005; Robardet, 2009)exemplified the recurrent patterns of a graph as pseudo-cliques and projected an algorithm that firstly derives each graph snapshot of a dynamic graph for local patterns and then syndicates them with outlines from the previous snapshot having some limitations, to form evolving and dynamic patterns. They use two self-motivated device networks and a vibrant mobility network to calculate the algorithm.

Borgwardt et al. (Borgwardt et al., 2006) familiarized the notion of the dynamic subgraph, with the extension of the approach to use the sequence of subgraphs based on successive snapshots of a dynamic network. They also introduced a novel algorithm to identify regular patterns in dynamic data. Next year, Jin et al. (Jin, McCallen, & Almaas, 2007) presented the idea of a network motif, which is a connected-induced graph of a network. Moreover, they concentrated upon the issue of discovering frequent patterns in dynamic, temporal networks where a series of patterns relate with each node but its weight and the topology of the network remains the same. Lahiri and Berger-Wolf(Lahiri & Berger-Wolf, 2008)introduced a mining approach to find periodic subgraphs in the dynamic networks using two associated features, recurrent patterns and periodic patterns. The proposed algorithm applied to dynamic networks like ENRON and IMDB celebrities' datasets predicted the periodic behaviour in future. Berlingerion et al (Berlingerio et al., 2009) proposed an algorithm to sense recurrently connected subgraphs in time-evolving graphs to formulate graph-evolution rules for the satisfaction of a minimum confidence limit. They assumed as the graph grows the nodes and edges only increase and never delete. They used Flickr, Y! 360, DBLP and arXiv datasets to calculate the algorithm and gather more exciting rules.

Protein-Protein Interface network data from yeast and a time series of yeast gene manifestation levels were used in the study of (Wackersreuther, Wackersreuther, Oswald, Böhm, & Borgwardt, 2010) to develop a framework to perform frequent subgraph discovery in dynamic networks. This methodology is identical to (Borgwardt et al., 2006) in its use of suffix trees. The algorithm first discovers regular subgraphs in the combination graph of a time series of graphs and then examines the resulting static frequent subgraphs for recurrent dynamic patterns.

Duan et al. (Duan, Li, Li, & Lu, 2012) introduced an algorithm to resolve community mining in dynamic weighted network graphs (DWDG). This technique first develops compact communities by calculating graph's significance matrix specifying the degree of a node belonging to a community using the Random Walk with Restart technique. Then it combines the compact communities along the course of extreme increment of the modularity. This study is similar to (J. Sun et al., 2007) and both have used synthetic and real-world (ENRON) datasets to calculate their algorithm. Inokuchi et. al (Inokuchi & Washio, 2010) highlighted the problem of deriving recurrent, pertinent, and induced 25 subgraph subsequences, called FRISSs, from graph sequence data. These induced subgraph subsequences control the changes of a subgraph over the subsequence.

3.6 Related Work

Network dynamics analysis has become an important area of research as it is a well-known fact that networks evolve and adapt with time. Early change mining problems focused on the statistical and graph method to detect changes in dynamic networks without considering the heterogeneity of networks in consideration. Moreover, existing methods of network change detection before the introduction of HINs in 2009 were pattern recognition , spectral graph theory, mean or median and diameter of graphs(Gaston, Kraetzl, & Wallis, 2006). A dynamic network can be represented by the time snapshot and each snapshot represents a logical time-stamp of the correlation between individuals of the network system.

Common types of changes in DHINs include gradual change over time, recurring, abrupt, macroscopic (global), community level and microscopic (local). Detection of these changes help us to answer various research questions like; how does the network evolve? and how can we find the suspicious activities in a network(Y. Wang et al., 2017)? For example, Figure 11 represent the structure-based diametric changes in the network at two different timestamps t_1 and t_2. While Figure 12 presents two main types of changes in a temporal snapshot of a dynamic network, some links disappear in timestamp 1 (T1) which impacts the local structure of the network. The timestamp 2 (T2) represents the breaking of the connection between two parts of the network which will result in the global structural change in the network.

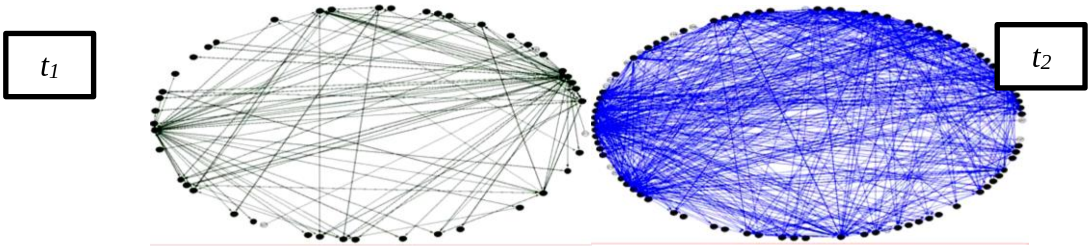


Figure 11 Network before the change and after the change at t1 and t2 (Gaston et al., 2006)

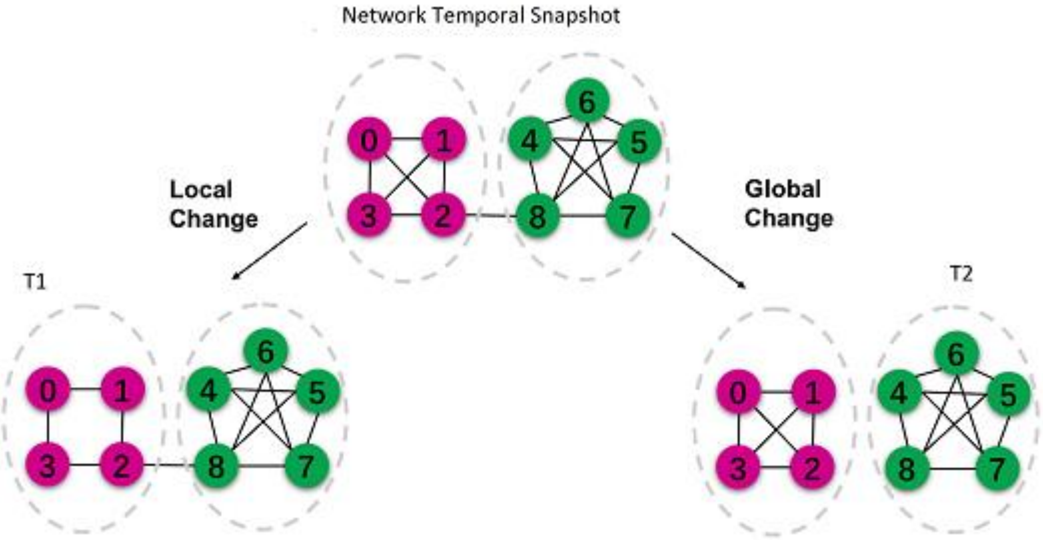


Figure 12 Local and global changes in a dynamic network(T. Zhu et al., 2020)

Change detection has received greater attention from the researcher in recent years and the current related studies in this field can be categorised into two groups: (1) Supervised Methods and (2) Unsupervised Methods.

3.7 Supervised Methods

Supervised methods refer to data mining (machine learning) algorithms or frameworks that are used for classification and prediction based on labelled training data. For example, to classify or predict the sentiment of a piece of text such as tweets and posts is supervised learning(Phyu, 2009). The dynamic network analysis utilising supervised methods can be divided into two subcategories link prediction, and change-point detection.

3.7.1 Link Prediction using supervised learning methods

Link prediction is the process of predicting the presence of an edge between two entities in a network(Juszczyszyn, Musial, & Budka, 2011). The link prediction on dynamic networks is an extensively researched topic from a decade. Juszczyszyn et al.(Juszczyszyn et al., 2011) propose a method Triad Transition Matrix (TTM) containing the probabilities of transition between triads found in the dynamic network, for characterising the dynamics of the complex social network to solve the link prediction problem. Zayani et al. (Zayani, Gauthier, Slama, & Zeghlache, 2012) used the connected wireless networks for link prediction by applying the method of topology dynamicity tracing. Real-world complex systems have underlying structures of evolving networks where nodes and links are deleted and added with time. Short-term and long-term link change prediction is also an important research problem. Bliss et al. (Bliss, Frank, Danforth, & Dodds, 2014) introduced the Covariance Matrix adaptation evolution strategy (CMA-ES) to predict short-term and long-term link prediction. Similarly, Chen et al. (K.-J. Chen, Chen, Li, & Han, 2016) proposed a link prediction that can learn the long-term evolution of the dynamic graph network.

Temporal information of a network for instance community structure and node centrality plays an important part in the structural development of a dynamic network. Ibrahim, Ahmed and Chen (Ibrahim & Chen, 2015) employed the node centrality measure to predict the link evolution in the dynamic networks. Later, they(N. M. Ahmed et al., 2018) introduced the DeepEye framework to predict links in dynamic networks employing non-negative matrix factorization. The dynamic networks are very huge and Zhu et al. (L. Zhu, Guo, Yin, Ver Steeg, & Galstyan, 2016) proposed a temporal latent space method for link prediction in the large sequence of graph snapshots. Moreover, Li et al.(T. Li, Wang, Jiang, Zhang, & Yan, 2018) using Boltzmann machine-based approach; Chiu and Zhan(Chiu & Zhan, 2018) employing deep learning-based method; and Yang et al.(M. Yang et al., 2019) utilising a generative model developed different approaches for dynamic link prediction.

All the supervised learning approaches for link prediction discussed in the above two paragraphs consider that dynamic network is single typed and dynamic but real-world network systems are multi-typed, dynamic and heterogeneous. Recently, research studies by Kong et al. (Kong, Li, Zhang, Zhu, & Liu, 2019) and Jia et a.(Jia et al., 2017) have considered the multi-type and heterogenous properties of dynamic real-world networks to predict links evolutions.

3.7.2 Change detection using Supervised learning methods

Research related to dynamic network analysis can be classified into two categories i.e anomaly detection and change detection(Ze Li et al., 2017). Change detection is significant because the evolution of the networks has hidden information present and each state of the network. For example, Figure 13 shows the evolutionary changes in a social network. The transition from time step t to t+1 introduces two new links. From time t+1 to t+2 edges are disappeared from nodes A and C. Anomaly detection helps to discover an abrupt and sudden temporal change in a dynamic network. Some important research approaches(Heard, Weston, Platanioti, & Hand, 2010; Kaur & Singh, 2016; Peel & Clauset, 2015; Silva & Willett, 2008; Vigliotti & Hankin, 2015) in the past have used supervised learning methods to find anomalies in a dynamic network.

Change detection and anomaly detection algorithms or frameworks developed on supervised learning methods consider the network as single typed dynamic networks. Raghavan et al. (V. Raghavan, Galstyan, & Tartakovsky, 2013) developed an event-based Hidden Markov Model (HMM) framework to detect the activity profile of terrorist groups. The change detection is classified as active and inactive using their framework. Moreover, Asur, Srinivasan, & Ucar (Asur, Parthasarathy, & Ucar, 2009a) proposed an event-based framework for depicting the evolutionary behaviour of relational graphs. They used nonoverlapping temporal snapshots of interaction graphs and used time-varying events to characterize complex behavioural patterns of individuals and communities over time.

Some other studies also focused on applying supervised machine learning methods to classify event-based change behaviours. PBCS(Xue et al., 2011) classification algorithm for event classification used association structures to find context-based organizational behaviour. This algorithm and some other methods (Subrahmanian, Mannes, Roul, & Raghavan, 2013; Subrahmanian et al., 2012) was used to find the terrorist organization change behaviour. Li et al. (Ze Li et al., 2017) used a neural network to detect event-related changes in organisational dynamic networks. Recently, Bian et al. (Bian, Koh, Dobbie, & Divoli, 2019) developed a change2vec algorithm which divides the dynamic network as snapshots of networks with different time stamps. This model embeds changes between two consecutive static networks by capturing the newly added nodes.

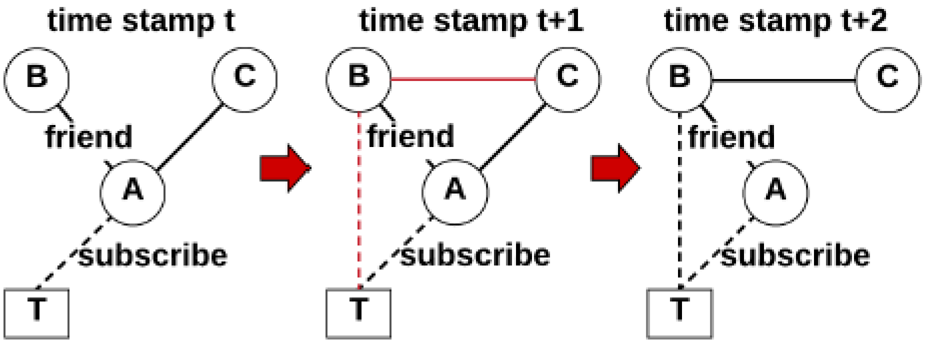


Figure 13 Dynamic social network changes in different time stamps(Bian et al., 2019)

3.8 Unsupervised methods of Change detection

In contrast to supervised change detection methods, unsupervised learning methods and frameworks need no prior information about the network to detect changes in the dynamic network. Unsupervised methods to detect changes in a dynamic network are divided into two categories, (1) Generative methods and (2) feature-based methods(T. Zhu et al., 2020).

3.8.1 Generative Methods of Change discovery

The generative methods of change detection employ the probabilistic framework to find changes. In this method, latent space is used to detect changes and the differentiating point of generative models are the type of model used and the methods for change detection. Various generative models have been introduced to find changes in dynamic networks(Y. Wang et al., 2017). These models help to detect gradual and abrupt changes in the dynamic network. This section elaborates the comparative analysis of different notable algorithms based on the generative method of change detection.

3.8.1.1 Generalized Hierarchical Random Graph (GHRG)

Peel and Clauset (Peel & Clauset, 2015) proposed the GHRG model to detect change points capturing assortative and disassortative community patterns and the community structures. GHRG method models a network G=(Agarwal et al., 2012) where V are vertices and E are the edges of the network. This method decomposes the N vertices into a sequence of nested clusters whose relationships are characterized by a dendrogram. For example, Figure 14 represents a temporal snapshot of the Enron email dynamic network in its equivalent GHRG cluster dendrogram. The vertices (V) in the networks graph (G) are leaves of the dendrogram while the parameter p_r shows the probability based on the connectivity of the two vertices *u* and *v*. The generalized hierarchical random graph (CHRG) is a graph that has focused on hierarchical community structures. It supports organizing the network as a tree and tree leaves represent nodes in the network while the internal nodes correspond to communities. The good thing about this model is that it encloses many features of the other models (e.g. (Aicher, Jacobs, & Clauset, 2015; Holland, Laskey, & Leinhardt, 1983; Jurij Leskovec et al., 2005; Moreno & Neville, 2013; Nowicki & Snijders, 2001)) for change point detection and shows better results from them with its two distinctive features:

1. The First feature is the interpretation which compactly models nested community structure at all levels in a network and provides an interpretable output for later analysis.
2. The second feature is that as the network evolves, it adapts the dendrogram structure naturally to fit the network, adding or removing levels in the hierarchy.

The model parameters are estimated from a window of various networks of varying sizes. The model examines whether and when a change point occurred, they employ the generalised likelihood ratio test. Consequently, this method is not performing an explicit mapping from the network to anomalies. However, it provides a p-value that quantifies the confidence of the detecting results and that support the changes in patterns of the nested community structure may give an intuitive explanation of how precisely change points have occurred. Besides these benefits, the CHRG has some limitations such as the bootstrapping from the generative model for p-value calculation makes it a time consuming and highly complicated approach. Due to bootstrapping and not using the divide and conquer method the complexity of this model is exponential $O(2^N)$. Moreover, it is also complicated to fit data to the specific model requires discarding other possible choices.

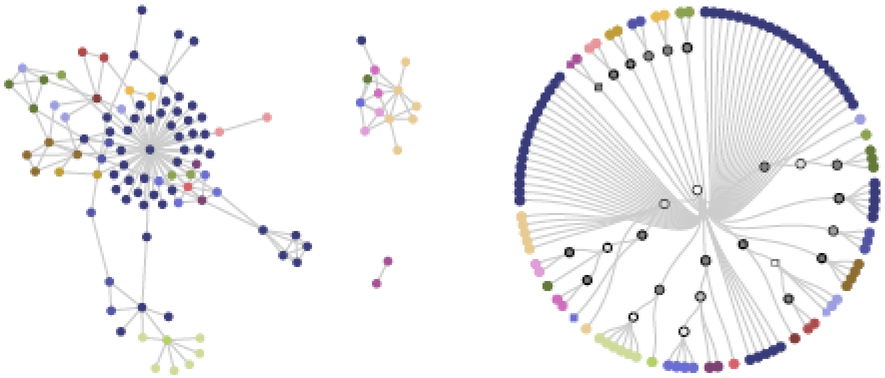


Figure 14 A Temporal snapshot of Enron email and its GHRG dendrogram cluster(Peel & Clauset, 2015)

3.8.1.2 EdgeMonitering

Wang et al. (Y. Wang et al., 2017) introduced EdgeMonitering framework of change detection. EdgeMonitering is an edge probability estimation based change point detection algorithm on synthetic and real-world datasets. This framework is simple and uses the following steps to detect changes:

1. Extract a feature vector from each temporal snapshot using joint edge probability
2. Quantify the dissimilarity between a consecutive temporal snapshot of the network employing Kolmogorov-Smirnov statistics, Kullback-Leibler divergence and Euclidean distance
3. Show the change point when the dissimilarity score is above a specified threshold based on the permutation test method.

It makes zero assumptions on the concrete form of the generative model and this feature differs it from GHRG. Therefore, it can detect not only a change of parameter values for a given model but also a change in the model type. There are also some similarities between GHRG and EdgeMonitoring methods. For instance, the parameter estimation will be done within time-slicing windows. Subsequently, it quantifies the variations between consecutive snapshots by comparing the estimated model and flags out change facts by using the permutation test on the sequence of the dissimilarity score. Further, it tracks the presence or absence of the edges by introducing a conditionally independent two-state. The complexity of this method is O($\bar{M}T$), where $\bar{M}$ is the averaged number of edges in each snapshot and T is the number of snapshots. Markov chain. Since this method uses Kolmogorov-Smirnov (KS) statistics, Kullback-Leibler (KL) divergence for comparing distribution across consecutive windows. Both of these methods have good quality but large sample bootstrap from each window makes this method extremely slow(Y. Wang et al., 2017).

3.8.1.3 Size agnostic change point detection (SizeCPD)

Recently Miller and Mokryn (H. Miller & Mokryn, 2020) presented SizeCPD a framework for change point detection in dynamic networks. It does not model the snapshots with a specific generation model (e.g. hierarchical random graph or Stochastic Block Model). However, it characterizes the snapshots using degree distribution measures. This method uses the following steps to find changes:

1. Take a series of networks (graphs) where the changes in the generative model occur.
2. Compute the cumulative distribution function of the degrees(CDF) for each graph
3. Compute the distance using a non-parametric two-sample test i.e Kolmogorov-Smirnov (KS) and perform hypothesis testing to infer the change between two CDF's.

The SizeCPD shows better results as compared to GHRG model which is failed to identify some of the change points. It found 13 out of the existing 14 points of change in the data and achieved recall and precision of 0.9 when real dataset (Enron) utilised and this framework outperforms previous solutions. Moreover, SizeCPD can be used for sliding windows over several snapshots as is used in GHRG to compute the cumulative degree distribution. This method also uses Kolmogorov-Smirnov (KS) statistics for distance measuring which makes it slower on large datasets.

3.8.2 Feature extraction methods

In contrast to the generative methods that compare the dynamic network snapshots in the estimated model latent space, the feature-based methods calculate the difference of the timed network snapshots directly. The most important building blocks of feature-based methods are their anomaly detection and unsupervised feature extraction strategies.

3.8.2.1 DeltaCon

DeltaCon (Koutra, Shah, Vogelstein, Gallagher, & Faloutsos, 2016) algorithm is an extension of (Koutra, Vogelstein, & Faloutsos, 2013) and it computes the similarity between two graphs with the same set of vertices. This algorithm uses a compound graph feature-based similarity computation technique to find change points in dynamic networks. DeltaCon find changes in the dynamic network by using the following steps:

1. Use two graphs G1 and G2 with the same set of Vertices set V but different edge set E1 and E2.
2. Find the similarity score between the two input graphs if the similarity score result in 0 value it means that the input graphs are different otherwise the graphs are identical if the similarity score is 1.

DeltaCon uses fast belief propagation (Koutra, Ke, et al., 2011) based on sociological theories to obtain the matrices of pairwise vertex affinities to summarize networks. Essentially, it reflects network connectivity, containing far more data than the adjacency matrix since it captures 1-hop, 2-hop and higher-order neighbourhoods. This algorithm determines the time series of the consecutive snapshot similarities to detect irregular snapshots of the graph sequence and employs the quality control with individual moving range on it. The lower control limit below which the corresponding time points are recorded as change points, i.e. they vary "too much" from the previous and subsequent snapshots, is thus established. It specifies the lower control limit below which the corresponding time points are reported as they differ from the previous and following snapshots(T. Zhu et al., 2020).

3.8.2.2 Community Identification based Change-point Detection (CICPD)

Zhu et al. (T. Zhu et al., 2020) introduced a new structural feature extraction method and CICPD model which uses summarization techniques to efficiently classify the temporal snapshots of a network into distinct patterns based on the extracted features. The change detection in CICPD framework is based on the following steps:

1. Construct a graph network whose vertices shows the dynamic network temporal snapshots.
2. Find the node importance features using the PageRank algorithm that captures local structural properties.
3. Describe the network with the probability distribution of the node's features which represents global structural information.
4. Use Jesen-Shannon divergence to compute the distance of temporal network snapshots.
5. Do community detection on the constructed network in step-4 and serialize community detection results in sequential order.
6. The sequential order results obtained in step-2 indicate potential changes in the network.

There are four variants of CICPD framework:

- CICPD-PR (Community Identification based Change-point Detection - Page Rank)
- CICPD-DD(Community Identification based Change-point Detection-Degree Distribution)
- CICPD-Edge (Community Identification based Change-point Detection- Edge)
- CICPD-NA ((Community Identification based Change-point Detection- Node Affinities)

This change point framework extracts effective features and scales well but does not consider the multi-typed and semantic information of the dynamic heterogeneous information network as it is only developed for single typed networks. Compared to the generative methods of change detection this community-based method also lacks interpretation.

3.9 Rule-Based change detection

Loglisci et al.(Loglisci et al., 2015) introduced a rule-based algorithm to detect changes in a dynamic heterogeneous information network. This algorithm considers the problem of detecting changes in the same subgraphs across different time windows using the structural updates of the network instead of the insertion and deletion of network nodes. The computational solution to detect changes proposed in this algorithm involves the following steps:

1. Divide the same subgraphs data in different time snapshots or a sequence of *n* observations (O_1,….O_n) with width *w* of the periods to create bipartite graphs.
2. Discover a set of temporal frequent relational patterns from each deductive database built on target objects and Non-target objects using SPADA(Lisi & Malerba, 2004) system.
3. Generate candidate patterns based on the minimum support threshold.
4. Use frequent patterns extracted in step-1 to produce change patterns
5. Find the structural dissimilarity between patterns if the dissimilarity is less than the maximum support threshold then it is a stable pattern
6. Generate change chains from both change patterns and stable patterns

This algorithm has been inspired by Loglisci et al.(Loglisci, Ceci, & Malerba, 2013) and uses SPADA system to extract frequent patterns in dynamic data to mine changes in the dynamic network environment.SPADA system discovers multi-level association rules. This system was developed employing Inductive logic programming (ILP) and have been used to mine multi-level association rules in spatial databases and applied to geographic data(Malerba & Lisi, 2001). But we find the following problems, related to this system that has been used to extract frequent patterns from graphs.

- SPADA was considered to extract patterns at different levels of granularity and was used in later studies. However, it also has a scalability issue and it is difficult to apply it on large dynamic data. Moreover, to use the SPADA system we also need to know deductive databases and inductive logic programming (ILP) (Appice, Ceci, Turi, & Malerba, 2011).
- According to the developer of SPADA system (Lisi & Malerba, 2004), the user should be an expert in data engineering and know-how spatial database works.
- Although ILP has been used both in theory and implementation, there is still a gap between correctness, completeness and complexity (Badea, 2001)
- SPADA for the same subgraph to find the change in the same graph at different times observations using the shortest distance between nodes. It does not give any details about the complete graph or patterns related to an individual node and its effect on the subgraph containing a node.

Attributes	**Existing algorithms and Frameworks**					
	CHRG	EdgeMonitoring	SizeCPD	Rule-base	DeltaCon	CICPD
Year	2015	2017	2020	2015	2016	2020
Method	Statistical Probabilistic Learning	Statistical, edge probability feature vector	A statistical, cumulative distribution function	Frequent pattern mining, ARM	Statistical Feature-based Node afinities	Probability Distribution, clustering
Structure of data	Structured Synthetic data	Structured Synthetic data	Structured	An unstructured, deductive database	Structured	Structured
Data sets	MIT proximity network and Enron email network Synthetic Network	Senate cosponsorship network) Enron Email Synthetic Network	AskUbuntu forum And Enron email network Synthetic Network	KEDS DAYS INFECTIOUS DBLP	Enron email network, KKI-42 dataset Synthetic Network	MIT Proximity Network Legislative Cosponsorship Network
Network type	Single-type synthetic dynamic	Single-type synthetic dynamic	Single-type synthetic dynamic	Multi-type Real-world	Single-type synthetic dynamic	Single-type synthetic dynamic
Similarity Measure	Markov Chain Monte Carlo	Kolmogorov-Smirnov Statistics, Kullback-Leibler divergence	Kolmogorov-Smirnov	Neighbour path-based measure	RootED	Jesen-Shannon
Performance Measure	F1-Precision-Recall	Time series analysis, Time efficiency	F1-Precision-Recall	Evolution rules	N/A	F1- Precision-Recall
complexity	Exponential $O(2^N)$	Linear $O(\bar{M}T)$	N/A	$O(n*l_1+n*l_2^2)$	O(g* max{m1,m2}|)	N/A

Table 3.2 Unsupervised change detection algorithms and Frameworks

3.10 Existing Unsupervised Algorithms and Frameworks Summary

There are many methods available in past studies that can detect changes in graph networks. Some of these algorithms and have been developed to detect changes in static networks. But due to the nature of the networks these days we cannot apply them to extract effective changes in the dynamic networks. Studies on dynamic network change detection can also be classified into two categories i.e supervised learning methods and unsupervised learning methods. Supervise learning methods of change point detection require prior information to detect changes in the dynamic network. Supervise methods are further subdivided into link prediction and dynamic change detection. Link prediction algorithms and frameworks are used to predict the presence of an edge between two entities in a dynamic network. On the other hand, change detection is concerned with, how much structural and node-level change has occurred in a dynamic network between two timestamps. Supervised event classification based change detection is also very important and uncover hidden information related to temporal changes in a network.

This thesis only considers unsupervised learning methods as this research sits in this data mining dimension. Unsupervised learning methods in comparison to supervised change detection methods do not require prior information about a dynamic network. There are two types of unsupervised learning dynamic network change discovery methods these are generative methods and feature extraction based methods. Table 3.2 give a detailed overview of existing algorithms and frameworks available to detect changes in dynamic networks. Most of the unsupervised algorithms and frameworks e.g CHRG, EdgeMonitoring, SizeCPD, DeltaCon and CICPD that are available to detect changes use statistical methods to infer the changes. These methods also detect changes in a single type network where the nodes have only one type. In real-world systems, networks are heterogeneous, dynamic, multi-type and complex. The real-world dynamic heterogeneous information networks also have rich semantic meanings. Table 3.2 presents that all the existing studies have used a single type of dynamic network except the rule-based method to detect changes in dynamic networks. A network may consist of three types of data i.e structured, semi-structured and unstructured and we can also see in Table 3.2 none of the existing unsupervised methods has used all three types of data to detect changes in a dynamic heterogeneous information network.

3.11 Research Gap

There are obvious research gaps in the existing unsupervised learning methods to extract changes in dynamic heterogeneous information networks these gaps are:

- Existing methods have considered dynamicity of the network but did not consider the heterogeneity of the DHINs except rule-based change detection method
- Existing methods of change detection do not consider the multi-type property of the DHINs
- Existing methods do not support the semantic reasoning of DHINs.
- Existing methods do not use meta-path based pattern extraction techniques
- None of the existing frameworks detects all three levels of changes (Local, global and community level) in DHINs
- There is no existing framework that is employing both the knowledge engineering process and data mining method to detect changes in DHINs.

This research will fill these research gaps by introducing a novel framework that uses three types of data (structured, unstructured and semi-structured) to construct DHIN. Develop temporal snapshots of knowledge graphs and use these knowledge graphs to detect changes at three levels using data mining methods.

3.12 Summary

This chapter have described a detailed review of Dynamic heterogenous information networks, the importance of change discovery, and different methods to analyse changes in a DHIN. The presented review also explains the difference between dynamic network and DHINs. It also analyses different measures of dynamic networks, applications of dynamic networks. A detailed survey of change mining algorithms for static and dynamic networks were also reviewed. Moreover, the comprehensive related work on dynamic network change mining was discussed and different algorithms were also critically explained to find the problems in the existing works which lead to the decision of developing a system that can discover the changes from DHINs using knowledge engineering and data mining methods.

The next chapter will elaborate the semantic web technologies from the data mining perspective. The next chapter will also discuss how the use of knowledge engineering helps to improve the performance of data mining methods.

Chapter 4 Semantic Web Technologies in the perspective of data mining

4.1 Introduction

This chapter presents an overview of semantic web technologies from the perspective of data mining. An overview of the Semantic Web technologies is presented first and then what semantic technologies or layers of Semantic Web architecture can make the data mining process efficient are discussed in detail.

4.2 Semantic Web Technologies

Semantic Web technologies and frameworks are a collection of technologies that empower the Web data to provide a formal definition and description of classes, roles, and relationships within a given field. Companies like Google, Amazon, YouTube, Facebook, LinkedIn and others are constantly growing apps and their data volume and variety is rapidly growing (Domingue, Fensel, & Hendler, 2011). Semantic Web technologies support to the extraction of useful information which can be accessible everywhere. The Semantic Web layer architecture is briefly introduced in this chapter. The Web Ontology Language (OWL) is a prominent layer of semantic web architecture that is explored in detail in this chapter. The description and examples of OWL are demonstrated in the Manchester syntax which is user-friendly and easier to understand (Horridge et al., 2006; Motik et al., 2009).

4.2.1 Semantic Web layered architecture

The Semantic Web is an extension of the World Wide Web has emerged a prominent role in the web as well the machine readability, understanding and reasoning. The Semantic Web aims to render machine-readable Internet data. Web pages are documents on the web. These web documents or web pages are used for data storage and retrieval. The issue is that the system does not know the semantics of the contents of certain documents. The semantic web allows the machine to consider the context behind the web pages or documents. The Semantic Web layered architecture proposed versions of Berners-Lee (Tim Berners-Lee, James Hendler, & Ora Lassila, 2001) are most well-known. Haytham Al-Feel et.al evaluated the four versions of Tim Berner-Lee identified some shortcomings of those architectures (Al-Feel, Koutb, &

Suoror, 2009). The latest version of semantic web layering cake tweak is presented in Figure 15.

Layer 1: The URI/RI (Uniform Resource Identifiers / Internationalised Resource Identifier) is the first layer of the architecture that identify web pages, things or concepts or resources on the Web and in the repository that are meant to be viewed in a browser.

Layer 2: Extensible Mark-up Language (XML) easy to use knowledge representation language for creating a simple knowledge-base. It defines a set of rules on how to encode documents in a format that is both human-readable and machine-readable.

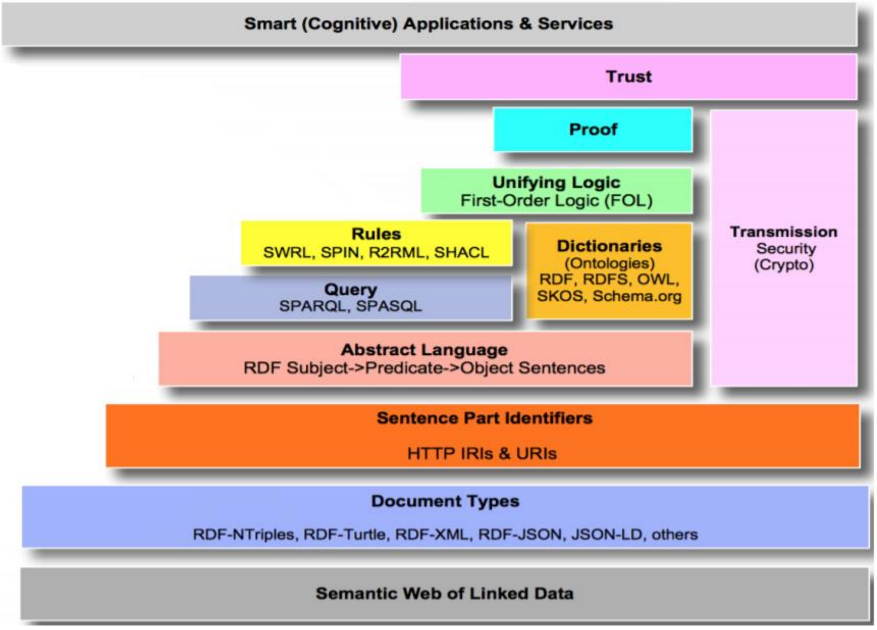


Figure 15 Semantic Web Layer Cake Tweak (Idehen, 2017)

Layer 3: Resource Description Framework (RDF). It represents information about resources in graph form. RDF is based on triples subject-predicate-object which form a graph. All data in the semantic web use RDF as a primary standard.

Layer 4: Web Ontology language is the heart of all Semantic Web applications. The benefit of using OWL is that its ontological framework permits the data to be shared and reused across applications, organisations, institutions and community boundaries (Simperl, 2009). A detailed discussion on this representation is presented in the next section. Moreover, Figure 16 elaborates the concept hierarchy of semantic web layers.

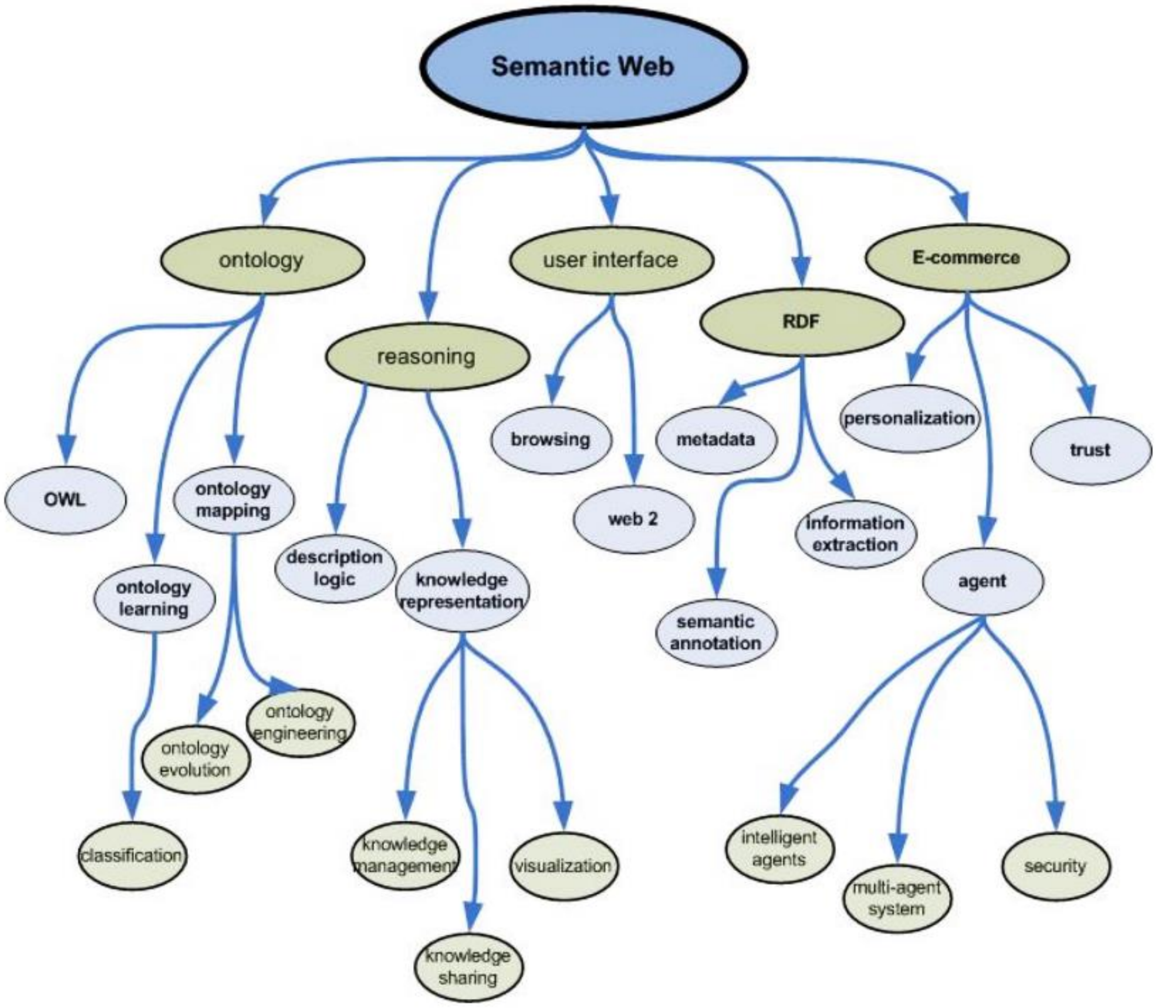


Figure 16 Graph to explain concept hierarchy semantic web layers(Wei, Barnaghi, & Bargiela, 2008)

4.2.2 Web Ontology Language

Web Ontology Language (OWL) (L. Ding, Kolari, Ding, & Avancha, 2007; Grau et al., 2008) is a functional layer of semantic web layered architecture. It is the latest standard, which was developed by members of the World Wide Web Consortium (W3C) and Description Logic community. OWL supports defining explicit classes, roles, constraints and restrictions. OWL helps in machine understanding and interpretation of web contents with additional vocabulary and formal semantics. OWL is the next layer on Resource Description Framework (RDF) (E. Miller, 1998) and it is a rich language for defining concepts. At present, it is becoming the most common research subject. It attracts many research communities which are interested to implement it in knowledge engineering processes and natural language processing (NLP) applications (L. Li, Yang, & Wu, 2006).

It also supports the interoperability between heterogeneous systems involved in commonly interested domain applications, by providing a shared understanding or conceptualisation (X. Wang, Gorlitsky, & Almeida, 2005). Jorge (de Vergara, Villagrá, & Berrocal, 2004) mentioned that OWL is quite useful in the integration of diverse management domains as compared to other computer representations. The use of ontologies has many general advantages: (1) it is capable of capturing communal knowledge that is not private to a person or a member but accepted by a broad community, (2) It is capable of facilitating information sharing, (3) it is capable of processing content based on meaning rather than syntax since it is a deliberate semantic structure (Guarino, 1998), (4) the consistency of ontologies can be verified by using an OWL reasoner and (5) it supports in machine reasoning and understanding of human represented knowledge. OWL Lite, OWL DL and OWL Full are expressive sub-languages of OWL representation (Horridge, Knublauch, Rector, Stevens, & Wroe, 2004).

1) OWL-Lite is a simple language and it is easier to implement for the experts. It is utilised in problems where only a simple class hierarchy and simple constraints are needed.
2) OWL DL is based on Description Logics and is a sublanguage of OWL Full. It supports checking inconsistence knowledge in ontology and facilitate automatically computing the classification hierarchy.
3) OWL-Full is an OWL sub-language and is the most expressive. It uses all OWL language concepts. It is used in states where very high expressiveness is obligatory than being able to guarantee the decidability of the language. Therefore, automatic reasoning on OWL-Full ontologies cannot be carried out.

The OWL ontology consists of classes, individuals and properties and brief descriptions of these terms will be presented next.

4.2.2.1 *OWL Properties*

Properties are binary relationships that connect individuals. OWL has three property types: (1) Data Type Properties, (2) Annotation properties and (3) Object properties and these can be defined as sub and super properties.

a) Data type properties

A data type property connects an individual to an XML schema data type value or an RDF literal. These properties have only one characteristic (i.e. functional).

- **Functional properties**

This property limits one relationship with a given data type value or an RDF literal. For instance, a functional property is the 'hasDateOfBirth'. If it is claimed that ‘Maria hasDateOf Birth 03-05-2018'. then the individual 'Maria' cannot hold another date of Birth information by using the ‘hasDateOfBirth’ property.

b) Annotation properties

These properties are used to define the metadata or description of the terms (i.e. classes, individuals, properties).

c) Object properties

These properties are used to connect an object to another object and can be defined as the inverse. If a property connects individual ‘a’ to individual ‘b’ then its inverse property will connect individual ‘b’ to individual ‘a’. For instance, the ‘hasDirSubNode ’ is the inverse property of the ‘hasDirSupNode’ property (Ramzan, Wang, & Buckingham, 2014), both are object properties. Moreover, an object property can have functional, inverse functional, transitive, symmetric, asymmetric, reflexive and irreflexive characteristics.

i. Functional properties

This property limits one relationship with a given individual. For instance, if it is claimed that ‘Maria hasGender Female' ('Maria' and 'Female' are individuals) then the 'Maria' (individual) cannot have a relationship with ‘Male' (individual) by using the 'hasGender' property (have functional characteristics).

ii. Inverse functional properties

This property holds the characteristics of a functional property. The inverse property of this also holds the same characteristics.

iii. Transitive properties

This property involves the shift of relationships. For instance, ‘Marry’ has brother ‘William’ and ‘William’ has brother ‘Tom’ (‘Marry’, ‘Tom’ and ’William’ are individuals). The relationships are defined among these individuals via using ‘hasBrother’ property. The ‘hasBrother’ is defined as a transitive property, then the OWL reasoner can infer that ‘Marry’ has brother ‘Tom’.

iv. Symmetric properties

These properties make identical relationships for the individuals. For instance, the ‘isEqualTo’ property is an asymmetric attribute. If the ‘x’ is equal to ‘y’ then it is inferred by using the OWL reasoner that ‘y’ is equal to ‘x’. In this way, the specified property (i.e. ‘isEqualTo’) is its inverse property.

v. Asymmetric properties

These properties are opposite to symmetric properties. For instance, If the individual ‘Tim’ is connected to the individual ‘Nick’ via the ‘isChildOf’ property, then it can be inferred that ‘Nick’ is not connected to ‘Tim’ via the ‘isChildOf’ property. However, ‘Nick’ could be related to another individual ‘Steve’ via the ‘isChildOf’ property. In other words, if ‘Tim’ is a child of ‘Nick’, then ‘Nick’ cannot be a child of ‘Tim’, but ‘Nick’ can be a child of ‘Steve’.

vi. Reflexive properties

This property links an individual to itself. In comparison to irreflexive properties, these properties are opposite.

vii. Irreflexive properties

These properties are used in cases where both individuals are not the same or must be unique to each other. For instance, the 'isFatherOf' property (irreflexive): an individual 'Alex' can be linked to 'Matt' alongside the 'isFatherOf' property, but 'Alex' cannot be 'isFatherOf' himself.

4.2.2.2 *OWL classes*

The classes are the main building blocks of the OWL language. The OWL classes are used to define the concepts of the knowledge domains. The ‘class descriptions’ support in the description OWL classes that can be combined into ‘class axioms.

1. **Class description**

It describes an OWL class. For instance, the Pizza is the class name and it is extending Food class and has further description ‘hasBase some PizzaBase’. In this example, the Pizza class name and the class extension (Food) of the Pizza class are class descriptions.

(a) **Named classes**

The Named classes are called ‘primitive’ classes. The ‘Thing’ is an example of a predefined ‘primitive’ class. All the classes defined in the ontology are the subclasses of this class. For example, Pizza is the grand subclass of the ‘Thing’ class because the ‘Food’ is the subclass ‘Thing’.

(b) **Unnamed classes**

An unnamed class is also known as an ‘anonymous’ class (Horridge et al., 2009; Horridge et al., 2004). A restriction is applied to a named class that represents itself as an anonymous superclass of the ‘Named’ class. For example, the ‘hasBase some PizzaBase’ is an anonymous superclass of the Pizza class. Such classes contain the individuals who satisfy the logical description. It means the individuals who satisfy the ‘hasBase some PizzaBase’ condition will classify the subclasses of Pizza. The Boolean operators (AND (Π), OR (U) and NOT (¬)) are used to construct the logical descriptions and expressions from other classes.

a. Enumeration

An enumeration allows a class to be defined by comprehensively listing its instances. For example, the ‘Spiciness’ class has the following enumerated individuals (i.e. ‘Hot’, ‘Medium’ and ‘Mild’).

```
Class: Spiciness
        SubClassOf:
                hasSpiciness only {‘Hot, Medium, Mild}
```

b. Property restriction

Class descriptions created by constraints on properties are called restrictions. OWL restrictions are classified into two distinct categories (1) qualified cardinality restriction and (2) value restriction.

I. Value restriction

The ‘value restriction’ along with a property and a filler are employed to state a set of individuals. The ‘value restriction’ is further divided into three distinct categories: (1) existential restriction, (2) universal restriction and (3) hasValue restriction.

(1) Existential restriction

The existential restriction is also called ‘Some’ restriction. These restrictions describe the individuals who have at least one relationship to individuals who are members of some other specified class. For example:

```
        hasBase some PizzaBase
```

This existential restriction describes the set of individuals that have at least one ‘hasBase’ relationship to an individual that is a member of the ‘PizzaBase’ class (Horridge & Bechhofer, 2011).

(2) **Universal restriction**

This restriction is also known ‘AllValuesFrom’ and ‘only’ restriction. This restriction along the specified property has all values from the filler class.

hasBase only ThinAndCrispyBase

It means that the set of individuals that only has ‘hasBase’ relationships to individuals of the ‘ThinAndCrispyBase’ class.

(3) hasValue restriction

This restriction describes the set of individuals that have at least one relationship to another specific individual along with a specified property. For instance, the 'America' has at least one connection to a specific individual along with the 'hasCountryOfBirth' property after 'hasValue' constraint.

hasCountryOfBirth value America

II. Qualified cardinality restriction

Cardinality restrictions describe sets of individuals in terms of the number (positive integer) of relationships that the individuals must contribute to for a given property. (1) Minimum, (2) maximum and (3) exactly are three different types of cardinality restrictions.

(1) Minimum cardinality restriction

These restrictions employ the minimum number of relationships that an individual can participate in for a given property. For example, an ice cream required minimum of 3 fruit toppings.

```
Class: IceCream
        SubClassOf:
                hasTopping min 3 FruitTopping
```

This restriction represents the minimum cardinality restrictions which indicate the ’greater than or equal to several relationships via given property.

(2) **Maximum cardinality restriction**

These restrictions define the maximum number of relationships that an individual can contribute to a given property. For example, a cricket team can have a maximum of 11 players.

```
Class: CricketTeam
        SubClassOf:
                hasPlayers max 11 Players
```

This restriction represents the maximum cardinality restrictions which indicate the ’less than or equal to’ several relationships via a particular property.

(3) Exactly restriction

These restrictions define the exact number of relationships that an individual must participate in for a given property. For example, a person must have exactly one national insurance number (NIN).

```
Class: Person
        SubClassOf:
                hasNIN exactly 1 NIN
```

viii. **Intersection, union and complement**

The intersection, union and complement are also anonymous classes that are used to define further constraints. Their details are given below.

(1) **Intersection**

The intersection is used to integrate two or more classes or anonymous classes using the AND operator. For example, the ‘Food’ and ‘Pizza’ are two classes and the ‘Pizza’ is a subclass of the ‘Food’ class.

Food and Pizza

It means that the anonymous class is described as a subclass of ‘Food’ and also describe a subclass of the ‘Pizza’.

(2) **Union**

A union is used to combine two or more classes. The OR operator is used to define this restriction. For example, the ‘Men’ and ‘Women’ are two classes.

Men or Women

It means that it holds the individuals that belong to either the class Men or the class Women (or both).

(3) **Complement**

It represents the individuals that do not belong to the class extension of the class description. For example, the ‘Male’ class complement the ‘Female’ class.

```
Class Male
        SubClassOf:
                not (Female)
```

2. **OWL Axioms**

OWL contains some language constructs for integrating class descriptions into class axioms. Class axioms typically contain additional components that hold necessary or necessary and sufficient conditions.

(a) **Subclass axioms**

Subclass axioms show ‘necessary’ conditions. The necessary condition is a state that must be overcome if another is to occur. For example, if ‘Pizza’ is a necessary condition for the ‘VegetarianPizza’ class. The semantics of this restriction is that without the ‘Pizza’, the ‘VegetarianPizza’ cannot exist. However, the existence of the ‘Pizza’ does not guarantee the existence of ‘VegetarianPizza’.

```
Class: VegetarianPizza
        SubClassOf: Pizza
```

(b) **Equivalent class axioms**

These axioms are also called ‘necessary and sufficient conditions. A necessary and sufficient condition infers that, if and only if the conclusion is true, the previous statement is true. For example, if an individual is a member of the class ‘Pizza’ and it has at least one individual that is a member of the class ‘SeaFoodTopping’ then these conditions are sufficient to determine that the individual must be a member of the class ‘SeaFoodPizza’.

```
Class: MeatyPizza
        EquivalentTo:
                Pizza and (hasTopping some SeaFoodToppingt)
```

(c) **Disjoint axioms**

These axioms characterise supplementary ‘necessary’ conditions. OWL classes are overlapped by default. Disjoint axioms support stopping the overlapping mechanism. For example, two classes are defined as disjoint classes, if an individual belongs to one class, then that individual cannot belong to another class. For example, the ‘Male’ and ‘Female’ are completely distinct classes and these classes can be defined as disjoint classes. Furthermore, any individual of the ‘Male’ class cannot belong to the ‘Female’ class.

```
DisjointClasses:
        Male, Female
```

(d) **Covering Axioms**

OWL make the open-world assumption. It implies that if anything is explicitly not added to the knowledge base, it does not mean that it is false. For instance, if ‘Weak’ is a superclass of ‘Monday’, ‘Tuesday’, ‘Wednesday’, ‘Thursday’, ‘Friday’, ‘Saturday’ and ‘Sunday’ classes. No individual of ‘Week’ class needs to be an individual of its subclasses. Such an individual could be simply loose between the subclasses. In this case, an anonymous class can be created that makes the subclasses (i.e. (‘Monday’, ‘Tuesday’, ‘Wednesday’, ‘Thursday’, ‘Friday’, ‘Saturday’ and ‘Sunday’) to cover the ‘Week’ class completely.

```
Class: Week
EquivalentTo:
        (Monday or Tuesday or Wednesday or Thursday or Friday or Saturday or
Sunday)
Class: Monday
        SubClassOf: Week
Class: Tuesday
        SubClassOf: Week
Class: Wednesday
        SubClassOf: Week
Class: Thursday
        SubClassOf: Week
Class: Friday
        SubClassOf: Week
Class: Saturday
        SubClassOf: Week
Class: Sunday
        SubClassOf: Week
```

(e) **Closure Axioms**

A closure axiom in a ‘Named’ class is comprised of a universal restriction. For example, an ‘Margherita’ pizza has only ‘MozzarellaTopping’ and ‘TomatoTopping’ (Horridge et al., 2009).

```
Class: Margherita
        SubClassOf:
                hasTopping only (MozzarellaTopping or TomatoTopping)
```

3. **Inconsistent classes**

If a class cannot possibly have any instances then it is classified as an inconsistent class (Rector et al., 2004). An OWL reasoner(Dalwadi, Nagar, & Makwana, 2012) is always used to infer inconsistent classes from a knowledge-base. The OWL reasoners or reasoning engines are used for concepts classification and consistency checking. There can be several reasons for inconsistency (i.e. subsumption and disjointness). For instance, the ‘CheeseTopping’ and ‘VegetableTopping’ are two disjoint classes. If one creates the ‘CheeseyVegetableTopping’ class and designed, it is a subclass of the ‘CheeseTopping’ and ‘VegetableTopping’ classes. It means that ‘CheeseyVegetableTopping’ is a ‘CheeseTopping’ and a ‘VegetableTopping’. More formally, all individuals that are members of the class ‘CheeseyVegetableTopping’ are also (necessarily) members of the class ‘CheeseTopping’ and (necessarily) members of the class ‘VegetableTopping’. This modelling leads to inconsistency of the ‘CheeseyVegetableTopping’ class. In the next section, the uses of ontologies in the data mining domain are discussed in detail.

4.3 Ontology in data mining

Research in the area of ontology has led to standards for modelling and codifying domain knowledge. Generally, ontologies are developed to specify a specific domain for example Gene Ontology (Go) (Consortium, 2004) and more than 300 ontologies for the medical domain in the National Centre for Biomedical Ontology NCBO. Ontology is a clear conceptualization specification and a recognized method to define the semantics of data (Gruber, 1993). Knowledge engineering is a process to formalise human expertise into ontology (Wielinga, Schreiber, & Breuker, 1992). Currently, ontologies have become a key technology for the intelligent knowledge engineering process by providing a framework for shared conceptual models about a domain. Data mining is the process of mining nontrivial, implicit and previously unknown useful information patterns from data. The benefits of integrating domain information into data mining have been verified by various research studies. Semantic data mining is the data mining task that logically combines domain information particularly formal semantics into the process (Dou, Wang, & Liu, 2015). Using ontology-based knowledge engineering have various advantages by providing prior knowledge to bridge the gap between the data applications, data mining algorithm and performance while ontologies are used in several fields of science, such as artificial intelligence, biomedical science, and knowledge engineering but their use in change mining is very limited and as per authors knowledge there is no system ontology-based system that can detect changes in temporal snapshots of graph data. Following are the descriptions and details of data mining methods using ontologies.

4.3.1 Knowledge engineering

Knowledge engineering (KE) is a field of artificial intelligence that studies the representation acquisition, reasoning, decision making and application of knowledge. Knowledge engineering also includes big data, machine learning, knowledge discovery, uncertain reasoning, knowledge mapping, machine theorem proving, and expert systems. Moreover, KE is the process of developing knowledge-based systems in any field of life, these systems can be in the public or private sector, in commerce or industry (Z. Shi, 2021).

4.3.2 Association rule mining using Ontologies

Association rule mining is a basic data mining task and well used in various applications. The associative classifiers use association rule mining to extract interesting rules from the training data, and the extracted rules are used to build a classifier. Association rule mining based on ontologies can have constraints for making queries and extracting interesting facts and rules in the association mining process. Bellandi et al. (Bellandi, Furletti, Grossi, & Romei, 2007) designed an ontology-based association rule mining method, which queries the ontology to retrieve the instances used in the association rule mining process. Association rule mining based on ontologies can benefit in different stages of the mining process: data understanding, task, design, results in interpretation, result dissemination, classification of data, data retrieval and traversing and discover the hidden patterns, identification of consistent data, filtration of inconsistent data (Martínez-Romero et al., 2019).

4.3.3 Information retrieval using Ontologies

The information retrieval using ontologies is a task to extract information from natural language text by processing them automatically. The inference engine or OWL reasoner (Parsia, Matentzoglu, Gonçalves, Glimm, & Steigmiller, 2017) supports the extraction of relevant information from the ontologies and this can also help to reason the knowledge that is even not explicitly added in the knowledge-base.

4.3.4 Classification using Ontologies

Classification (Sundar, Chitradevi, & Geetharamani, 2012)is a process of classifying the data based on their characteristics. Ontology-based classification supports a description of the characteristics and then infer the data patterns. The common understanding about the knowledge-base is that it can only classify the labelled data but the unlabelled can also be classified with the help of semantic encoded classification. For example, Balcan et al. (Balcan, Blum, & Mansour, 2013) have utilised the unlabelled classification method through ontology generation. Their classification method produced classification consistency by decreasing the error rate of the unlabelled classification. Kleinberg et al. (J. M. Kleinberg, Kumar, Raghavan, Rajagopalan, & Tomkins, 1999)developed an algorithm called HITS to classify core relational entities in the semantic knowledge graphs for the classification of dynamic text topics. Similarly, Allahyari et al.(Allahyari, Kochut, & Janik, 2014) also used ontology-based techniques for the classification of document-based text.

4.3.5 Clustering using knowledge-base

Clustering (Jain, 2010) is a branch of data mining that involves grouping a set of objects in the same cluster which is similar to each other. Semantic similarity can be estimated using the knowledge-base to determine the distance between concepts or terms (Hotho, Staab, & Stumme, 2003; Jain, 2010) Most of the research work is found on text and document clustering. For example, a self-organized genetic algorithm for text clustering based on ontology method to capture the associated semantic similarity is developed (Song & Park, 2009).The focus of this algorithm is to enhance the performance in comparison with the standard genetic algorithm and *k*-means in the same similar environments. Jing (Jing, Covell, & Rowley, 2010)proposed a new knowledge-based vector space model (VSM) for text clustering. They employed ontology to re-weight the vectors in knowledge-based vector space for text clustering. (Fodeh, Punch, & Tan, 2011)utilised ontology to greatly reduce the number of features needed in the document clustering task. Gene Ontology (GO) is a comprehensive computational model of the biological system (Consortium, 2004). (Ovaska, Laakso, & Hautaniemi, 2008) implement a gene clustering task from microarray experiments with the help of GO. They designed two types of structures: (1) Graph structure and (2) Information Structure. The graph structure-based methods are utilised the GO hierarchical structure to measure the gene similarity. In contrast, information structure-based methods employ the GO information content in a reference gene set.

4.4 Summary

This chapter presented an overview of Semantic Web technologies. The latest version of Semantic Web Layer architecture has been presented that help in understating its different layers and their connections with each other. OWL is a prominent layer of the architecture and in this study, we specifically used this standard for knowledge engineering processes. The uses of the ontologies are discussed for data mining purposes in Section 4.3. Thus, this conceptualisation supports the understanding of the key functionality of the ontologies in the proposed model presented in Chapter 5.

Chapter 5 A New Framework for DHIN Change Discovery

5.1 Introduction

The algorithms and frameworks described in chapter 3 are developed to detect change points from static and dynamic single-type networks. However, real-world network systems are dynamic, multi-type, heterogeneous and evolving with time. Chapters 2, 3 and 4 above also give an in-depth background understanding of HINs, DHINs literature review, and an overview of ontologies in the perspective of data mining respectively. In this chapter, we will present a new change mining framework called “ChaMining” which is Ontology and Data mining-based change discovery system. This chapter presents the design and details of the proposed framework and is ordered as follows:

- Section 5.2 outlines each component of the proposed framework and how these components work interactively to detect changes in DHINs.
- Section 5.3 depicts the scenarios of the proposed framework.
- Section 5.4 elaborates on the data acquisition and pre-processing layer and its components.
- Section 5.5 explains the knowledge engineering layer and its components.
- Section 5.6 expands the change detection layer and its related components.
- Section 5.7 elaborates the data mining methods used in this research
- Section 5.8 presents the summary.

5.2 Proposed Framework for Change discovery using Knowledge Engineering and Data Mining Methods

Change detection algorithms and frameworks described in section 3.6 to 3.9 of the literature review are mainly designed to detect change points for single type and synthetic data. These algorithms also elaborate that the initial point to detect changes in a network is to divide it into temporal snapshots. Dividing a large dynamic heterogeneous information network will help to divide the network into smaller networks which can be further exploited to discover changes.

This thesis proposes and develops a novel framework for change discovery in DHINs, where this framework aims to discover changes at different levels. Thus this framework will include components that support the process of temporal change discovery and visualisation for:

- Constructing a DHIN using knowledge engineering processes to develop a temporal knowledge base and then dividing it into temporal knowledge graphs for further analysis.
- Analysing rule-based changes in the networks at different time windows, these changes discover interesting relations between the different nodes at different times. The intention is to find strong rules in the DHINs using the measures of support and confidence.
- Extracting patterns to discover node insertion or deletion based changes and temporal similarity measure time-based data can be used for understanding the changes or stability of the patterns.
- Assessing the clustered network section, overlapping clusters and discover community-level changes to find different updates in the different time stamps.

The main challenge is how these can be achieved? This research proposes to develop a three-layered framework. This framework will find changes in a dynamic heterogeneous information network by:

- Firstly, it will use all three types of DHINs that are structured, semi structured and unstructured networks data to convert it into knowledge-base. It is convenient to convert a structured data into a knowledge base by employing parser or machine-readable engine but semi-structured and unstructured data set requires additional processing to convert them into a temporal knowledge base.
- Secondly, use temporal knowledge graphs at various time frames to extract time-related knowledge graphs.
- Thirdly, extract node-based and community-based patterns from these time frames and apply the data mining method at each time-sliced DHIN knowledge graph.
- Fourthly, calculate similarity measure between node based extracted data patterns and community based extracted data patterns at different temporal knowledge graphs to find stable and change patterns.

These reflections produce the main framework presented in Figure 17, which is explained in more detail in sections 5.4, 5.5, 5.6 and 5.7. The framework is divided into three layers by the dotted lines. The first layer is the data acquisition and pre-processing layer, the second layer is the knowledge engineering layer, and the third layer is the change detection layer. The parallelograms used in the figure shows input data, the rectangles represent processes, the database icon displays the knowledge base, and the arrow are the connectors that shows relationships between different components of the framework.

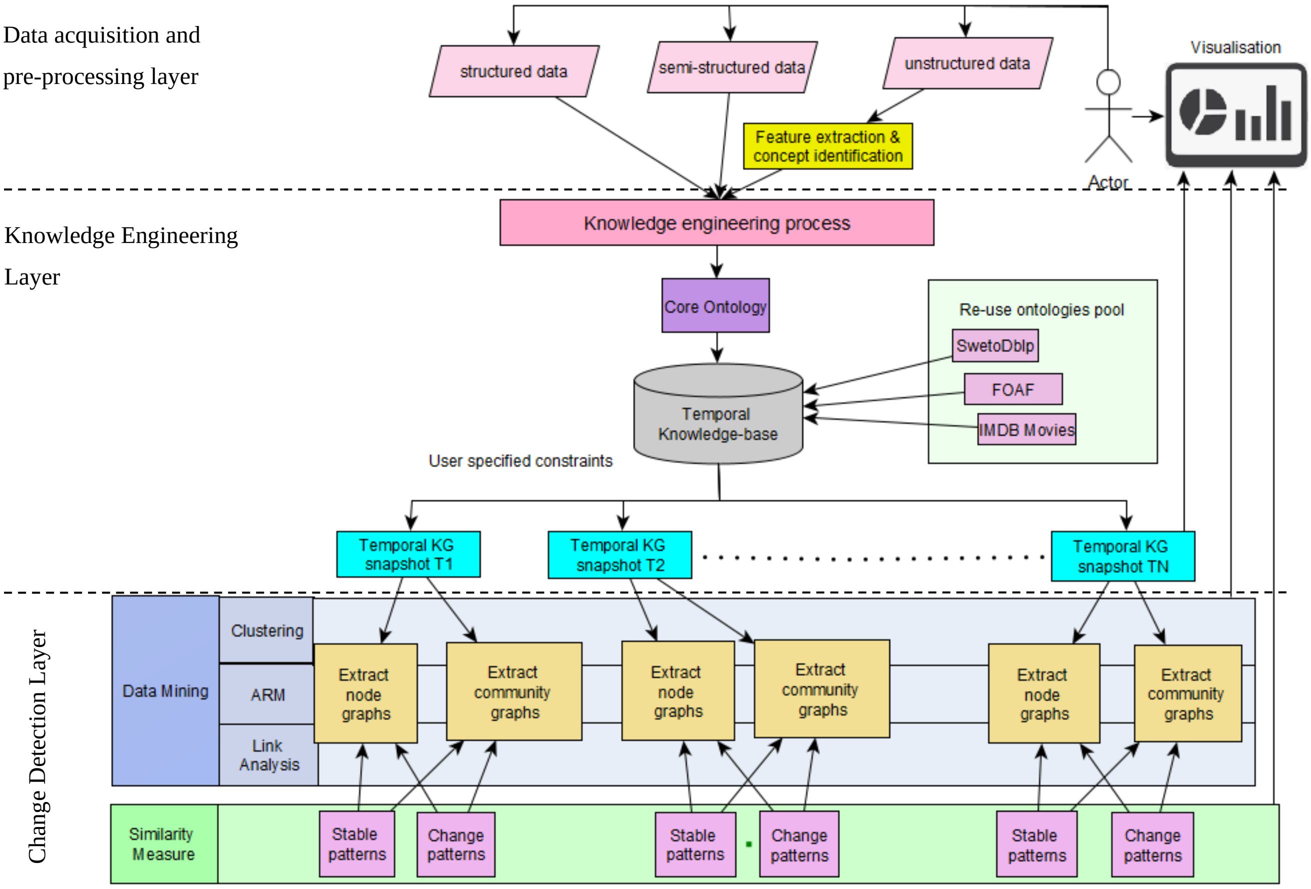


Figure 17 ChaMining system that discovers changes in DHINs

5.3 Proposed system scenarios

This section presents a scenario diagram presented in Figure 18 to elaborate on some of the potential features that can be expected in the ChaMining change discovery system. User input the temporal dynamic heterogeneous network data snapshots in this system. The user will also specify the target meta-paths to develop the knowledge graphs. Then system parses the user given data into knowledge graphs using a domain ontology framework. Next, these knowledge graphs will be used by the system as a baseline to discover changes. The system not only parses or understands structured data but also transforms the unstructured or semi-structured DHINs data into a structured format (see details in Section 5.3) and maps it to the knowledge-base.

Moreover, the user can specify certain criteria or constraints to extract multi-type patterns from the knowledge graphs. The system will respond to user constraints or queries and will produce the required multi-type knowledge graphs. The system will employ data mining methods clustering, association rule mining and link analysis to discover changes. In case a user is interested in the structural changes in the temporal knowledge graphs. The system will use clustering to find these changes at two different time snapshots. The Association rule mining component of the system will discover interesting relations between temporal graphs. In the next stage, the system will use these multi-type knowledge structures to detect change at different time-sliced windows by calculating similarity measures between two-time windows. The user also has the option to select local (node level) changes, global (edges level) changes and community-level changes. The system will compute, calculate and then display these changes in the form of numerical values and knowledge graphs which are in a human-understandable format.

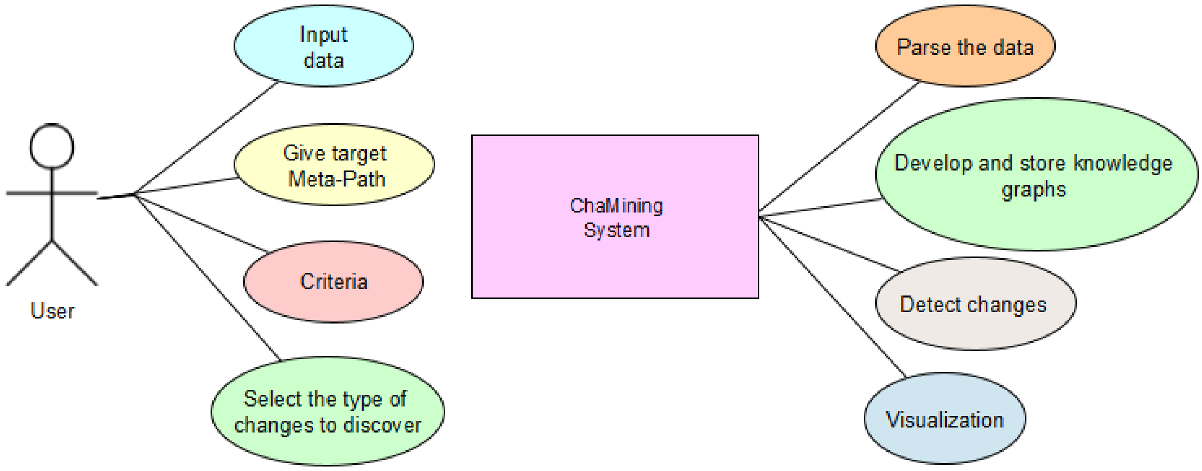


Figure 18 Scenario Diagram of the ChaMining system that discovers changes in DHINs

5.4 Data acquisition and pre-processing Layer

Data acquisition and pre-processing is mandatory step for the knowledge engineering process and machine learning methods. The data must be pre-processed in an appropriate format to convert it into a knowledge base otherwise “garbage in garbage out” is valid saying for this kind of knowledge engineering and data mining project(Chicco, 2017). Most of the real-world systems such as healthcare systems, co-authorship systems, emails and social media websites contain interacting objects that can be modelled as DHINs. These networks have a sequence of time-oriented observations of networks acquired at different time windows. These observations can also be described as $O_1, O_2 \ldots O_n$ at time $T_1, T_2, \ldots. T_n$ where each observation contains set of Nodes $N=\{n_1,n_2, \ldots n_n\}$ and edges $E=\{e_1,e_2, \ldots. e_n\}$. Thus the data acquisition and pre-processing layer in this study have three data mapping components which are:

- Unstructured data mapping
- Structured data mapping
- Semi-Structured data mapping

Unstructured data mapping component also has feature extraction & concept identification sub-component to convert unstructured data for knowledge engineering process. In the data acquisition and pre-processing layer, three algorithms have also been developed to map unstructured, semi-structured and structured data. Algorithm 1 is designed to map the unstructured Enron email data in the Temporal knowledge base. Algorithm 2 and Algorithm 3 are developed to model semi-structured and structured data into DHINs temporal knowledge base respectively.

5.4.1 Unstructured data mapping

When the data is unstructured, it is a challenging task to construct a DHIN by extracting objects and their relationships. It is essential to categorise the data into different sets such as entities, attributes and relationships if exist, before the transformation of unstructured data into a knowledge-base for further processing and change discovery. There are different tools available for information extraction from the unstructured data, for example, NLTK(Bird, 2006) and GATE(D. M. H. Cunningham & Bontcheva, 2011) are most commonly used for extracting objects and their relations. The General Architecture for Text Engineering (GATE) (H. Cunningham, 2002) is also utilised to process text problems. The GATE application provides a list of nouns (concepts) and verbs (properties).

For example "David has sent an email to Martin in 2004". GATE will tokenise this sentence into a set of words as follows:

David, sent, emails, to, Martin, on, 2004

Then GATE states the parts of speech tags for each word as follows:

David/NN sent/VB emails/NNS to/IN Martin/NN on/IN 2004/CD

Where NN means singular noun, NNS means plural noun, VB means verb, IN represents a preposition and CD means a cardinal number.

After the part of speech (POS) corpus has been developed, it can be used as input for the WordNet database for extracting the required semantic information. The user can also take part in the development of classes and relations by specifying meta-paths for example if the user is interested to develop a temporal knowledge-base on a specific meta-path using email data such as EFETS EmailFrom-EmailTo-Subject then the user can also specify these meta-paths to extract relational patterns.

The initial idea was to use these tools but after analysis and evaluation, it was decided to use python library email. message[10] to extract objects and their relationships from Enron emails unstructured corpus[11] otherwise the system user need to be an expert information extraction engineer to effectively use these information extraction tools. Figure 19 shows the steps that are employed to prepare unstructured Enron email data files in a structured (.csv) data file:

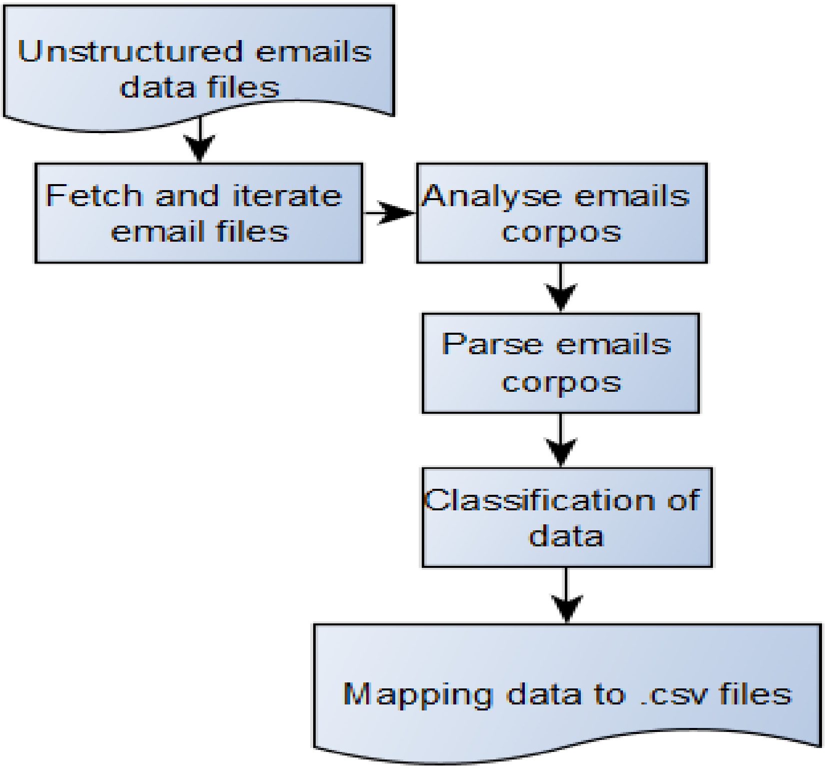

Unstructured emails data files
Fetch and iterate email files
Analyse emails corpos
Parse emails corpos
Classification of data
Mapping data to .csv files

Figure 19 Steps to convert Enron emails in structured data

[10] https://github.com/shantnu/Enron-Data-Set
[11] https://www.cs.cmu.edu/~./enron/

The unstructured Enron data have thirty-five thousand directories and about half a million files which contain emails records of 150 users. Before the system can perform any analysis it needs to read and analyse these large numbers of files to convert these files in required comma separated version format. To get the required format data the root directory will be specified and "os.walk" function will be used to read all the files iteratively. This process will give us directories, subdirectories and lengths of email files. In this process, the fetched and iterated files will be analysed to see which information need to be extracted from a mail for example, if the user is interested in extracting the Email from an email to a relationship user can specify and extract the required relations. This preprocessing of email unstructured data to structured data has been performed using this code available at GitHub[12]. The generated comma separated version (.csv) data is mapped to the core ontology for developing a knowledge-base by using Algorithm 1. The feature extraction and concept identification process in the ChaMining framework uses the user interruption to identify entities in the text and employ other semantic annotation. For example, in Enron email data EmailFrom, EmailTo and Message can be different entities specified by the user. The data of the CSV file will be mapped in the ontology via using Algorithm 1. The first step is to import, load and extend the core ontology (see details in Section 5.4.2). The OWL reasoner (see details in Chapter 4 and in Section 4.2.2.2) is called and initialised to extract the nodes, properties, relationships and constraints of the core ontology. This step supports extending the core ontology classes which are associated with the domain of discourse and aid in the mapping of the CSV file data. The said data is modelled and the classes of the core ontology are extended in the target ontology. The nodes of the core ontology are iterated and the CSV data is also iterated for mapping. The relationships of the attributes are designed conditionally, if the value is an object then the property will be designed as a sub-property of an object property, otherwise its data value helps in the determination of primitive data types (integer, double, string, boolean) (see details in Chapter 4 and Section 4.2.2.1). The subclass relationship is designed between the core ontology and the classes of the target ontology. All the new and extended classes, properties, constraints and relationships are saved in the target ontology on a defined path.

[12] https://github.com/shantnu/Enron-Data-Set

Algorithm 1: The CSV data mapping algorithm

```
Input: Comma separate version data file
Output: Domain based ontology
Import, load and extend core ontology
IF (NOT file ==NULL) THEN
foreach  Attribute[a] do
        IF (NOT Attribute[a] ==NULL) THEN
               Foreach CoreNode[c] do
                     Foreach Node[n] do
                     SetSubClassRelationship(Node[n], CoreNode[c])
        IF(Attribute[a].TYPE==Object) THEN
           setAssociation(Node[n], Attribute[a], Node[n+1])
        END
        IF(NOT Attribute[a].TYPE==Object) THEN
          Foreach Value[v] do
        Node[n] Attribute[a]← Value[v]
          END
        END
        Add(Node[n], Attribute[a], Value[v])
        SaveOntology← PATH
     END
   END
END
END
generate: .OWL file
```

5.4.2 Semi-Structure data mapping

Algorithm 2 is used to map the semi-structured data to the knowledge-base. Semi-structured data is in XML format containing different types of nodes and their relations. The difference between Algorithm 1 and Algorithm 2 is that the input data for the semi-structured algorithm is not raw data. An XML well-formed file is utilised as data input. Therefore, the GATE and WordNet applications for the identification of classes and predicates are not utilised. The nodes of the XML documents are retrieved in sequence with their associated parent nodes that support building the classes and properties hierarchical structure in the target ontology. The relationships between the classes are developed using the OWL object properties (see details in Chapter 4 and Section 4.2.2.1) and further these are defined as a subclass of existing classes in the core ontology. In case, a node holds some data values like string, floating and whole numbers in the XML then data properties (see details in Chapter 4 and Section 4.2.2.1) are employed for mapping of such data in the ontology. Finally, the ontology is saved on a given path.

Algorithm 2: The XML data mapping algorithm

```
Input: XML data
Output: Domain-based ontology
Import and load core ontology;
IF (NOT (file.Empty) AND (Length (file) > 1)) THEN
            Foreach CoreNode[c] do
                    Foreach Node[n] do
                           get (Node[n])
                             foreach  Attribute[a] do
                           IF (NOT Attribute[a] ==NULL) THEN
                              SetSubClassRelationship(Node[n], CoreNode[c])
                        END
                          IF(Attribute[a].TYPE==Object) THEN
                             setAssocaition(Node[n], Attribute[a], Node[n+1])
                      END
                    IF(NOT Attribute[a].TYPE==Object) THEN
                      Foreach Value[v] do
                          Node[n] Attribute[a]← Value[v]
                    END
                 END
            Add(Node[n], Attribute[a], Value)
            SaveOntology← PATH
          END
       END
   END
END
generate: .OWL file
```

5.4.3 Structured data mapping

Structured data is a form of data that support making queries and get the required chunks of data from it. Examples of such data are relational databases, graph databases, and knowledge-base. In this study, we considered and used only the knowledge-base due to the research scope. Algorithm 3 depicts the mechanism of mapping the existing knowledge-bases in the core ontology (a component of ChaMining system). The existing ontologies are reviewed before utilised in the chaMining system. We only found three relevant ontologies (SweetoDblp, FOAF and IMDB) and these ontologies are re-used to evaluated in our model. The examples of the data chunks of SweetoDblp and IMDB ontologies are discussed in detail in Chapter 6. Further, the existing ontology classes, properties, and relationships are mapped to the core ontology to produce a customised ontology. The generated customised ontology is saved to use for the next component of the ChaMining system.

Algorithm 3: The structured data mapping algorithm

```
Input: .owl file or knowledge_base
Output: updated knowledge-base
Import, load amd extend Core Ontology
Foreach coreOnt[c] do
    Foreach Node[n] do
         IF(coreOnt[c] == Node[n]) THEN
                 Add(Node[n], coreOnt[c])
            End
         SaveOntology← PATH
     END
 END
Return Ontology
```

The input of this algorithm is .owl or .rdf file. Therefore, this algorithm does not do a formal codification of knowledge. This algorithm only extends the core ontology to align the existing classes and properties according to the domain of discourse. This algorithm needs human expertise in alignment or extending the core ontology classes and properties.

5.5 Knowledge Engineering Layer

Knowledge engineering is the second layer of the ChaMining system. This support the knowledge engineering process of the input data (unstructured, semi-structured and structured). This layer holds the core ontology component which supports in the knowledge engineering of knowledge associated with any social networking domain (see details in Section 5.4.2). The re-use ontology pool is another component of this layer that facilitates the interaction of the ChaMining system with the existing ontologies (see details in Section 5.4.3). Finally, the generated knowledge-base from this layer is ready for further processing to the change detection layer.

5.5.1 Knowledge engineering process

Knowledge engineering (Kendal & Creen, 2007) is a process to interpret and understand human represented knowledge. It is a way to formalise the human uttered knowledge in computer understandable form. The machines make predictions based on explicit knowledge is defined in the knowledge-base. Semantic Web technologies (such as RDF and OWL languages) are used to create a formal knowledge-base. In this study, OWL is used to demonstrate how DBLP and IMDB data is formalized in the form of a knowledge-base. Further, the details on these knowledge-bases see Section 6.3.

5.5.2 Core Ontology

The core ontology consists of some OWL classes which are associated with each other using OWL properties. Since the OWL classes and OWL properties are independent of each other as compare to other computer languages (i.e. Java, .Net, etc.). So, the existence or non-existence of OWL classes and properties does not impact each other. Therefore, OWL properties can be employed and re-used in any OWL class relationship. The OWL reasoner support traversing the tree or graph structure and infer the relationship which is not explicitly defined in the knowledge-base. For example, Some resources are utilized on activities and their records can be found from the Documentation class. The classes in the core ontology will extend according to the specified domain model. The DHIN class is an aggregation of Activities, Event, Resources, Data_Time, Actor, Document, Type, Venue, and Activities classes (see Figure 19).

The details of these classes are given below. The Core ontology has some properties (e.g. hasActor, hasDoc, hasType, etc.) and these properties are used to link different classes with each other. These properties will be extended with sub-properties that are associated with the domain of discourse (e.g. DBLP, IMDB, etc).

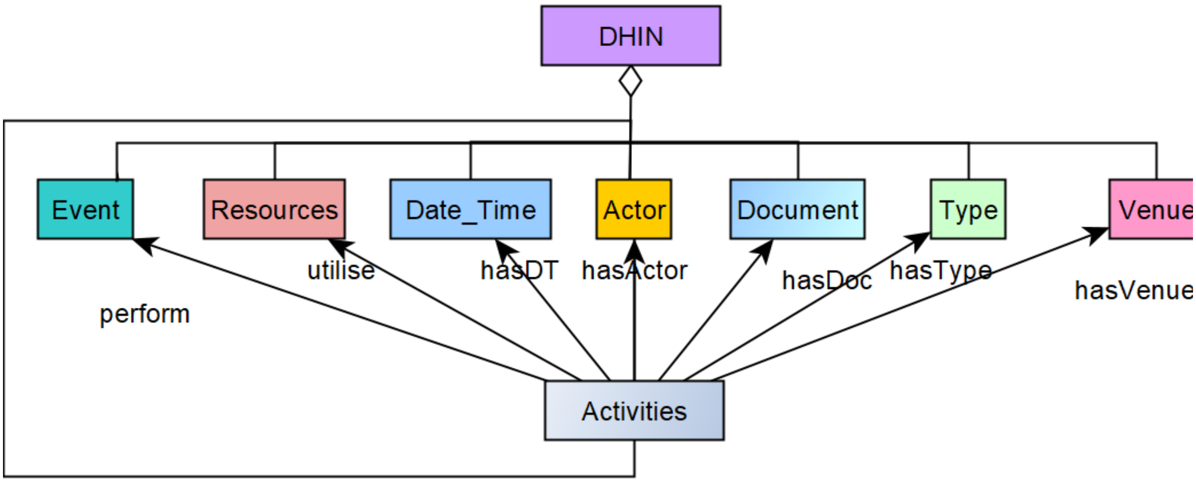


Figure 20 The Core ontology of ChaMining framework

5.5.2.1 Activities

The Activities class is a key component in the core ontology. All the classes of ontology are directly linked with this class. It means that its implemented or extended classes can be retrieved to access core knowledge of the domain. The following constraints are defined in this class.

Class: Activities

SubClassOf:

perform only Event

utilise only Resources

hasDT only Date_Time

hasActor only Actor

hasDoc only Document

hasType only Type

hasVenue only Venue

5.5.2.2 Event

The Event class represents various events that are performed on certain activities. Events may be organised by an organiser (Actor) and certain resources (printing, censoring, editing publishing, marketing) may be utilised for managing activities (i.e. paper writing, blogging, movies, etc.). The Event class is associated with the Activities class is using the "perform" property.

5.5.2.3 Resources

This class is designed to hold the resources which may be allocated for managing activities. The Resources class is linked with the Activities class by using the "utilise" property.

5.5.2.4 Date_Time

The Data_Time class is modelled to hold information about the time and date of an event is occurred and the activities are organised. The "hasDT" property is designed to link the Data_Time class with Activities classes. This property is designed as a functional property. It is a class or instance that can have only one relationship with data and time literal via using this property.

5.5.2.5 Actor

This is the key class of the core ontology. This represents a human entity that organize different activities in the domain of discourse. An actor may be an organizer, sponsor, writer, player, and reader. This class is linked with the Activities class using "hasActor" property.

5.5.2.6 Document

The Document class represents different documents used in certain activities. The scripts of movies, contracts of suppliers, and tenders of companies are some examples of documents. The "hasDoc" property is designed to link the Document class.

5.5.2.7 Type

This class keeps the concept of the types of activities that are organised. For example, a conference is organised for discussion and presentation of a paper. The “hasType” class is utilised to hold information about the type of activity.

5.5.2.8 Venue

The Venue class is designed to represent the place for activities and events. The “hasVenue” property is used to link the venue information with the activity class.

5.5.3 Temporal Knowledge-base

The temporal knowledge-base keeps the knowledge that has a triplet structure based on the RDF. An RDF expression can be defined using a subject, predicate, and object. For instance, the Titanic movie has James Cameron director. The following constraint will be designed to model this information.

```
Class: Titanic
        SubClassOf:
                hasDirector some James Cameron
```

Temporal knowledge-base represents human understanding and interoperability in the machine-understandable form. This knowledge-base is distinct due to keeping the knowledge of date and time information. Such knowledge-base holds a set of information that may have various time stamps. Based on different time stamps the OWL reasoner infers or tracks the changes in the knowledge-base. For example, Bad Boys (a Hollywood movie) movie is pictured in 1985 and again in 1995, a movie is also pictured with the same name. If such information is added in the knowledge-base with some other information such as story, actors, actresses, scenes, director, then it is easier for a human being to find out the differences between them. Similarly, the OWL reasoner imitates human reasoning power and infer the required knowledge from the knowledge-base.

5.5.4 Re-Use Ontology Pool

This component of the system holds the existing structures of the knowledge-base. For example, in this study, we found SweetoDblp, FOAF and IMDB knowledge-bases. These knowledge-bases are designed for three different knowledge domains, for example, SweetoDblp is designed for the domain of digital bibliography and library project, the FOAF is developed for describing persons, their activities and relations to other people and objects and IMDB is for the internet movie database. These knowledge-bases are rich in their structures and terminologies. The only limitation we found in these existing structures is that they do not contain time-related information. Therefore, we are unable to portray some results from the data of these structures. The only reason for adding this component to the ChaMining system is to re-use the existing structure rather than reinventing the wheel.

5.5.5 Temporal knowledge Graphs snapshots

In this stage of the model processing, temporal knowledge which can be extracted from the temporal knowledge-base will be decomposed into G_1, G_2, …G_N temporal graph sequences. These temporal graph sequences will be based on links with associated timestamps and then will be used to discover changes between two-time stamps T_n and T_{n+1}. As in this research, it is the aim to discover local, global and community-level changes. Thus, this study will divide the temporal knowledge graphs into reduced graphs based on nodes, communities and meta-path constraints specified by the user. For example, let's take academic domain dynamic heterogeneous information network, nodes may be classified into three types: authors (A), paper (P) and venue (V). The edges of this DHIN represent co-authors relationship between authors and also shows the venue where they have published or presented their paper. Furthermore, each node in this network also contains various dynamic information dates and times. For example, Figure 21 represent temporal knowledge graphs at two-time stamps with user-specified meta-paths. As shown in figure 21 at time-window T1 authors A1, A2, A3 and A4 have published P1, P2 and P3 at three different venues KDD, DS and DB respectively. Moreover, at time-window T2 a new paper p4 has also been published by authors A3 and A2 in DB venue which shows the difference between these two temporal graphs. In the next stage, these knowledge graphs will be used to extract node, community and meta-paths based graphs.

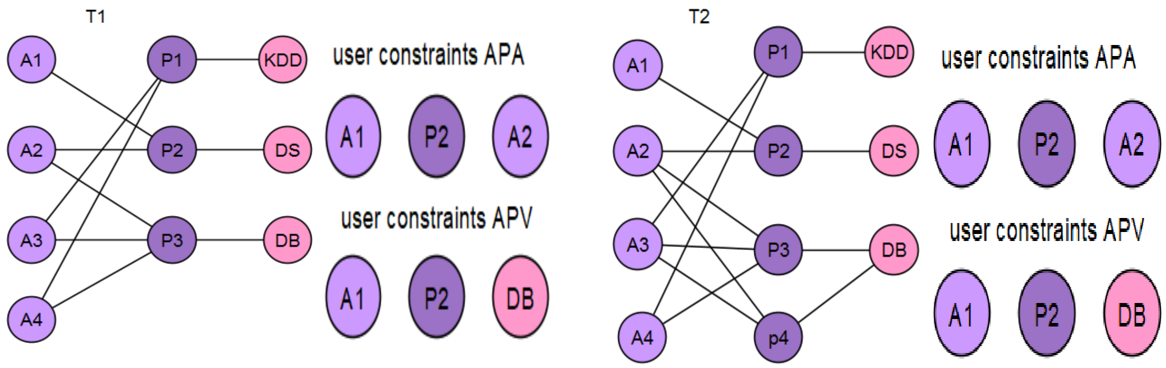


Figure 21 Temporal knowledge graphs with meta-paths

5.6 Change Detection layer

The knowledge graphs developed in the previous layer will be utilised in the change detection layer of the proposed system. This layer is the third main component of the proposed system. This component is further divided into data mining and similarity measure subcomponents. The change discovery solution using knowledge graphs, proposed in this study operates in four steps:

1) Extracting node oriented relational graph patterns P_n of each time observation T_{n+1} where $n=0,1,2\ldots\ldots\infty$.
2) Extracting community-oriented relational patterns C_n of each time observation T_{n+1} where $n=0,1,2\ldots\ldots\infty$.
3) Finding stable patterns from the node and community-oriented graph patterns extracted step-1.
4) Finding Change patterns from the node-based and community-based graph patterns

5.6.1 Extract node relational graph patterns

This research has considered all nodes in the temporal knowledge graph as units of analysis to extract node relational graph patterns. Moreover, all the nodes have been given equal importance instead of using frequent nodes used by research (Loglisci et al., 2015) to extract their relational node based patterns. Our proposed method provides plenty of benefits such as it reduces the loss of information as this research used each node to extract relational patterns associated with each node. For example, Figure 22 shows the node relational graphs patterns extracted from the T1 knowledge graph and Figure 23 represents node relational graph patterns extracted from the T2 temporal knowledge graph. It can be seen in the left part of Figure 22 shows that node A2 have relation with A4, A3, P3 and P2, nodes P2 and P3 also have links with nodes DB and DS. So, these nodes will also be extracted in the node A2 relational graph pattern. Similarly, the relational graph for node A1 contains A2, P2 and DS edges. The node A4 in the T2 temporal knowledge graphs have A1, P1, P3, DB and KDD extracted node based relational graph patterns. The system also generates relational graph patterns for each node available in the temporal knowledge graph using Algorithm 4.

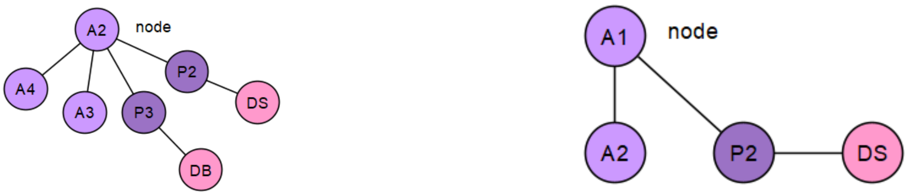


Figure 22 T1 Knowledge graphs decomposition in node relational graphs.

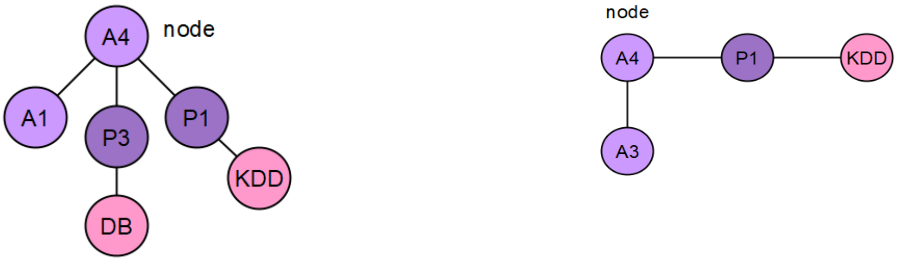


Figure 23 T2 Knowledge graphs decomposition in node relational graphs.

5.6.2 Extract Community relational graph patterns

The community relational graphs are also very important to help the user understand the structural and community-level changes in the dynamic heterogeneous network. This component of the proposed framework will support the user to extract community-oriented graphs patterns from temporal knowledge graphs. For example, Figure 24 depicts the two community-level relational graphs generated from the T2 temporal knowledge graph. These community-oriented graphs have two communities KDD and DB which consist of A3, A4 and P1 and A2, A3, A4, P3, and P4 nodes respectively. The community-level relational graphs will be extracted using Algorithm 4.

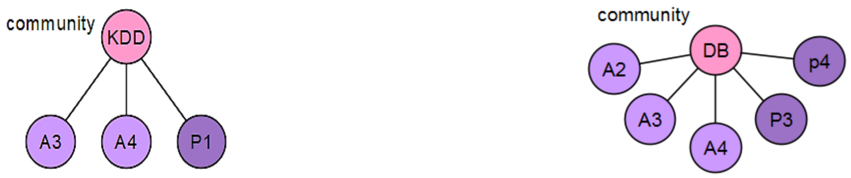


Figure 24 Knowledge graphs decomposition in community relational graphs.

Algorithm 4 is designed and developed for the translation of a knowledge-base into a CSV file. The purpose of this translation is to organise the knowledge in a meaningful way to support further data mining stage. The specified ontology file is added to the system by the user. Then the OWL reasoner extracts all OWL classes, properties and relationships which are iterated, evaluated and added in the CSV file on a given path.

Algorithm 4: The knowledge-base translation to CSV algorithm

```
Input: .owl file or knowledge_base
Output: .CSV file
Foreach OWLClass class OWLClassSet do
   Foreach OWLClassAxiom axiom OWLClassAxiomSet do
       IF (axiom(DataProperty)) THEN
              dataValue←axiom.getValue()
              writeCSV(class, dataValue)
       END
       IF (axiom(ObjectProperty)) THEN
              objectValue←axiom.getObject()
              writeCSV(class, dataValue)
       END

  END
END
   SaveCSV← PATH
Return .CSV
```

5.7 Data Mining

After the temporal knowledge graphs are decomposed into reduced subgraphs, the framework will develop temporal comma-separated versions of nodes and edges between the extracted, node oriented, community-oriented and meta-paths based graph patterns. As data mining is the process of discovering patterns in a dataset (Han, Kamber, & Pei, 2011) in this component of the framework the data mining methods such as clustering, association rule mining and link analysis will be used to extract changes in the node, community and meta-paths based graphs.

5.7.1 Clustering to find community-level changes

This component of the framework will use data mining method clustering to find community-level changes in the temporal knowledge graphs. Since we have already formatted the data in the form of node graph documents and community node graph documents, these will be used as the input to find the community-level changes in this phase of the research. Clustering is the process of separating a set of input graph data into subsets, where objects in each subset are considered interrelated by the similarity between them(Schaeffer, 2007). This process will provide the macro (community) level changes in the two different temporal snapshots of the DHIN subgraphs. There are two kinds of graph clustering, between-graph and within-graph clustering. The former approach will divide a set of graphs into different clusters, while the later approach groups the nodes of the same graph into clusters. Moreover, other clustering methods available in the research are, for example, shared nearest neighbour(Ertoz, Steinbach, & Kumar, 2002), highly connected components(Hartuv & Shamir, 2000), maximal clique(Rysz, Pajouh, & Pasiliao, 2018), and k-means graph clustering, hierarchical clustering and expectation-maximization algorithms. In this research since we are interested in finding the community-level changes which users can also understand so we will use two methods to find these changes in the temporal reduced graph documents. The methods we will use are:

1. Hierarchical Clustering
2. Expectation-Maximization Clustering

Hierarchical clustering: We will use Hierarchical clustering using the Ward minimum variance method. In this method, the distance between two clusters is calculated using equation (1).

$$\frac{(u_1 - u_2)'(u_1 - u_2)}{1/n_1 + 1/n_2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Where u_1 and u_2 are cluster means and n_1 and n_2 are the cluster frequencies. In each calculation step of clustering using ward the following steps are performed(Szmrecsanyi, 2012):

1. Calculate the mean of each cluster
2. Find cluster's mean by calculating the distance between the individual object in a cluster and then square the difference.
3. Add up the squared values from step 2
4. Sum all the sums of squares from step 3.

Expectation-Maximization Clustering: The expectation-maximization clustering algorithm considers that a mixture model approximates the data distribution by fitting k cluster density functions. The mixture model probability density function at a point x is a calculation by the formula in equation (2).

$$p(x) = \sum_{k=1}^{k} w_k f_k(x \mid \mu_k, \Sigma_k) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2)$$

Where f_k are density functions to the data with d variables. W_k is the proportion of data that belongs to cluster primary cluster k. Moreover, each cluster is modelled by Gaussian probability distribution equation (3)

$$f_k(x \mid \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\frac{1}{2}(x-\mu_k)^T (\Sigma_k)^{-1}(x-\mu_k)\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(3)$$

Where u_k and D_k are mean vector and covariance matrix for each cluster h.

5.7.2 Association Rule mining rule level changes

The Association rule mining (ARM) discover hidden patterns based on shared attributes among a large number of items in the data. ARM tasks create association rules based on support and confidence i.e. the rules must have supported greater than minimum support and confidence greater than specified confidence (Busarov, Grafeeva, & Mikhailova, 2016). Agrawal et al., (1993) are the pioneer of using association rule mining to analyse large datasets. They introduced the idea of using association rules to extract patterns from market basket data. A large number of studies are devoted to pattern discovery using association rule mining based on "candidate generation and test" and "pattern growth method" (Pei et al., 2007). Once the temporal reduced graphs have been created association rule mining method will be employed to find the change in the most interesting rule in different temporal snapshots. The methods that are used in association rule mining are taken from statistics. The Association rule is written

C=>B where C is the antecedent and B is the consequent. The rules can contain more than two items on both sides. Association rule mining has three evaluation criteria; support confidence and lift for relationship discovery between items.

5.7.3 Link analysis to find global changes

This component of the framework will help to find the global level changes in the temporal node and community node graphs. The link analysis part will give the changes between two DHIN temporal reduced snapshots by determining parameters such as out-degree of node(Menichetti, Dall'Asta, & Bianconi, 2014), cluster nodes intensity, eigenvector centrality(Ruhnau, 2000), closeness centrality and influence centrality between them. These parameters will be used between two-time windows of the DHIN temporal reduced graphs to see if any changes have occurred between two temporal snapshots of the network t_1 to t_2.

5.7.4 Similarity Measure

The second component of the change detection layer is similarity measure. The similarity measures computations are very important and have various applications in different domains of science such as; computer vision, image processing, social networks, chemical networks and biological networks. Therefore there are numerous algorithms available for graph similarity measures. The graph similarity measures techniques are categorised into four main categories: graph isomorphism, feature extraction, iterative methods and belief propagation-based methods. This research has surveyed the graph similarity measures and it is decided to use the Jaccard Similarity measure because the data in this research is sparse graphs and in the form of binary vectors(Koutra, Parikh, Ramdas, & Xiang, 2011). The data is already in the form of node and community-based graphs so this research will use graph similarity measure Jaccard coefficient to check the similarity between graphs of two timestamps T_n and T_{n+1} respectively, where n=0,1,2,3,…. ∞. The similarity measure calculates the similarity between two graph sets and then giving a minimum similarity threshold stable and change patterns can also be extracted from the given graph patterns. The similarity of the patterns is calculated using Algorithm 6.

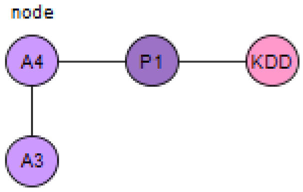


Figure 25 Set of nodes linked to A2 at T1

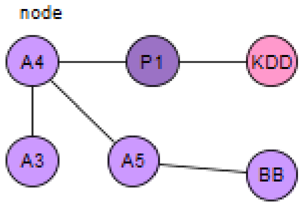


Figure 26 Set of nodes linked to A2 at T2

For example, Figure 25 and Figure 26 shows two-node oriented graphs extracted from temporal knowledge graphs at two different periods, then Algorithm 6 will calculate similarity using the Jaccard coefficient. So A4 node have edge set of {P1,KDD,A3} at T1 and {P1,KDD,A3,A5,BB} at T2 respectively. Using algorithm 6 the similarity will be 3/5*100=60%.

```
**Algorithm 5** Similarity between two graph patterns
**Input:** $G_{Tn}$, $G_{Tn+1}$ Where n=0,1,2,…. ∞// Node or community graphs at two time stamps
**Return**: $C_w$ //calculated similarity coefficient
**If** (getlabel($G_{Tn}$) == getlabel($G_{Tn+1}$)) **Then** // Node or community are same
$E_i$=getedges($G_{Tn}$) $E_j$=getedges($G_{Tn+1}$)
      **If** (length($E_i$!=0) and length ($E_j$!=0)) **Then**
        $C_w[k]=\frac{Ei \cap Ej}{Ei \, U Ej}$ *100//Jaccar similarity coeffiient
**Else**
$G_{Tn}$++, $G_{Tn+1}$++
Else
**Return** $C_w$L list
```

5.7.5 Stable patterns discovery

In this phase-stable patterns will be extracted from two consecutive node or community-oriented graphs. Stable patterns are the combination of two temporal graph patterns which have greater than or equal to the similarity coefficient specified in the minimum similarity threshold. The stability between the patterns G_{Tn}, G_{Tn+1} is quantified by the similarity value between their edges. Algorithm 7 elaborates how stable patterns are combined and extracted. To extract stable patterns by using temporal graphs it is important to have node or community graphs extracted from temporal knowledge graphs of two time periods T_n and T_{n+1} where n=0,1,2,3…..∞. In the first part, the algorithm takes two graph patterns G_{Tn}, G_{Tn+1} with a minimum similarity threshold min_{Th} . In particular, in the first stage algorithm checks if the labels of the root nodes of the G_{Tn}, G_{Tn+1} are the same, if the root node is the same it generates sets of edges or vertices for G_{Tn} and G_{Tn+1}. For each pair of graph patterns that either has the same vertice length or different vertices length, the algorithm computes the similarity using Algorithm 6. If the calculated similarity found in the similarity coefficient is greater than or equal to the minimum threshold the algorithm will combine the vertices of both graphs using union operation of set theory with their similarity coefficient and Stable keyword. For example, if we have DS={P2, A1, A2} the community-oriented graph pattern extracted from the T1 knowledge graph and DS={P2, A1, A2} is the community-oriented graph pattern extracted from the T2 knowledge graph. Then by using these two community-oriented graphs at two different timestamps and min_{Th} =95% we can see algorithm will return a stable pattern P_s=({P2, A1, A2},100%, Stable).

Algorithm 6 Stable patterns discovery

```
Input: $G_{Tn}$, $G_{Tn+1}$, $min_{Th}$ Where n=0,1,2,…. ∞// Node or community graphs at two time stamps
Return: $P_s$ //Stable patterns List
for ($G_{Tn}$, $G_{Tn+1}$) do
If (getlabel($G_{Tn}$) == getlabel($G_{Tn+1}$)) Then // Node or community root label are same
$E_i$=getedges($G_{Tn}$) $E_j$=getedges($G_{Tn+1}$) Then
If length($E_i$)==length($E_j$) | | length($E_i$)==length($E_j$) Then
        $C_w$=compute_similarity($G_{Tn}$, $G_{Tn+1}$)
If $C_w$>= $min_{Th}$
$P_s$[z]= combine($E_i$ U $E_j$, $C_w$, Stable)
Return $P_s$ list
```

5.7.6 Node level or local Change patterns discovery

In this component of the framework node level or local changes are discovered. Each change represents the difference between the edges or vertices of a node or community-oriented patterns extracted from temporal knowledge graphs. This change discovery component is in charge of generating change patterns from temporal node and community-oriented graph patterns between two timestamps. Algorithm 7 is used to find the changes by quantifying the similarity value between edges of two subgraphs at different times. A changing pattern will be extracted from Algorithm 7 if the similarity score is less than the specified minimum threshold (min_{Th}). In this algorithm, each pair of graph patterns that have same length will be checked whether the similarity value difference between their edges is less than or greater than the minimum threshold. In case the similarity value is less than min_{Th} then both the graphs will be combined and different nodes are also specified in the change patterns list.

Algorithm 7 Change patterns discovery

```
Input: $G_{Tn}$, $G_{Tn+1}$, $min_{Th}$ Where n=0,1,2,…. ∞// Node or community graphs at two time stamps
Return: $P_c$ //Changed patterns List
for ($G_{Tn}$, $G_{Tn+1}$) do
If (getlabel($G_{Tn}$) == getlabel($G_{Tn+1}$)) Then // Node or community root label are same
$E_i$=getedges($G_{Tn}$) $E_j$=getedges($G_{Tn+1}$) Then
If length($E_i$)!=length($E_j$)
    $C_w$=compute_similarity($G_{Tn}$, $G_{Tn+1}$) //Using Algrithm 6
If $C_w < min_{Th}$
$P_c$[z]= combine($E_i$, $E_j$ (Diff=$E_i \cup E_j - E_i$), $C_w$, changed)
Return $P_c$ list
```

For example, figure 25 and figure 26 shows two-node oriented graphs extracted from temporal knowledge graphs at two different periods, then Algorithm 7 will first check if the labels of the nodes are the same or different. If the node labels are the same then the edges of the nodes will be extracted by creating edge sets. So A4 node have edge set of {P1,KDD,A3} at T1 and {P1,KDD,A3,A5,BB} at T2 respectively. Using algorithm 6 the similarity will be 3/5*100=60% which shows that it is less than the specified minimum threshold (say 95%) so it is considered as a changed pattern. Hence using the set theory subtraction operations it will find the difference {P1, KDD, A3}-{P1, KDD, A3, A5, BB}={A5, BB} which will be combined with the similarity measure and change the label.

This research has developed and implemented 7 algorithms to discover changes in dynamic heterogenous information networks. The first three algorithms, Algorithm 1, 2 and 3 have been modified and reused in this research (see details in Section 5.4.1, 5.4.2, and 5.4.3 for modification details and contributions). Moreover, Algorithm 5 have also been modified to calculate the similarity measure using Jaccard similarity formula (See details 5.7.4). Furthermore, there are three other algorithm (algorithms 4, 6 and 7) in this research which are originally developed (see details in sections 5.6.2, 5.7.5 and 5.7.6) from scratch to detect changes in dynamic heterogenous information networks.

5.8 Summary

In this chapter, a representative scenario was presented to explain the possible abilities of the change discovery system. ChaMining framework based on three layers was purposed. The proposed system is novel and facilitates change discovery from DHINs data. Each component is presented with explained examples to understand the concepts behind the proposed system. The system is based on knowledge engineering and data mining concepts to discover changes.

In the next chapter, the proposed ChaMining system will be implemented and applied on DBLP data set to test the capabilities.

Chapter 6 The Development and the Implementation of the ChaMining Framework

6.1 Introduction

Chapter 5 have proposed a new framework to detect change in DHINs, this chapter presents the implementation of the proposed system. The implementation shows how this system can support the use of many knowledge models and translate them into change pattern discovery. This chapter also demonstrates the extensibility of the ChaMining framework on different domain-specific data sets. For example, social networking data, digital library data, internet movie database data, etc. This chapter elaborates on the implementation of all the layers of the ChaMining system discussed in Chapter 5. The chapter is organised as follows:

- Section 6.1 represents the tools and standards used for the development of the proposed system
- Section 6.2 describe the datasets used for temporal knowledge-based development
- Section 6.3 demonstrates the implementation of the data acquisition and pre-processing layer and its components.
- Section 6.4 presents the implementation of data translation to knowledge-base
- Section 6.5 shows an extension of the Core ontology and generation of knowledge-bases for DBLP, IMDB and ENRON data corpus.
- Section 6.6 expands the change mining layer and its related components.
- Section 6.7 will present the summary of this chapter.

6.2 Development Tools

As described in Chapter 5, this study proposes a novel framework for change discovery in DHINs. The proposed system named “ChaMining” has been mainly implemented using Java programming. ChaMining is a console-based system that has been developed with Java 8 and Eclipse open-source software and other different tools, techniques and standards. The knowledge base is developed using Java owl API libraries and can be viewed and modified using Protégé ontology editor. This ontology editor is open-source and offers a user-friendly interface for the development and management of ontologies. The data mining part of the framework has been implemented in SAS programming.

6.3 Dataset description

Four different domains datasets are used in this research for demonstrating the implementation of the ChaMining algorithm. Their details are presented in sections 6.3.1 to 6.3.4 and Table 6.1 shows a summary table to compare the datasets' specifications and characteristics.

6.3.1 Digital Bibliography and Library Project (DBLP)

DBLP[13] is a computer science bibliographic information network, such network a typical heterogeneous network containing three types of information entities: papers, venues, and authors. Each paper has links to a set of authors, and a venue and these links belong to a set of link types. To understand the object types and link types better in a complex heterogeneous information network, it is necessary to provide the meta-level (i.e., schema-level) description of the network. Therefore, the concept of meta path is proposed to describe the meta structure of a network.

6.3.2 Internet Movie Database (IMDB)

IMDB[14] is an online database of information related to films, television programs, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews. The data used in this research covers the IMDB database network from 1894 to 2019 straddling across 125 years. The data is very large, so I have divided it into 5 equal parts to create the ontologies and then use it for data mining methods.

6.3.3 Enron Email

Enron Email[15] dataset was used and prepared by CALO (A Cognitive Assistant that learns and organizes) project. It has data from about 150 users and the corpus contains around 0.5 Million messages. It contains four different versions, March 2, 2004; August 21, 2009; April 2, 2011

[13] https://dblp.org/
[14] https://www.imdb.com/interfaces/
[15] https://www.cs.cmu.edu/~enron/

and May 7, 2015 version. We have used the latest one. Enron Email communication network covers the email communication between different users at different time windows. It was made public by the Federal regulatory commission during its investigation. Nodes of the network represent email addresses. There is an undirected graph from edge *i to j*, when an email address node *i* sent at least one email message to address *j*. The email addresses were properly hashed and given a numeric number to analyse.

6.3.4 High Energy Physics Citation Dataset

This dataset (High Energy Physics Phenomenology) is a citation graph from the e-print arXiv and covers all the citations within the network from January 1993 to April 2003 spanning 124 months. It starts within a few months of the inauguration of the arXiv, hence presenting the complete history of the High-energy physics section. In this network, the graph contains the undirected graph from *i to j* if an author *i* have cited j.

6.3.5 Direct vs undirected graph data

This research has employed directed and undirected graphs for experiments. Undirected graphs do not have a direction, these graphs represent two-way relationships, and each edge can be traversed in both directions. Directed graphs have edges with directions and show a one-way relationship, in these types of graph data edges can only be traversed in a single direction (Diao, Farmani, Fu, & Butler, 2014).

6.3.6 Unstructured, semi-structured and structured data

The proposed system has also been implemented and tested for Unstructured (see details in 5.4.1), semi-structured (see details in 5.4.2) and structured data (see details in 5.4.3) of dynamic heterogenous information networks. Table 6.1 shows the summary and comparative specifications of datasets used in this research. The symbol √ shows that data is used in this specification and X shows that data is not used in this format. For example, table 6.1 shows that Enron email data has been used as unstructured data with directed edges between nodes of the data.

Data Set Name	Directed	Undirected	Structured	Semi-Structured	Unstructured
DBLP	√	X	√	√	X
IMDB	X	√	√	√	X
Enron Email	√	X	X	X	√
High Energy Physics Citation	√	X	√	X	X

Table 6.1 Specification and format of used datasets

6.4 The implementation of data translation to knowledge-base

In this section, the implementation of the second layer of the ChaMining framework is discussed in detail. Three different algorithms are designed for the translation of unstructured, semi-structured and structured data to a knowledge-base (see details in Chapter 5 and Section 5.2). The translation of the unstructured and structured data is a more challenging task where user analysis and evaluations are continuously needed. The unstructured data needs correct classifications of terminologies and on the other hand the structured (i.e. existing) knowledge-base need analysis to extract the relevant terminologies. All these unstructured, semi-structured and structured knowledge-base are designed to extend the core ontology. Figure 27 shows the implementation of the core ontology and its design overview in the protégé editor.

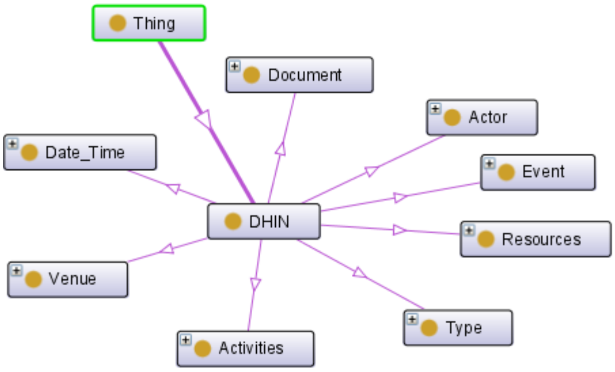
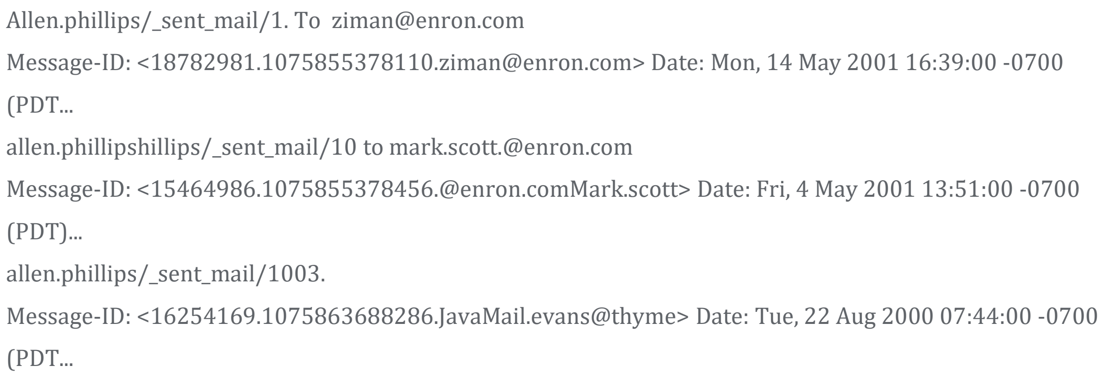


Figure 27 Overview of core ontology in Protégé editor

Figure 27, presents an example of unstructured data and Figure 6.4.3 depicts the translation of the unstructured data into the knowledge-base.



Allen.phillips/_sent_mail/1. To ziman@enron.com
Message-ID: <18782981.1075855378110.ziman@enron.com> Date: Mon, 14 May 2001 16:39:00 -0700 (PDT...
allen.phillipshillips/_sent_mail/10 to mark.scott.@enron.com
Message-ID: <15464986.1075855378456.@enron.comMark.scott> Date: Fri, 4 May 2001 13:51:00 -0700 (PDT)...
allen.phillips/_sent_mail/1003.
Message-ID: <16254169.1075863688286.JavaMail.evans@thyme> Date: Tue, 22 Aug 2000 07:44:00 -0700 (PDT...

Figure 28 An example of unstructured Enron email data

An example is extracted from Enron data unstructured text corpus which is modelled in the ontology by extending the core ontology.

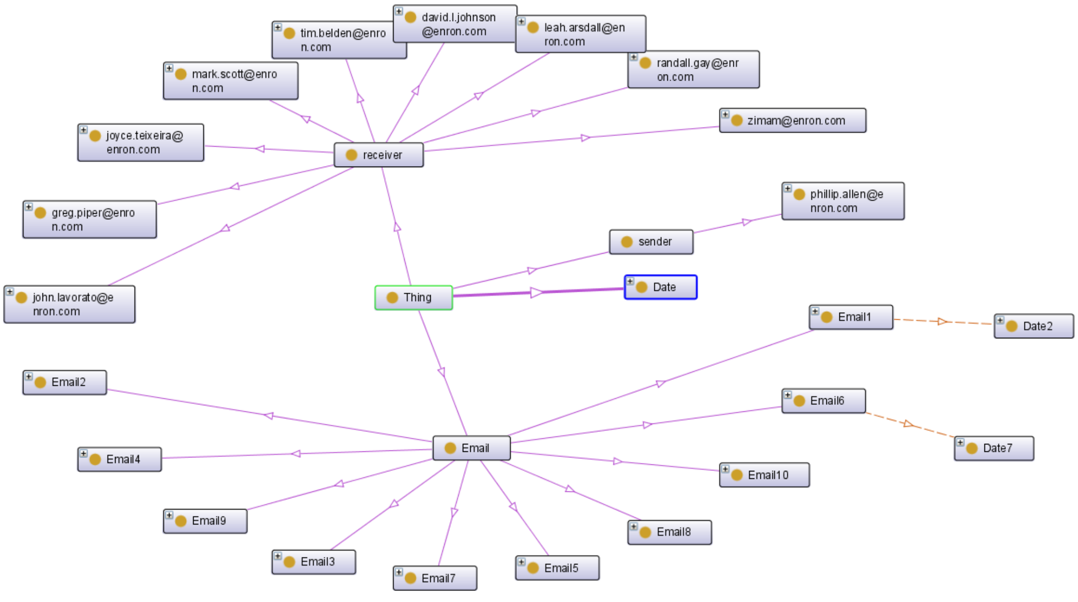
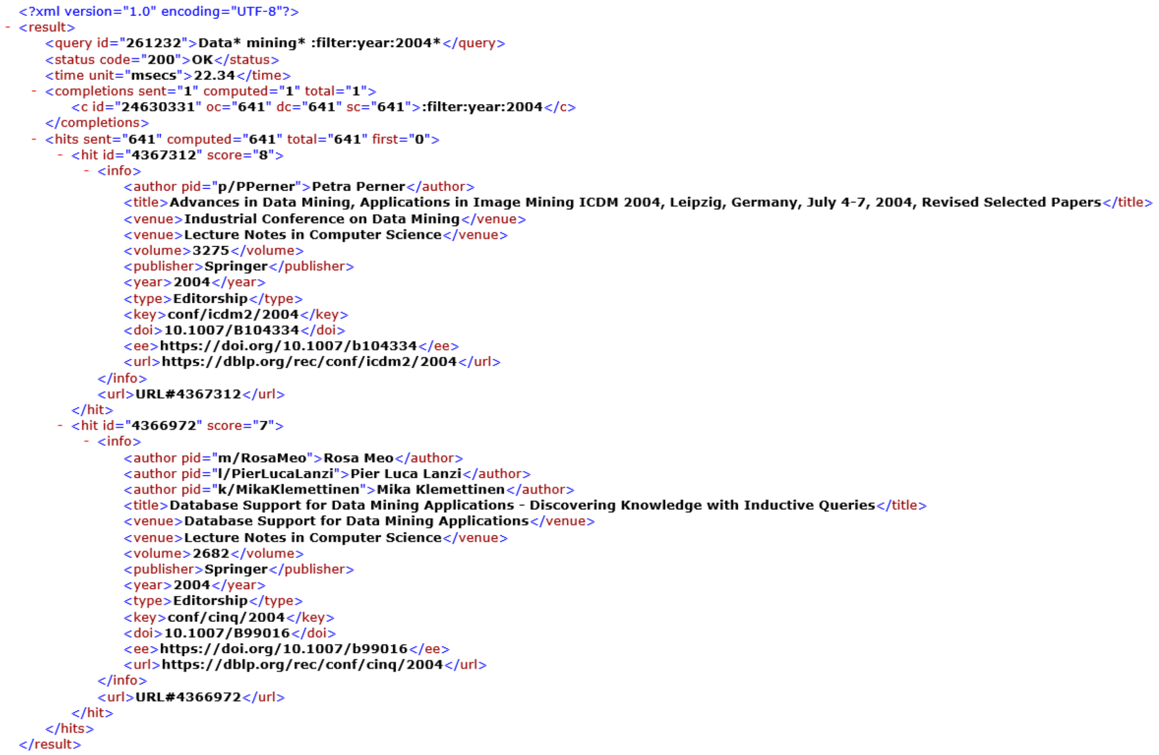


Figure 29 An example of knowledge-base developed from unstructured data

Further, the implementation of the semi-structured data is illustrated in Figure 28 and Figure 29.



```
  <?xml version="1.0" encoding="UTF-8"?>
- <result>
    <query id="261232">Data* mining* :filter:year:2004*</query>
    <status code="200">OK</status>
    <time unit="msecs">22.34</time>
  - <completions sent="1" computed="1" total="1">
      <c id="24630331" oc="641" dc="641" sc="641">:filter:year:2004</c>
    </completions>
  - <hits sent="641" computed="641" total="641" first="0">
    - <hit id="4367312" score="8">
      - <info>
          <author pid="p/PPerner">Petra Perner</author>
          <title>Advances in Data Mining, Applications in Image Mining ICDM 2004, Leipzig, Germany, July 4-7, 2004, Revised Selected Papers</title>
          <venue>Industrial Conference on Data Mining</venue>
          <venue>Lecture Notes in Computer Science</venue>
          <volume>3275</volume>
          <publisher>Springer</publisher>
          <year>2004</year>
          <type>Editorship</type>
          <key>conf/icdm2/2004</key>
          <doi>10.1007/B104334</doi>
          <ee>https://doi.org/10.1007/b104334</ee>
          <url>https://dblp.org/rec/conf/icdm2/2004</url>
        </info>
        <url>URL#4367312</url>
      </hit>
    - <hit id="4366972" score="7">
      - <info>
          <author pid="m/RosaMeo">Rosa Meo</author>
          <author pid="l/PierLucaLanzi">Pier Luca Lanzi</author>
          <author pid="k/MikaKlemettinen">Mika Klemettinen</author>
          <title>Database Support for Data Mining Applications - Discovering Knowledge with Inductive Queries</title>
          <venue>Database Support for Data Mining Applications</venue>
          <venue>Lecture Notes in Computer Science</venue>
          <volume>2682</volume>
          <publisher>Springer</publisher>
          <year>2004</year>
          <type>Editorship</type>
          <key>conf/cinq/2004</key>
          <doi>10.1007/B99016</doi>
          <ee>https://doi.org/10.1007/b99016</ee>
          <url>https://dblp.org/rec/conf/cinq/2004</url>
        </info>
        <url>URL#4366972</url>
      </hit>
    </hits>
  </result>
```

Figure 30 An example of semi-structured data

The interesting thing to notice here is that the Paper class is imported from the core ontology and the rest of information is mapped in its subclass (i.e. 4367312). The Author (i.e. Petra-Perner) is also associated with the subclass of the Paper class (see Figure 31).

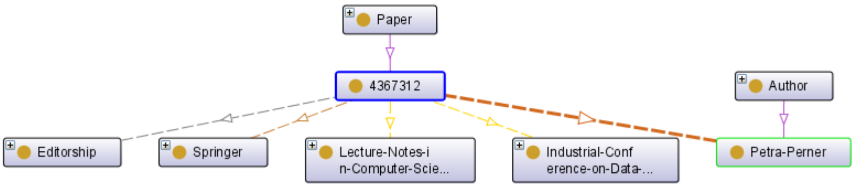

Paper
4367312
Author
Editorship
Springer
Lecture-Notes-in-Computer-Scie...
Industrial-Conference-on-Data-...
Petra-Perner

Figure 31 An example of knowledge-base developed from semi-structured data

Figure 32 depicts a glimpse of IMDB ontology[16]. This is re-used for creating a temporal knowledge-base by importing the core ontology. The IMDB ontology is first analysed and then its associated terms (classes, properties and relationships) are used by extending the core ontology.

[16] https://sites.google.com/site/ontopiswc13/home/imdb-mo

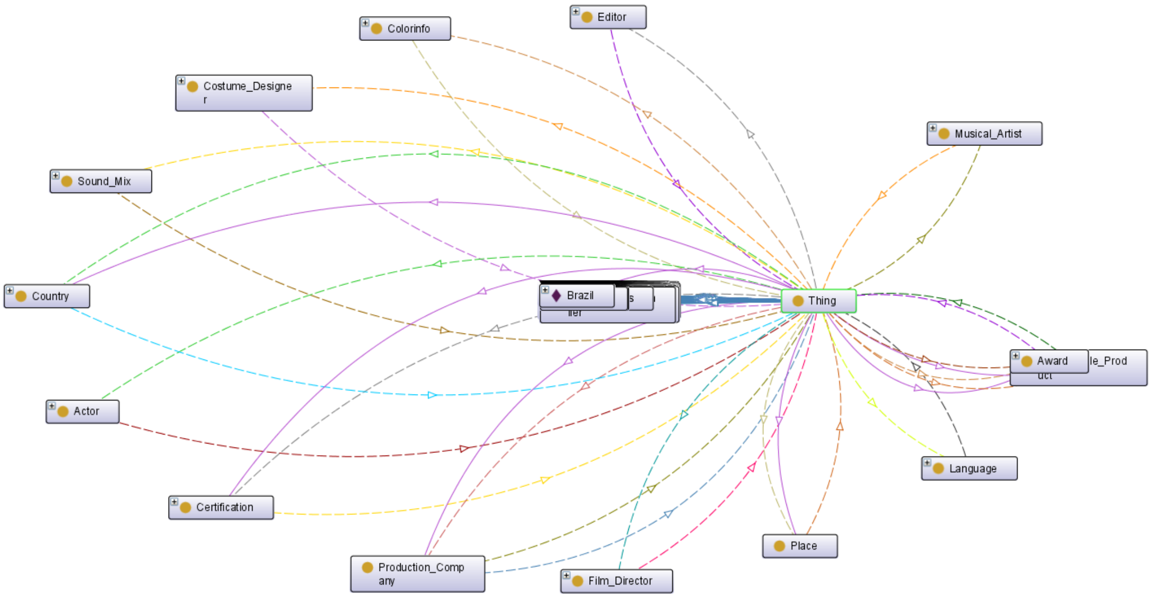


Figure 32 An example of structured data

Figure 32 shows that Smule-L-Jacks worked in different movies as an actor and director. The Film class (used in existing IMDB ontology) conceptualisation is utilised in the Movie class and Actor and Film_Director concepts are also added in the derived ontology. All these concepts are correctly aligned with the concepts of the core ontology. For example, the Movie or Film classes are designed as the subclasses of the Activity class. The Actors and Directors are mapped as the subclasses of the Actor class.

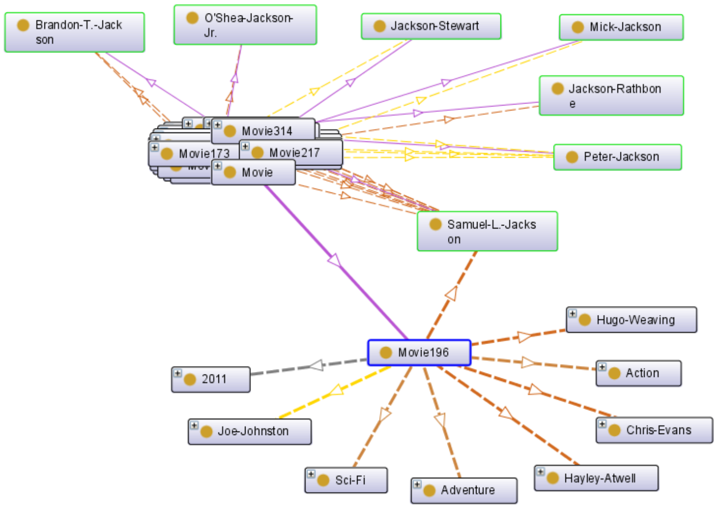


Figure 33 An example of knowledge-base developed from structured data

The extension of the core ontology using DBLP, IMDB and ENRON data is demonstrated in the next section.

6.5 Generating the Temporal knowledge-base for DBLP, IMDB and ENRON by extending the Core ontology

This section presents the details of how the derived ontologies extend the classes, properties, relationships and constraints of the core ontology and what further specialised constraints are needed to define them individually. The classes of the core ontology are extended for the DBLP, IMDB and ENRON domains for developing their temporal knowledge-bases. For generating temporal ontologies the Core ontology is imported and the nodes or data of the DBLP.xml, IMDB.xml and ENRON.txt are iterated. Their temporal knowledge-bases are organised and designed in a semantically precise way and some extra constraints are also added for enhancing their intelligibility. The following constraint in the Movies (a subclass of Activities class) class is an example of it for IMDB temporal knowledge-base.

```
Class: Movies
        SubClassOf:
                hasActor only Actors
                hasDirector only Director
                Activities
```

The purpose of the constraints is to make sure that the correct information is mapped and retrieved about who has directed and performed in a particular movie. This class has relationships with Actors and Directors (Both are the subclasses of Actor class in Core ontology) via using hasActor and hasDirector properties (see details in Cahpter 5 Section 5.4.2). In the next subsections, the extensibility of Core ontology is presented with details.

6.5.1 The extensibility of Activities class

The Activities class is extended according to the different tasks or activities are organised in the DHIN network. Therefore, this class is going to demonstrate next how specific constraints are defined for the temporal knowledge-bases of DBLP, IMDB and ENRON.

6.5.1.1 The Activities for DBLP

The paper is the key activity class in the DBLP and this class extend the Activities class of the Core ontology. This class inherit all the constraints defined in the Activities class (see Figure 34 and see details on Activities class constraint Chapter 5 and Section 5.4.2).

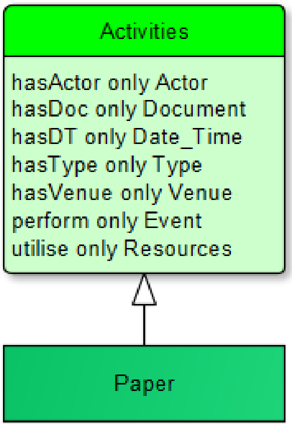


Figure 34 Paper class extended Activities class

Further, the following specialised constraints are implemented in the paper class.

```
Class: Paper
        SubClassOf:
                hasAuthor only Author
                hasPublisher only Publisher
                Activities
```

It means that the paper class has relationships with Author and Publisher classes using has Author and hasPublisher properties which are the sub-properties of has Actor property of Core ontology.

6.5.1.2 The Activities for IMDB

The Movies class extended the Activities class of the Core ontology in the IMDB knowledge-base (see Figure 35).

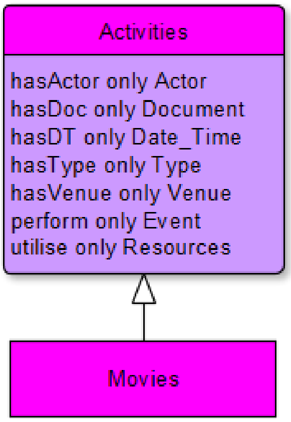


Figure 35 Movies class extended Activities class

Following constraints are add in the movies class.

Class: Movies

SubClassOf:

hasActor only Actors

hasDirector only Director

Activities

The meaning of these constraints is that the Movies class has relationships with the Actors and Director classes via using the hasAuthor and hasDirector properties.

6.5.1.3 The Activities for Enron

In the Enron knowledge-base, the Emails class extends the Activities class and inherit all the constraints defined in the Activities of Core ontology (see Figure 36).

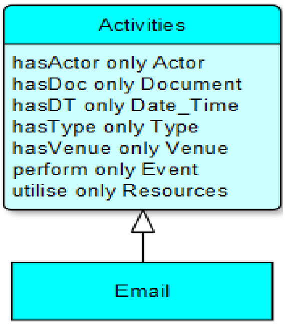


Figure 36 Email class extended Activities class

Class: Emails

SubClassOf:

hasSender only Sender

hasReceiver only Receiver

Activities

The meaning of these constraints is that the Movies class has relationships with the Sender and Receiver classes via using the hasSender and hasReceiver properties.

6.5.2 The extensibility of Actor class

The Actor class represents a human being involved in any activity. An actor can be an author, publisher, actor, actress, director, doctor, patient, sender, receiver, etc. The actor class is extended with a piece of specialised information. For example, an actor can send an email or perform in a movie or one can write a piece of research activity.

6.5.2.1 The Actor for DBLP

The Actor (Core ontology) class is extended in the DBLP temporal knowledge-base with the Author and Publisher classes. These classes are also extended to the constraints defined in the Actor class (see Figure 37). The hasActivity property is the inverse property of the hasActor property. So, if an actor has a relationship with either property then its inverse property help in inferring the required information.

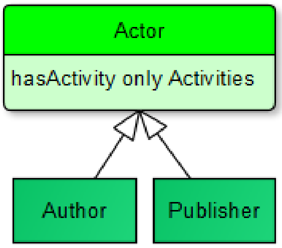


Figure 37 Author and publisher classes extended Actor class

6.5.2.2 The Actor for IMDB

The Actors and Director are the subclasses of the Actor class in the IMDB knowledge-base and these classes also extend the constraints defined on their superclass (see Figure 38).

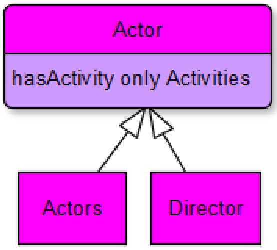


Figure 38 Actors and Director classes extended Actor class

6.5.2.3 The Actor for Enron

The Actor class is extended by the Sender and Receiver classes and these classes also inherit the constraints defined on their superclass (see Figure 39).

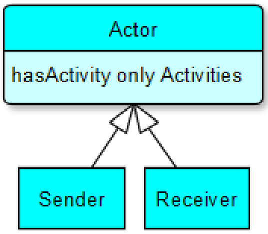


Figure 39 Sender and receiver classes extended Actor class

6.5.3 The extensibility of Date_Time class

The Date_Time class is the key component of the Core ontology. This information support traversing, searching, retrieve and manipulating the information of a DHIN networking in different time frames. In the next subsection, the extension of the Date_Time class will be demonstrated in DBLP, IMDB and Enron knowledge-bases.

6.5.3.1 The Date_Time for DBLP

The Date_Time class is extended by the Year class in DBLP knowledge-base. The constraints of the Data_Time class are also inherited by its subclass (i.e. Year) (see Figure 40). The hasYear property is defined as a sub-property of the hasDT property. This property makes a relationship with dateTime literal value. The following constraint is an example of the information.

```
Class: Year
        SubClassOf:
                hasYear value 2001
                Date_Time
```

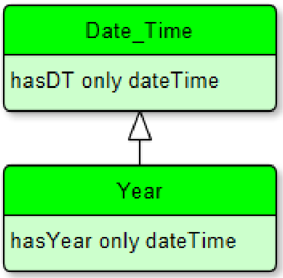


Figure 40 Year classes extended Date_Time class

6.5.3.2 The Date_Time for IMDB

The Date_Time class is extended by the Year class in the IMDB knowledge-base. The constraints of the Data_Time class are also inherited by its subclass (i.e. Year) (see Figure 41). The hasYear property is defined as a sub-property of the hasDT property.

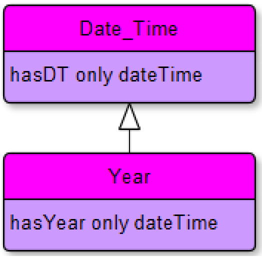


Figure 41 Year classes extended Date_Time class

6.5.3.3 The Date_Time for Enron

The Date_Time class is not extended in the Enron knowledge-base. Therefore, this class is directly used in the Enron knowledge-base. The hasDT property makes the relationship with dateTime literal value. The following constraint is an example of the information (see Figure 42).

```
Class: Date_Time
        SubClassOf:
                hasYear value 2009-05-14 16:39:00-07:00
```

In the next section, the extension of the Core ontology is demonstrated comprehensively. It is also demonstrated how the temporal knowledge-bases (DBLP, IMDB and Enron) has implemented this abstract core ontology in detail.

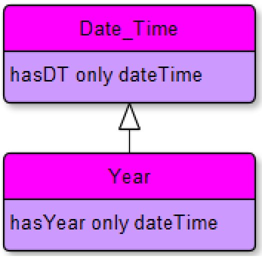


Figure 42 Year classes extended Date_Time class

6.5.4 DBLP knowledge-base

The DBLP knowledge-base is a temporal ontology. The Core ontology is extended for the creation of the DBLP ontology (see Figure 43).

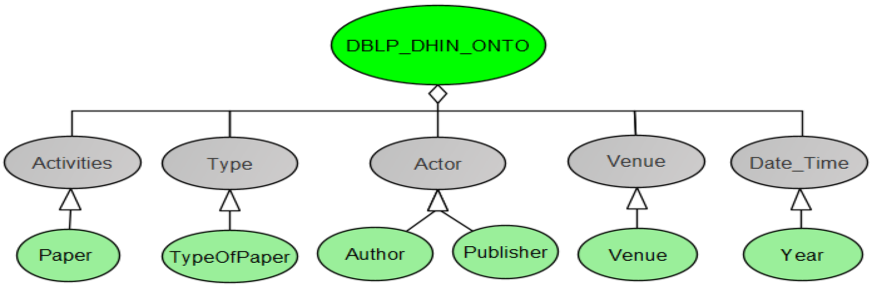
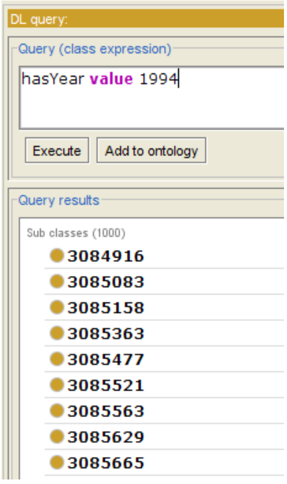


Figure 43 A glimpse of DBLP knowledge-base

The nodes or classes of the Core ontology are represented in grey colour and DBLP knowledge main classes are shown in green colour. Further, these main classes are extended with real-world data. For example, the following constraints are defined on the subclass (i.e. Paper3084916) of the Paper class.

```
Class: Paper3084916
    SubClassOf:
        hasAuthor value “Tom Gruber”
        hasAuthor value “Thomas R”
        hasAuthor value “Gregory R. Olsen”
        hasPublisher value “Morgan Kaufmann”
        hasTitle value “An ontology for engineering mathematics”
        hasYear value 1994
        hasType value Conference
        Paper
```

These constraints help to describe the Paper3084916 activity in detail which helps the OWL reasoner to infer the information which is mentioned in the description logic (DL) query. For instance, a user wants to see all the papers which are published in the year 1994. He will give input 1994 and the following query will be designed and trigger the OWL reasoner to infer the information which is relevant to the input (see Figure 44).



hasYear value 1994

Sub classes (1000)

- 3084916
- 3085083
- 3085158
- 3085363
- 3085477
- 3085521
- 3085563
- 3085629
- 3085665

Figure 44 An example of a DL query and its results in protégé editor

A command-line application is developed using Java that is embedded with OWL API and OWL reasoner libraries. The OWL API libraries support creating, loading, updating and deleting ontologies. The OWL reasoner libraries (in this application Pellet Reasoner (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007) libraries are utilised) support inferring the results of the queries on the data given by the user.

6.5.5 IMDB knowledge-base

The Core ontology is imported to develop the IMDB knowledge-base DHIN network. The main classes are created by extending the Core ontology classes (see Figure 45). This knowledge-base holds all the necessary information which is essential for the movies encyclopaedia. The correct organisation and management of such knowledge-base support in retrieving the right or relevant information.

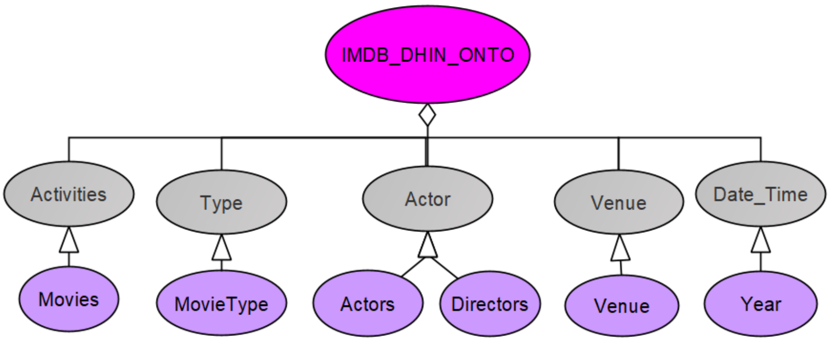


Figure 45 A glimpse of IMDB knowledge-base

Further, the main classes of the IMDB knowledge-bases are extended when data is added to the knowledge-base. For example, Movie1 has defined the subclass of the Movies class and the following constraints are defined on it.

```
Class: Movie1
        SubClassOf:
                hasActors some Bradley-Cooper
                hasActors some Chris-Pratt
                hasActors some Vin-Diesel
```

hasActors some Zoe-Saldana

hasDirector some James-Gunn

hasGenre some Action

hasGenre some Adventure

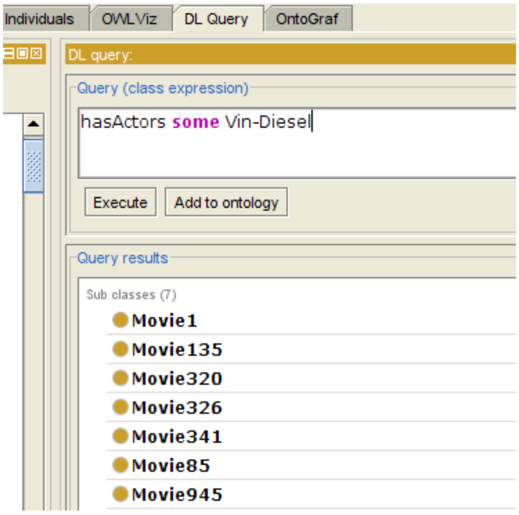
hasGenre some Sci-Fi

hasTitle value "Guardians of the Galaxy"^^string

hasYear some 2014

Movies

These constraints hold all information that is relevant to the movie. For instance, the title or name of the movie, who performed as actors in the movie, who directed it, what is its genre and what is released the year of the movie. In case, if someone is interested to see in how many movies Vin-Diesel has performed as an actor then following the DL query can retrieve information (see Figure 46).



hasActors some Vin-Diesel

Sub classes (7)

- Movie1
- Movie135
- Movie320
- Movie326
- Movie341
- Movie85
- Movie945

Figure 46 An example of a DL query and its results in protégé editor

6.5.6 Enron knowledge-base

The Enron knowledge-base hold the information about the email, sender, receiver, date and time of emails and attachments. Its main classes (e.g. Email, Sender, Receiver, Attachment) are extending the Core ontology classes. The Email_DHIN_ONTO is utilised the Date_Time class from the Core ontology and it is not extended (see Figure 47).

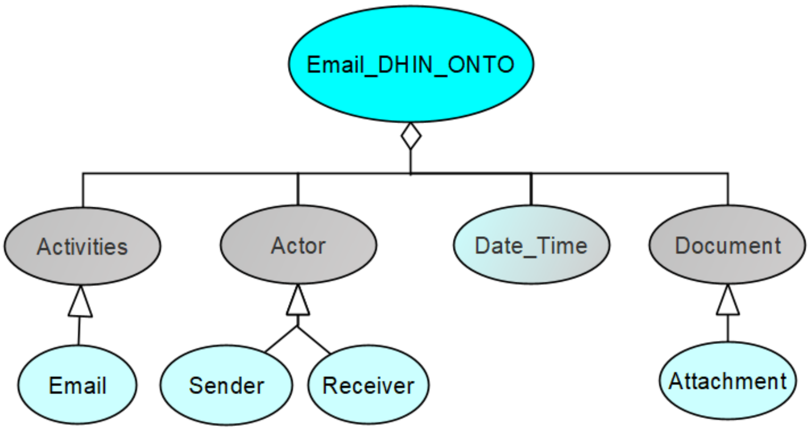


Figure 47 A glimpse of Enron knowledge-base

The constraints are defined on each concept or class which is the part of the DHIN network For example, the Email1 is the subclass of the Email class and this class hold the following information

Class: Movie1

SubClassOf:

hasDT value 2001-05-14 16:39:00-07:00

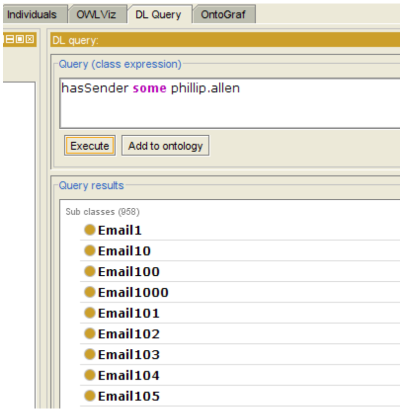
hasReceiver some tim.belden

hasSender some phillip.allen

hasAttachment value true

Email

An email class hold the information about the sender, receiver, date and time of the email and if there are some attachments. In case, if someone is interested to see how many emails philip.allen has sent to different people the following DL query can retrieve information (see Figure 48).



hasSender some phillip.allen

Sub classes (958)

- Email1
- Email10
- Email100
- Email1000
- Email101
- Email102
- Email103
- Email104
- Email105

Figure 48 An example of a DL query and its results in protégé editor

6.6 Temporal knowledge graphs snapshots

Once the knowledge base has been created we can also extract temporal knowledge graphs for different timestamps to use them for the change detection layer. For extracting the reduced node and community graphs for any time interval we will specify the timestamps to extract the knowledge graphs from the temporal knowledge base. For instance, Figure 49 shows the knowledge graphs for only DBLP Data Mining papers based on the timestamps of 1995 1996 and 1997 respectively. These temporal knowledge graphs contain paper, authors, venue, publisher and type of only data mining domain papers for three consecutive years. Each paper has been given a unique pattern number and we can also extract the information related to the papers from these knowledge graphs. As the Ontograph in 1995 shows that paper 5001944 has been published in the venue KDD conference and there are two authors (A-J-F-le-loux and J-W-van’t-zand) of this data mining related paper. Further information can also be extracted from the other knowledge graphs of timestamp 1996 and 1997 respectively. Moreover, these temporal knowledge graphs will be used to extract node and community-based graph patterns which will be the building block of the implementation of the next layer of our proposed system.

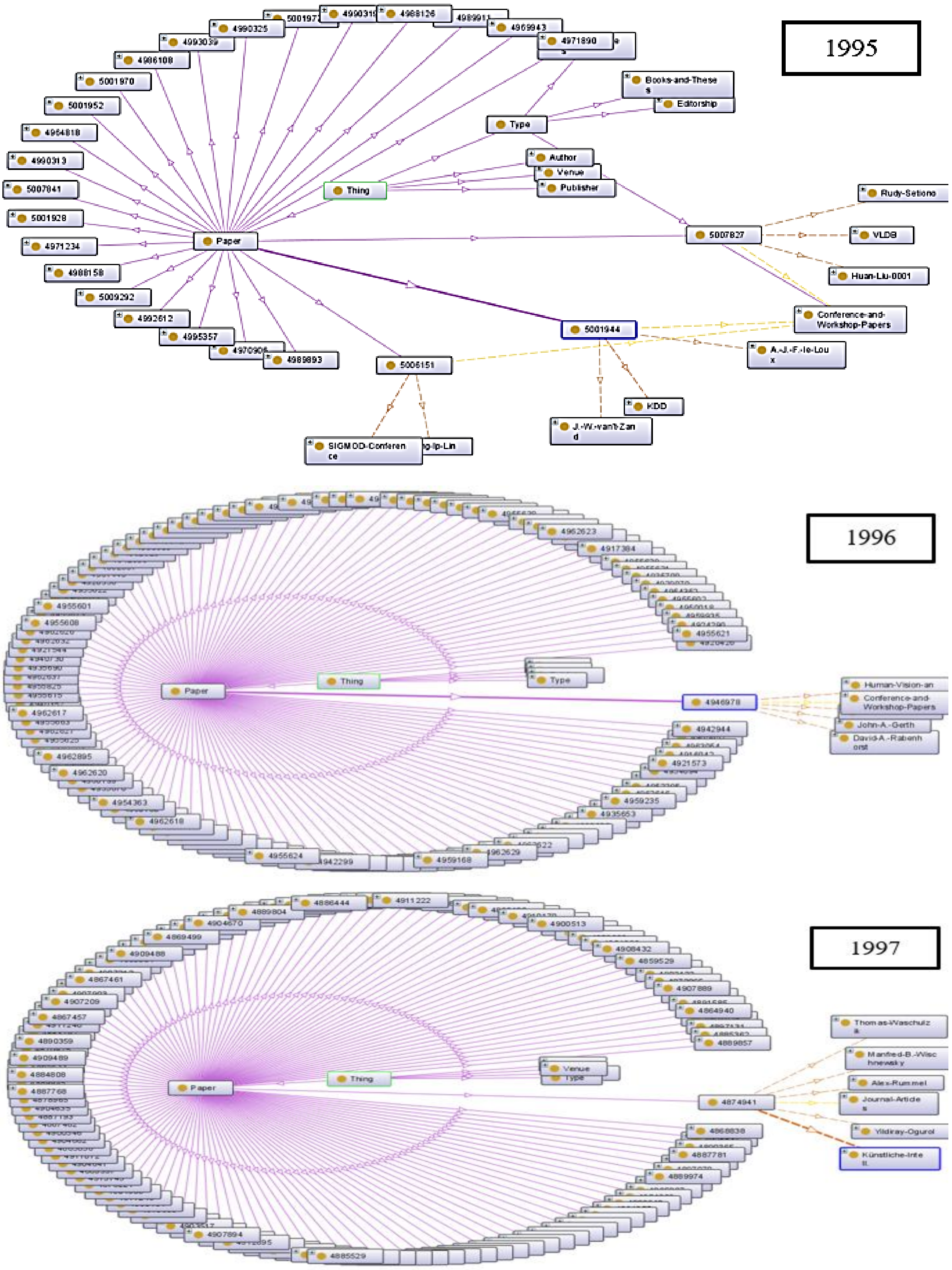
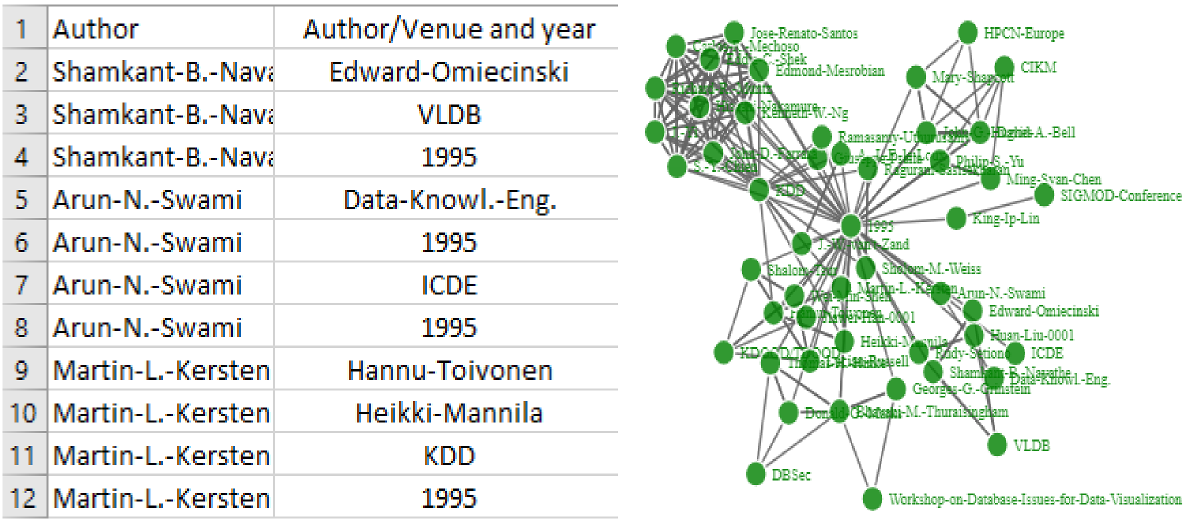


Figure 49 Temporal knowledge graphs of three consecutive years

6.6.1 Node or community relational graphs extraction

In the last component of the implementation, the system has developed temporal knowledge graphs and the outcomes are illustrated in Figure 49. which clearly show how each DBLP data point has been converted into the proper format. However, to achieve the desired outcome of this research aim, it is essential to convert the temporal knowledge graphs in the node and community relational graphs using Algorithm 4 (see detail in sections 5.5.1 and 5.5.2). This graph data is saved as a CSV file to be used in the next phase of the change detection. Figure 50 shows the CSV data file generated which can be mapped to a graph to see the details of each node of the graphs. It is shown in figure 50 Shamkant has published a paper with Edward at the VLDB conference in 1995. Moreover, the graph on the right side shows of the figure 50 shows the connection between each node of the dynamic heterogeneous network. Furthermore, these CSV data files will be used for further data mining steps for the discovery of the changes at the community, global and local or node level.

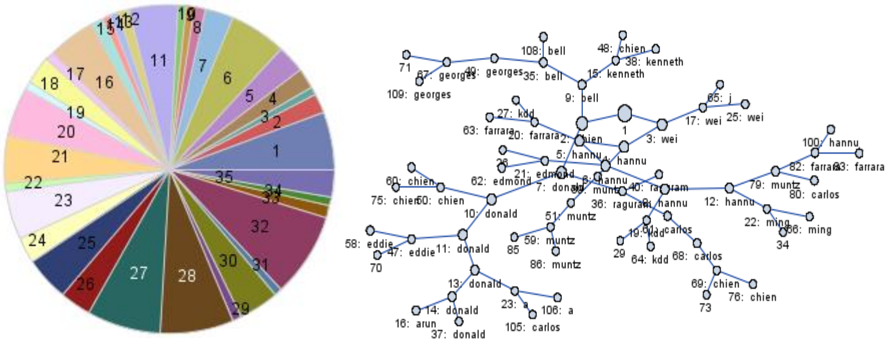


1	Author	Author/Venue and year
2	Shamkant-B.-Nava	Edward-Omiecinski
3	Shamkant-B.-Nava	VLDB
4	Shamkant-B.-Nava	1995
5	Arun-N.-Swami	Data-Knowl.-Eng.
6	Arun-N.-Swami	1995
7	Arun-N.-Swami	ICDE
8	Arun-N.-Swami	1995
9	Martin-L.-Kersten	Hannu-Toivonen
10	Martin-L.-Kersten	Heikki-Mannila
11	Martin-L.-Kersten	KDD
12	Martin-L.-Kersten	1995

Figure 50 Temporal knowledge base to node or community CSV graphs

6.6.2 Clustering implementation to discover community-level changes

Community-level changes are discovered by implementing SAS code for text cluster nodes developed in SAS enterprise miner. This text node cluster implements hierarchical and expectation-maximization clustering algorithms (see details Chapter 5 Section 5.6). Both of these methods used in SAS rely on singular value decomposition to transform the CSV document into dense but low dimension grouped sets. For instance, the extracted node or community-based CSV data example discussed in Section 6.6.1 is used to demonstrate how the clusters are developed for community-level change discovery. In this example, we can see that at this temporal snapshot the 35 clusters of the authors-paper-venue base cluster has been developed which can be seen in the pie chart shown in Figure 51. Moreover, we can also see hierarchical clustering which gives the hierarchical level importance of a node in the node community set and the text-based clustering of the item sets which also give important information to the user to understand the communities developed at this timestamp. The same steps have been implemented on each time-stamped dataset and created the number of clusters, hierarchical clusters and textual clusters to find the community-level changes between these periods. We can see some textual clusters like cluster 1 in Figure 51 has an important community "edward-omiecinski huan-liu- navathe rudy-setiono shamkant-b vldb" where Edward, omiecinski, huan-liu, navathe, rudy-setiono and shamkant are the co-authors in the data mining field in 1995 and they have published their work in the research venue VLDB conference.



1 edward-omiecinski huan-liu- navathe rudy-setiono shamkant-b vldb

4 bhavani-m thuraisingham mark donald-g hinke thomas-h

Figure 51 An illustration of the cluster development to extract community-level changes

6.6.3 Association rule mining for rule level change discovery

In this phase of the implementation, we have used SAS programming to discover the relationships among the sets of nodes in the node and community-based data extracted from temporal knowledge graphs. We have used MBANALYSIS SAS programming procedure to extract these relations in the data. This built-in procedure of the SAS programming uses the

FP-growth algorithm (Han, Pei, & Yin, 2000) to find the relational frequent pattern in the data. The rule mining is used to discover the required rules based on two parameters support and confidence.

- To find the set of rules for the items which happen with a frequency that is greater than a specification threshold will be specified in the support parameter.
- To find the rule set that is based on the confidence higher than a certain threshold confidence parameter is used in the implementation.

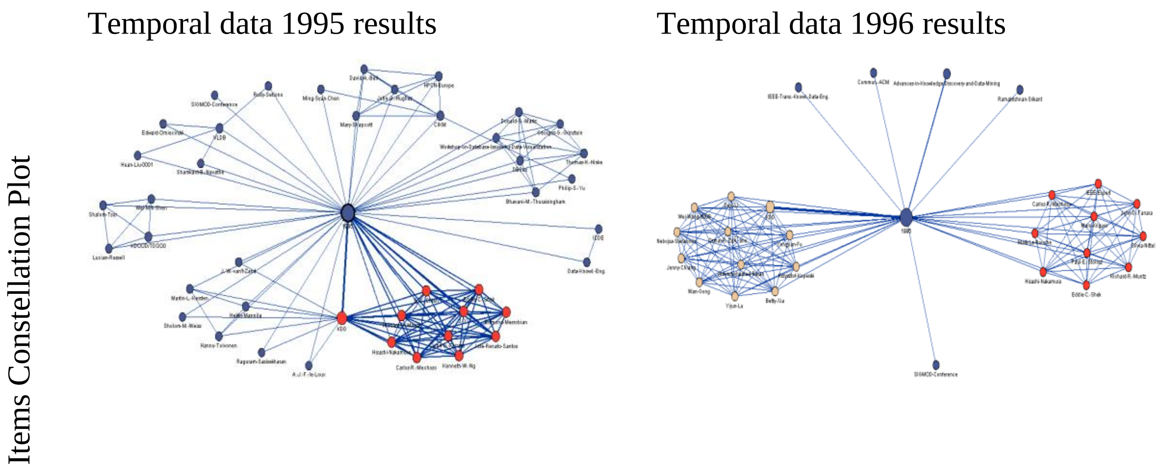
We have applied the SAS implemented the programme on the CSV DBLP data set for 1995 data mining papers and we have extracted around 20000 important rules. Table 6.2 gives a glimpse of the extracted rules. We can see that these rules can be used for the extraction of important information. For example, S.Y.Chien has published a paper with co-author Richard at the KDD conference in 1995. The same SAS programming implementation is applied on the other temporal node and community relational graph data to extract rule-based changes in these datasets.

Support(%)	Confidence(%)	Lift	Rule
23.529412	88.88888889	3.358025	S.-Y.-CHIEN ==> KDD & 1995 & RICHARD-R.-MUNTZ
23.529412	88.88888889	3.358025	HISASHI-NAKAMURA ==> KDD & 1995 & RICHARD-R.-MUNTZ
23.529412	88.88888889	3.358025	RICHARD-R.-MUNTZ ==> KDD & 1995 & HISASHI-NAKAMURA
23.529412	88.88888889	3.358025	RICHARD-R.-MUNTZ ==> KDD & 1995 & EDMOND-MESROBIAN
23.529412	88.88888889	3.358025	S.-Y.-CHIEN ==> KDD & 1995 & EDDIE-C.-SHEK
23.529412	88.88888889	3.358025	EDDIE-C.-SHEK ==> KDD & 1995 & CARLOS-R.-MECHOSO
23.529412	88.88888889	3.358025	CARLOS-R.-MECHOSO ==> KDD & 1995 & EDDIE-C.-SHEK
23.529412	88.88888889	3.358025	J.-YI ==> KDD & 1995 & HISASHI-NAKAMURA
23.529412	88.88888889	3.358025	EDMOND-MESROBIAN ==> KDD & 1995 & RICHARD-R.-MUNTZ
23.529412	88.88888889	3.358025	HISASHI-NAKAMURA ==> KDD & 1995 & J.-YI
23.529412	88.88888889	3.358025	EDMOND-MESROBIAN ==> KDD & 1995 & S.-Y.-CHIEN

Table 6.2 Association rules extracted from DBLP data for data mining papers for 1995

6.6.4 Link analysis to find global changes

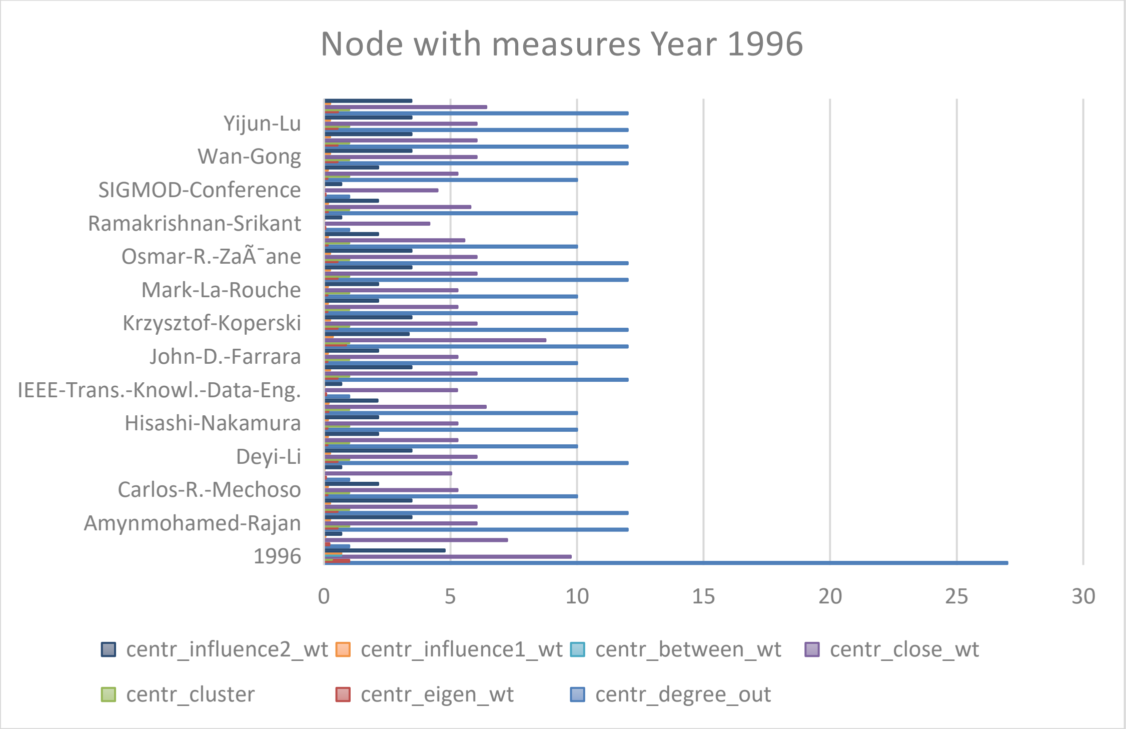
In this phase of the development, this research has employed the built-in link analysis[17] node of SAS enterprise miner to discover and examine the global relational changes between node items extracted from the temporal knowledge graphs. The implementation of the link analysis node offers different centrality measures to discover changes from the graphs. This node also helps to find the interlinked clustered items between graph objects. The link analysis node implementation provides clustering coefficient, influence centrality, closeness centrality, betweenness centrality, eigenvector centrality using the Jacobi-Davidson algorithm(Hochstenbach & Notay, 2006), measures to detect global level changes in the graph systems. For example, figure 52 gives the visual change description of two temporal snapshots of 1995 and 1996 from data mining papers of the DBLP dataset. The items constellation plot shown in Figure 53 also portrays the important nodes in each year which can be used as a baseline for change analysis. For example, in the 1995 plot, there are three dense connected components as compared to the 1996 plot which has only two dense networks nodes. Figure 53 also have shown the change measure of each node in the graphs data. This figure shows that KDD is a conference of interest for most of the authors in 1995 in contrast to 1996 where authors were interested to publish their papers at the IEEE conference. Moreover, various other changes can also be discovered for instance new authors, new conferences and venues between two timestamps of data.


Temporal data 1995 results

Temporal data 1996 results

[17] https://documentation.sas.com/doc/en/emref/14.3/n194wxm6x5sxb4n18j57p9kxc1vv.htm#p14hs6ji86ojlrn185al3mgb8x0h

Figure 52 link-based graph change representation in 1995 and 1996 DBLP data mining paper with authors and venues



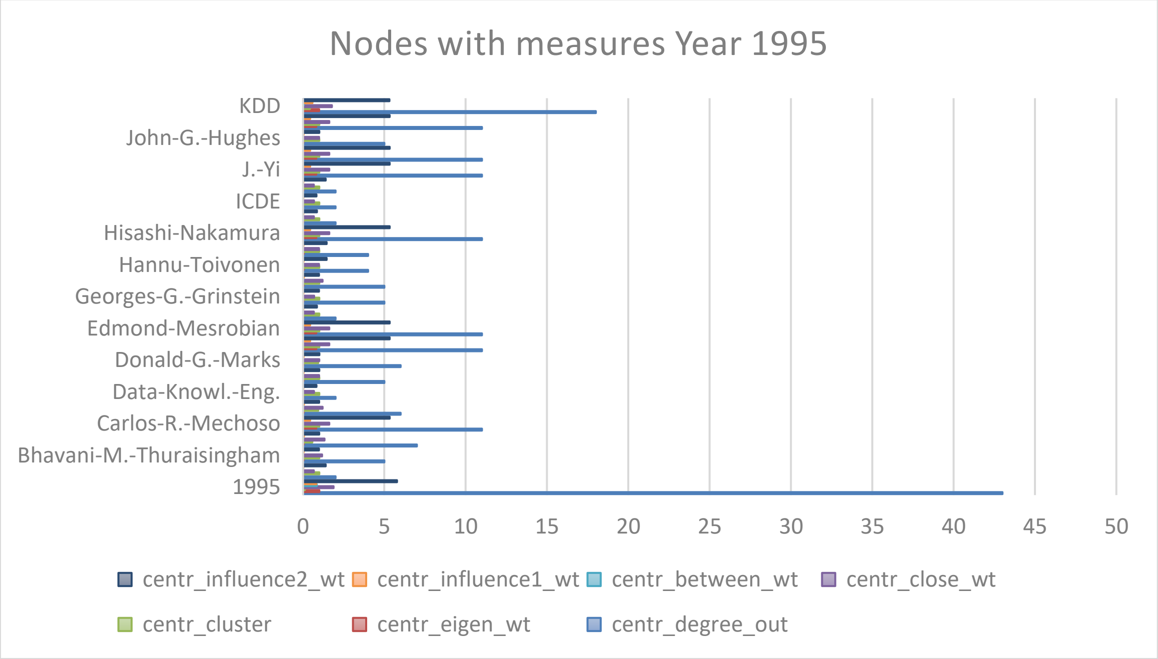
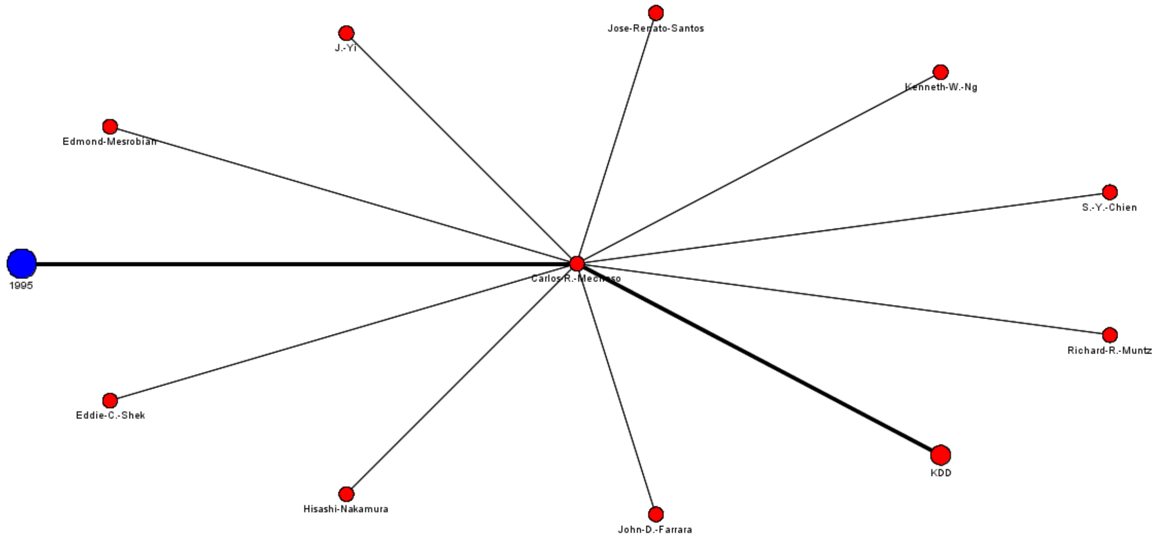


Figure 53 Illustration of global changes measures clustering coefficient, influence centrality, closeness centrality, betweenness centrality, eigenvector centrality for nodes of the years 1995 and 1996

6.6.5 Local change patterns discovery implementation

This is the final developed component of the implemented framework. The three algorithms (Algorithms 5-7 see details in 5.7.4, 5.7.5 and 5.7.6) has been developed using Java programming to find the node level or community level changes between two temporal graphs. Figure 54 gives the insight of the implementation results which shows there are changes in the two nodes it is mentioned in the created final file using the similarity calculation algorithm (Algorithm 5 see details in chapter 5) and stable/change pattern extraction algorithms (Algorithm 6 and 7). In Figure 54 change can be visualised that in 1995 one author Carlos R Mechoso has published data mining papers in collaboration with Jose-Renato-Santos, Kenneth-W.-Ng, Eddie-C.-Shek, Richard-R.-Muntz,S.-Y.-Chien, J.-Yi, Hisashi-Nakamura, Edmond-Mesrobian and John-D.-Farrara in KDD conference on the other hand the same author has published a data mining research paper with authors Mark-La-Rouche, Hisashi-Nakamura, John-D.-Farrara, Silvia-Nittel, Marc-Kriguer, Paul-E.-Stolorz, Eddie-C.-Shek and Richard-R.-Muntz in IEEE-Expert conference in the year 1996. Consequently, it makes a changing pattern as the similarity between these two node graphs is 0.31 which is less than 95% of the specified stable pattern threshold. So for example, if we have given a 95% minimum similarity threshold for stable pattern the CSV file generated will show us the implementation result as (Jose-Renato-Santos, Kenneth-W.-Ng, S.-Y.-Chien, J.-Yi, Edmond-Mesrobian, KDD, Mark-La-Rouche, Silvia-Nittel, Marc-Kriguer, Paul-E.-Stolorz, 0.31, Changed) using the local changed pattern discovery Algorithm 7 implementation (see detail in chapter 5 Section 5.7).



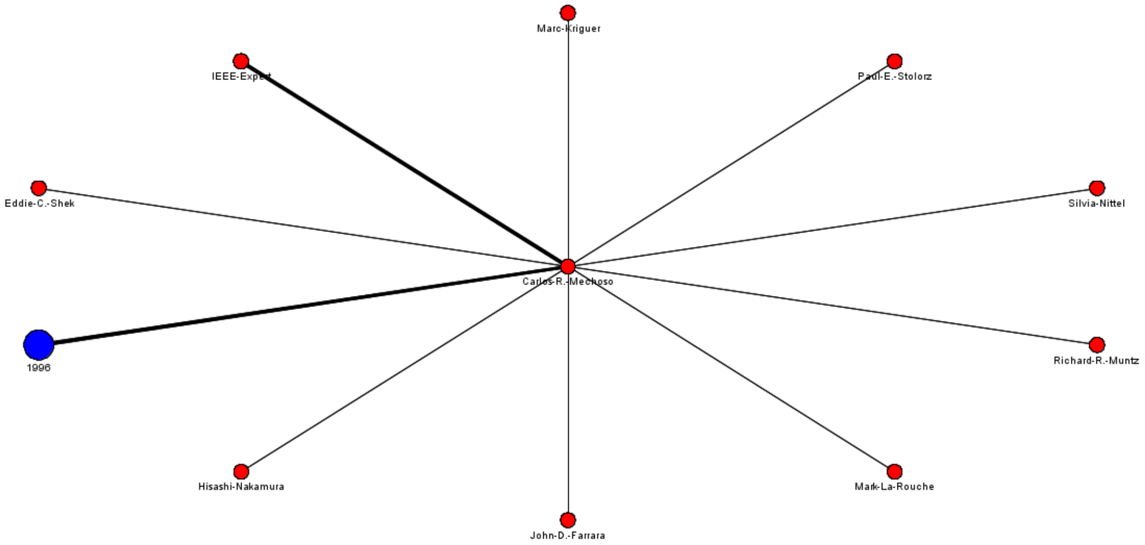


Figure 54 The node or local changes of Carlos R Mechoso author between 1995 and 1996

6.7 Summary

In this chapter, the proposed system ChaMining was implemented the development results of each steps has also been elaborated. Three algorithms described in Chapter 5 has been implemented to map structured, semi-structured and unstructured datasets to temporal knowledge base. The core ontology was also developed and extended for three datasets. Once the knowledge graphs have been developed the chapter has shown the implementation of change mining components of the proposed framework. Moreover each component's development has been elaborated by using DBLP dataset to test results.

The next chapter will present the results of the proposed system to detect community node and global level changes. Next chapter will also give the outcomes of empirical evaluation of developed method with existing methods for change discovery.

Chapter 7 Results and Empirical Evaluations

7.1 Introduction

Thus far this study has described the design and implementation of the ChaMining change discovery framework. Chapter 5 have given the system design methodology and Chapter 6 has explored the implementation of each component of the proposed system and how it works to discover changes using domain-specific temporal knowledge base and data mining methods. Therefore, the next step is to find the results and verify the effectiveness of the proposed change mining system. This research will be empirically evaluated with five approaches introduced in the related work section of chapter 3 which are characterized as a baseline for the change detection of dynamic heterogeneous information networks. In this chapter, we have performed experiments on four temporal datasets discussed in chapter 6, IMDB, DBLP, Enron Email and High Energy Physics Citation. We have already developed a knowledge base in chapter 6 now we will use the data set that is already formatted to use in the data mining algorithms. The data-mining task makes it easier to extract relational patterns out of relational data and we can use these relations to understand the changed behaviour of entities in different time windows. The experiments have been applied on four datasets to understand how the local, community and global level changes are discovered during different time windows. The rest of this chapter is organised as follows:

- Section 7.2 elaborates experiments setup
- Section 7.3 gives the details of evaluation Measures that are used in this research
- Section 7.4 explains why results and discussion are necessary
- Section 7.5 presents the results of community level change discovery
- Section 7.6 presents the results of global level change discovery
- Section 7.7 represents the results of local level change discovery
- Section 7.8 gives Empirical Evaluation with existing change discover system
- Section 7.9 Information loss calculation
- Section 7.10 ChaMining system with Other existing algorithms and frameworks
- Section 7.11 represents summary of the chapter

7.2 Empirical experimental setup

To effectively test the performance of the proposed framework approach for change discovery different data sets are utilised and the empirical evaluation was performed on a Windows operating system with 16 gigabytes of primary memory with an intel core i5 central processing system. This research has been evaluated with the existing approached in the same experimental environment to give the level playing field for each method.

7.3 Evaluation Measures

There are various evaluation measures to evaluate the performance of a change detection or information retrieval system with other related algorithms/frameworks. Evaluation methods used in the previous related researches are quantitative, qualitative, precision, recall and F-Score to evaluate the algorithms/frameworks of change mining or information retrieval system. In this research, we will consider all these measures to evaluate our results with the related algorithms.

7.3.1 Quantitative method

The quantitative evaluation measure tests the influence of the input parameter in discovering the change patterns. The quantitative method uses statistical measures, for example, respective minimum support and threshold to find the changing pattern computed in the temporal periods. Using this method quantitative community and global level temporal changes will be investigated, for example, the total number of clusters, item constellation, nodes out-degree centrality, eigenvector centrality, clustering coefficient centrality, closeness centrality, weighted betweenness centrality and influence centrality will be computed to find the global or community level changes between temporal graph between different timestamps. This research will also be using this measure in the rule mining method to find several changes in association rules in every timestamp based on the effect of changing the minimum support threshold.

7.3.2 Qualitative evaluation

In this research, the qualitative evaluation method will be used to find the clustered description of changed terms between two temporal DHINs. This approach will give humans understandable change community patterns and will show the evolution of the frequency of the connected terms in each community.

7.3.3 Precision recall and F-Score

The precision-recall and F-Score (Buckland & Gey, 1994) are widely employed in the empirical studies of information retrieval and artificial intelligence systems to measure their effectiveness.

7.3.3.1 Precision

It is a classification method that measures the ratio of retrieving relevant patterns divided by the total number of the patterns. In this study, precision as shown in equation (7.1) will be the ratio of the relevant change patterns discovered divided by the relevant plus irrelevant patterns discovered by the system.

$$\boldsymbol{P} = \frac{\boldsymbol{relevant\ change\ patterns\ discovered}}{\boldsymbol{relevant\ change\ patterns + irrelevant\ patterns\ discovered}} \quad \textbf{(7.1)}$$

7.3.3.2 Recall

The recall is a classification ratio that measures the probability of discovering relevant change patterns divided by the total number of change patterns that are expected to be discovered.

$$\textbf{R}= \frac{\boldsymbol{relevant\ change\ patterns\ discovered}}{\boldsymbol{total\ number\ of\ change\ patterns}} \quad \textbf{(7.2)}$$

7.3.3.3 F-Score

F-score gave in the equation (7.3) calculates the harmonic mean of precision and recall to evaluate the accuracy of the classification model.

$$\textbf{F}=\frac{\boldsymbol{2 * precision * recall}}{\boldsymbol{precision + recall}} \quad \textbf{(7.3)}$$

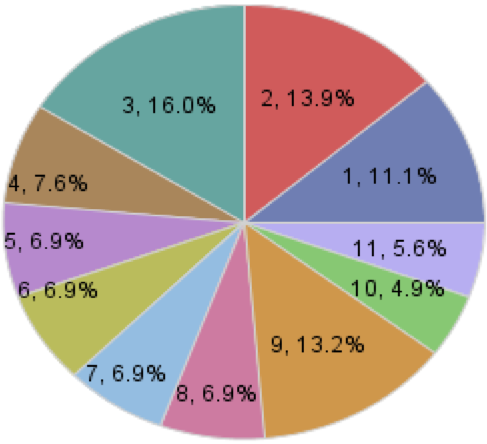
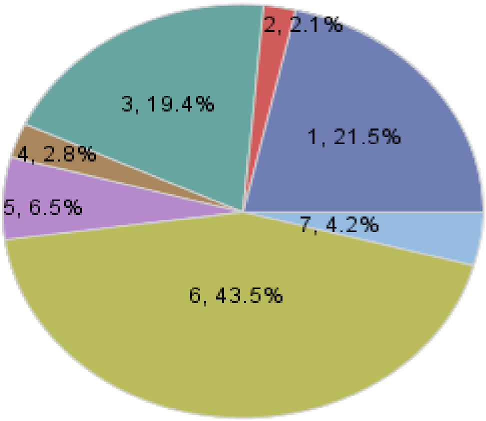
7.4 Results and discussions

The goal of this research is to find changes at different levels in a dynamic heterogeneous information network. Considering the concept of this research the four datasets were used and to reduce the complexity and length of the results two datasets IMDB and DBLP were employed to detect community and local level changes while Enron email and High energy physics datasets were used to detect global level changes. The datasets have already been prepared in temporal timestamps and we will use these temporal snapshots of data to gather and discuss the changes between consecutive periods. The results discussions will be based on the results generated by the experiments conducted in the developed system. The system has been employed on four domain-specific knowledge graph data sets. It is significant to state that the system is effective on others domains of discourse as well but data need to be prepared into the specific data format using data pre-processing and the knowledge engineering components of the implemented system.

7.5 Community-level changes discovery

It is very important to show the number of community-level changes in the textual form as it is understandable by the end-user. IMDB the data is dynamic and we have used five different snapshots of data which are T1 (Movies data from 1894 to 1955), T2(Movies data from 1956 to 1980), T3(Movies data from 1981 to 1996), T4 (Movies data from 1996 to 2003) and T5 (Movies data from 2004 to 2019) for analysis. We have found a different number of clusters 11,7,6,9 and 30 at each timestamp T1, T2, T3, T4 and T5 respectively. Table 7.1 shows the community-level changes in the IMDB data based on the target variable as Actors. In this changes discovery, we have only considered seeing changes in the Actors of the movies in the different temporal snapshots of IMDB database. We can also specify the target data to any other variable such as titles, genre, director, writer, and description to analyse changes based on these variables. Since this dataset contains movies information starting from 1894 to 2019 so the clusters generated does not provide any sense because most of the actors could have left the movies industry in different time windows so we can not compare the results generated with the original data set. The main purpose of this analysis was to show this framework works on large datasets as well. Table 7.1 shows that Movies data from 1894 to 1955 have 11 clusters, Movies data from 1956 to 1980 have 7, Movies data from 1981 to 1996 have 6, Movies data from 1996 to 2003 have 9 and Movies data from 2004 to 2019 have 30 clusters respectively.

Each cluster generated in the table provides group of important movies actors who have worked together in each time stamp T1,T2,T3,T4 and T5. For example cluster 1 contains community (group) of actors who have worked together in this timestamp and their names are “helen talmadge herbert norma thomas tully elliott french roberts theodore marshall florence charles edna and erich”. Comparatively, cluster 2 have another community of actors who have worked together and their names are “douglas fairbanks grasse george blake higby jewel lawrence pallette carmen tom fred wilson alice beranger”. So Table 7.1 shows that, number of clusters generated in each timestamp show community level changes in that timestamp.

Time	**Number of clusters Generated**	**Important movies actors in each cluster number**
Movies data from 1981 to 1996	 	1.helen talmadge herbert norma thomas tully elliott french roberts theodore marshall florence charles edna erich 2.douglas fairbanks grasse george blake higby jewel lawrence pallette carmen tom fred wilson alice beranger 3.clark lee carey arthur frederick harris frank moore howard raymond edward gertrude violet charles blanche 11.jenny larsson sjã strã tschernichin-larsson victor gã edith emil william nilsson john carl
Movies data from 1956 to 1980	 	**3.**de josã franco antonio maria mario marã luis alberto carlo ã fernando luigi aldo 4.vladimir aleksandr mikhail nikolay evgeniy boris viktor yuriy leonid georgiy natalya lyudmila oleg ivan sergey 6.john robert richard michael james peter george david jack paul william charles lee frank martin

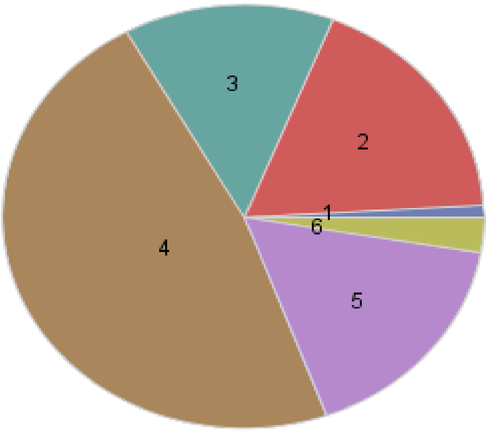
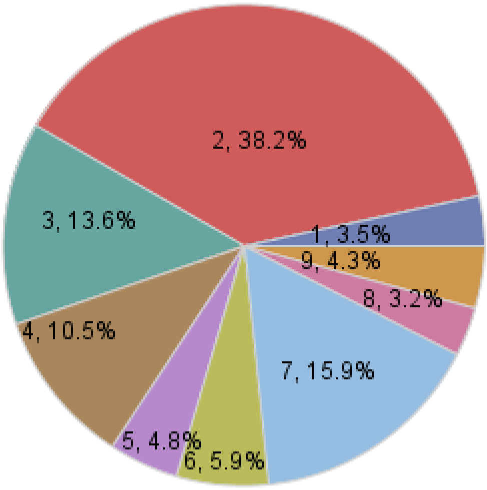
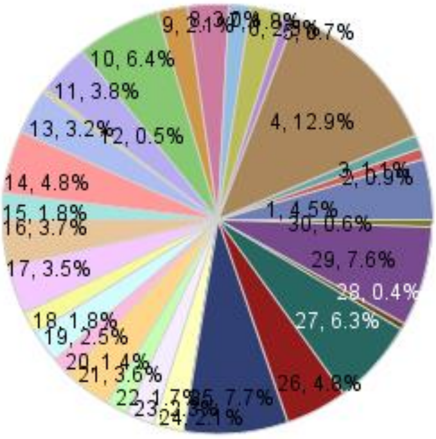
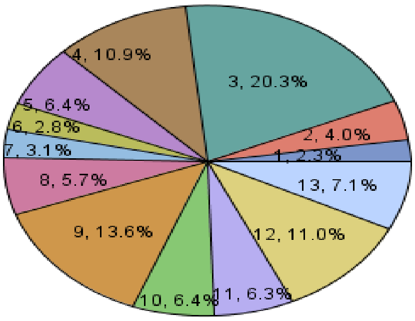
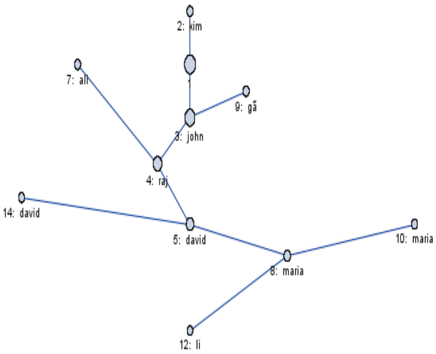
Movies data from 1981 to 1996	 	**1.**bata dragan predrag velimir zivojinovic todorovic bogdan diklic ljiljana miodrag jovanovic petar dragomir markovic miki 4.john michael david richard robert james peter paul tom william mark joe bill christopher George
Movies data from 1996 to 2003	 	**1.**giorgos kostas nikos dimitris giannis vasilis alekos thanasis spyros maria eleni labros andreas hristos mary 2.john michael david james +richard robert paul peter scott william tom mark kevin christopher lee 6.jean franã jacques pierre michel ois le gã philippe patrick lã andrã bernard daniel rard
Movies data from 2004 to 2019	 	16.david john chris tom ryan matthew matt kevin robert sarah jason christopher paul brian jones 24.kim lee +park choi jeong jo jung jang han yoo kang shin jeon song seo 30.singh sharma gill gurpreet kaur dhillon anmol karamjit rana bhalla bajwa binnu grewal rishi b.n.

Table 7.1 Clusters showing community level changes of IMDB data 1894 to 2019

To effectively analyse the community-level changes in more detail we divided the data into four-time windows five years snapshots of movies data starting from 2000 and to 2020. We have used all the variable actors, country, director, genre, and writer to find the communities between them so we have found these results. The results show very interesting (See table 7.2) facts about the movies data for a temporal snapshot of data 2000 to 2005, 2006 to 2010, 2011 to 2015 and 2016 to 2020 13,9, 13 and 28 communities were discovered. Each community represented in the clusters shows a group of actors, directors and writers who have worked together in these years and we can see a very interesting fact that in most of the important communities john, David michael and james are together or we can also say that they work together in movies for around 15 years. We can also see that from 2016 to 2020 they are not captured in the community which shows the community-level change, so we can conclude they are not working together in this period. The cluster hierarchy plot also shows similar objects in hierarchical groups.

Time	Number of clusters Generated and cluster hierarchy	Important Communities generated
Movies data from 2000 to 2005	  2: kim 7: ali 9: gã 3: john 4: raj 14: david 5: david 10: maria 8: maria 12: li	3. john david michael james ryan richard scott paul chris mark jason brian jennifer steve brown

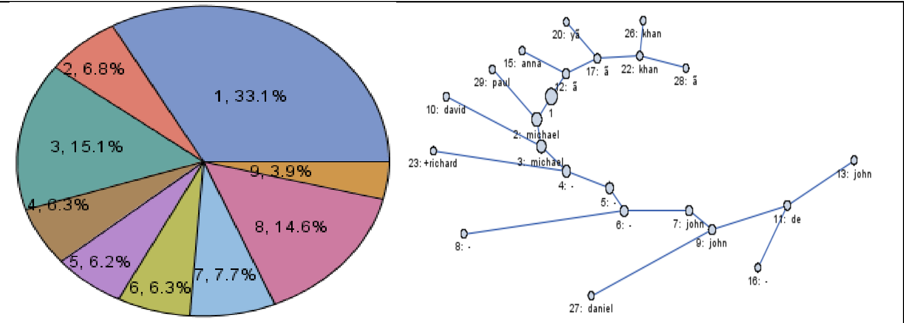
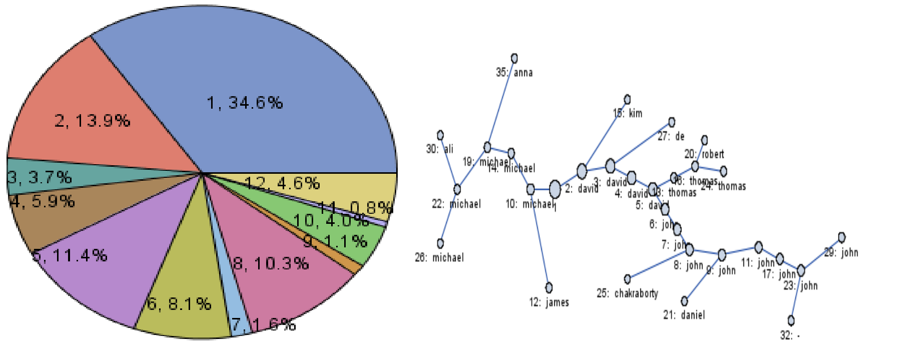
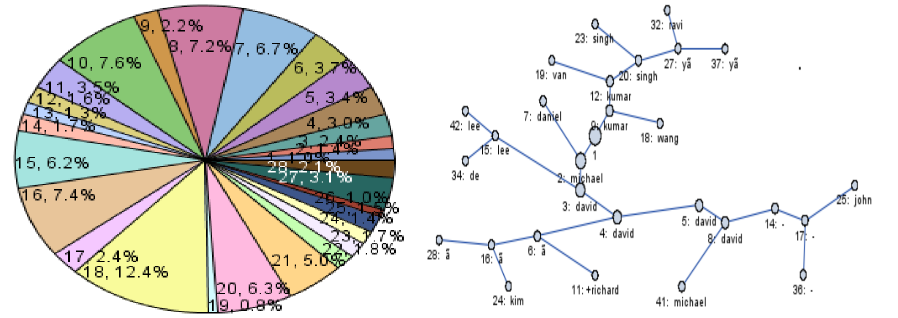
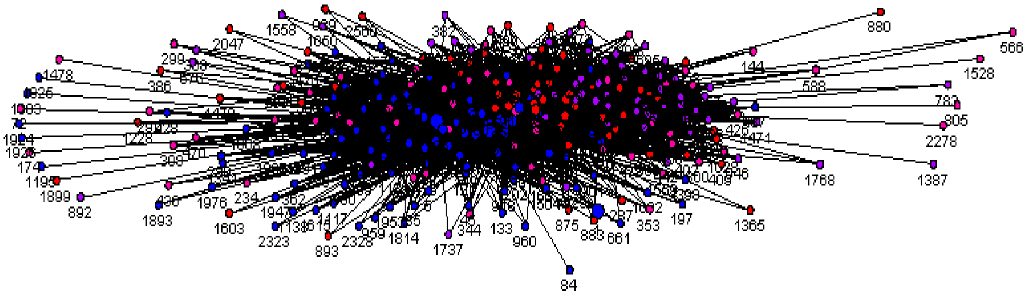
Movies data from 2006 to 2010		1. michael john david james chris robert richard paul scott jason kevin tom brian ryan christopher
Movies data from 2011 to 2015		michael john david james richard mark robert paul chris jason ryan scott tom christopher kevin
Movies data from 2016 to 2020		18. john chris jason richard robert paul tom mark scott ryan patrick christopher alex ben kevin

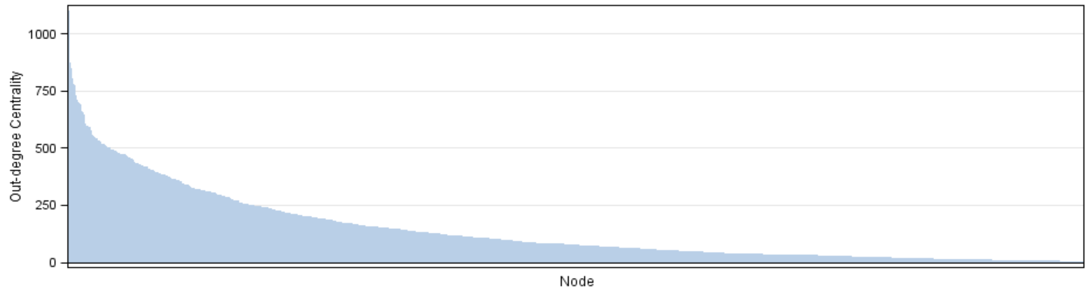
Table 7.2 Clusters showing community-level changes of IMDB data 2000 to 2020

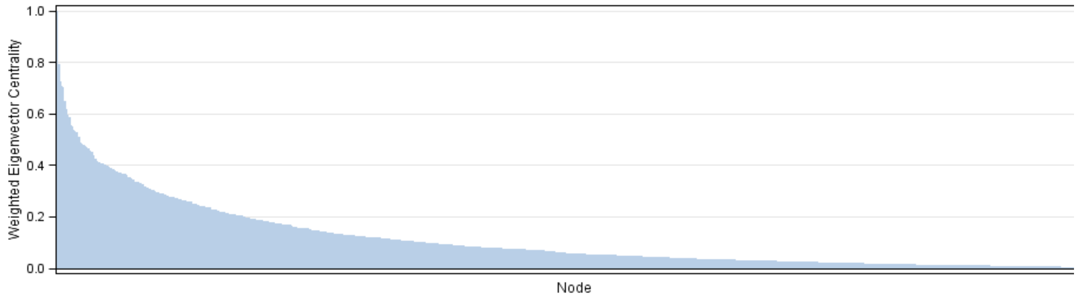
7.6 Global level change discovery

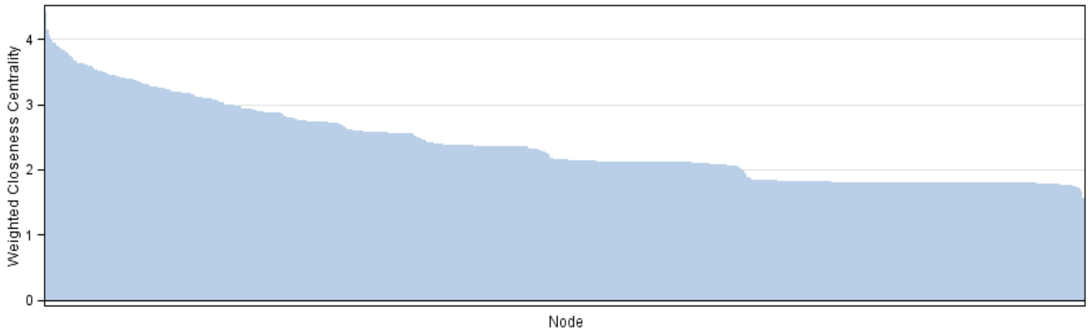
We have used Enron email personalised data for 4 different time stamps to analyse the global level changes that occurred between these periods. Each number in the data set shows an email sent to a receiver. The dataset has been divided into four snapshots to reduce the size of the output and show that the developed method work for the global change discovery of the nodes in the DHINs data. Table 7.3 shows the items constellation plot to show how dense or sparse the network entities are connected. We can see that the nodes of the Enron email network are densely connected at T1, T2 and T3 but sparsely connected at T4. The table also shows the comparative analysis of global change measuring parameters out-degree centrality, eigenvector centrality, clustering coefficient centrality, closeness centrality and betweenness centrality. The out-degree shows the number of outgoing edges, eigenvector centrality specifies the centrality of a node to be the fractional sum of the scores of all nodes that are linked to it, clustering coefficient centrality is the number of links between a node within its neighbourhood divided by the number of possible links, closeness centrality and betweenness centrality calculates the shortest path between the nodes. So by calculating these measures we have presented global level changes in a DHIN. For example, in table 7.2, out-degree centrality graphs shows changes in T1,T2,T3 and T4 temporal snapshots of Enron Email data which means the number of outgoing edges are varying with time, eigenvector centrality graphs in T1,T2,T3 and T4 which specifies the centrality of a node to be the fractional sum of the scores of all nodes are also showing changes in the nodes that are linked to each node in these temporal snapshots, clustering coefficient centrality graph presented for these four temporal snapshot also shows the varying nature of the nodes because the number of links between a node within its neighbourhood divided by the number of possible links shown in each centrality graph are different. Moreover, closeness centrality calculates graph shown also represents shortest path between the nodes is changing over time. So, in table 7.2 by calculating these graph measures we have presented global level changes in a DHIN the measures also helps us to understand the global level changes between different snapshots of the Enron email network.

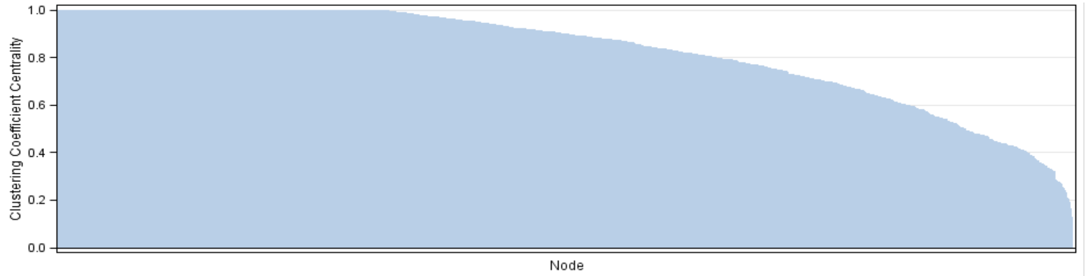
Temporal Email data T1



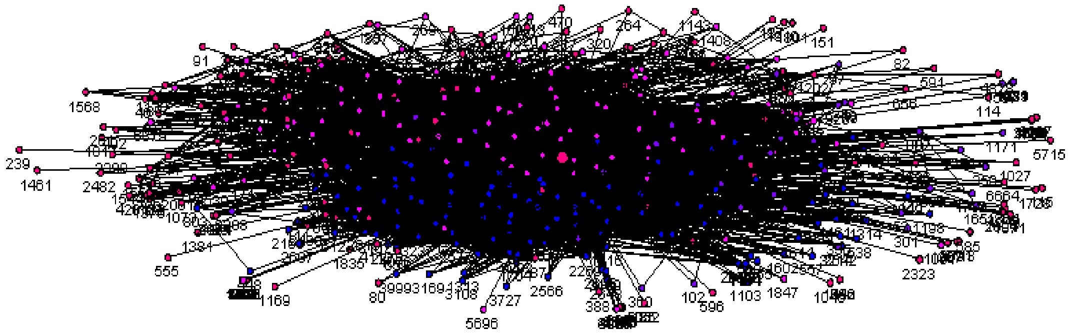


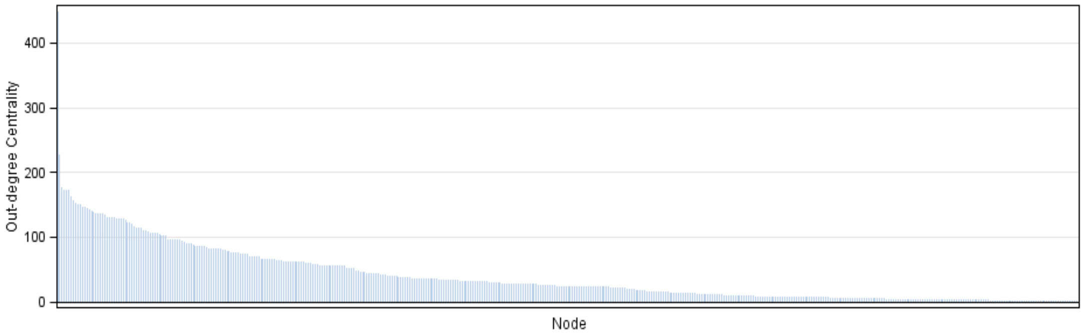


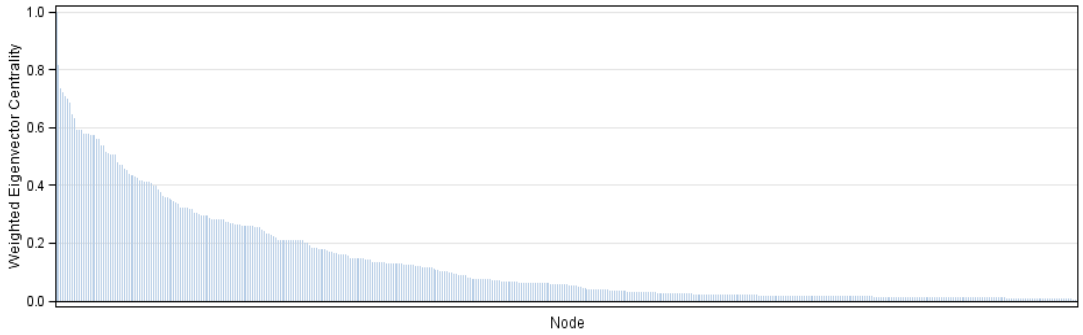


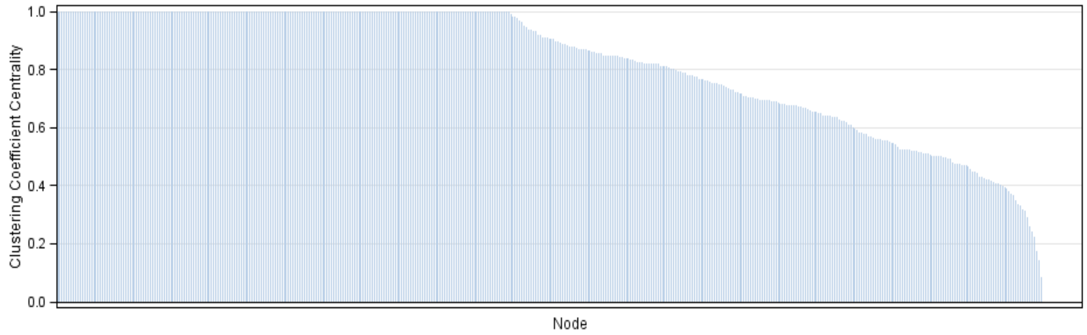


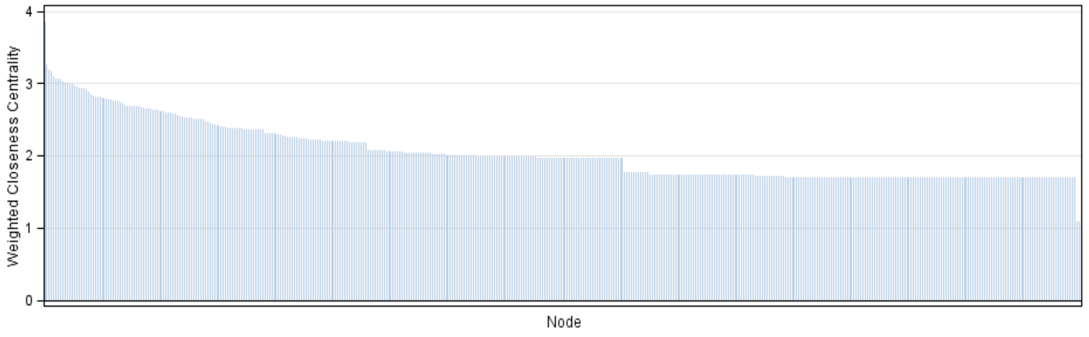
Temporal Email data T2



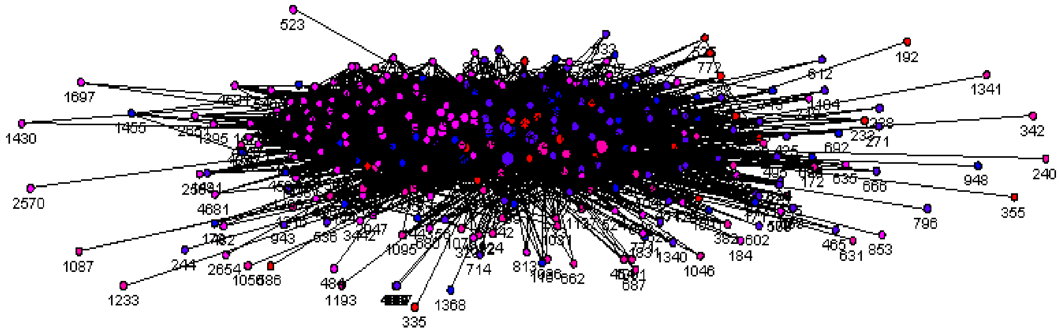


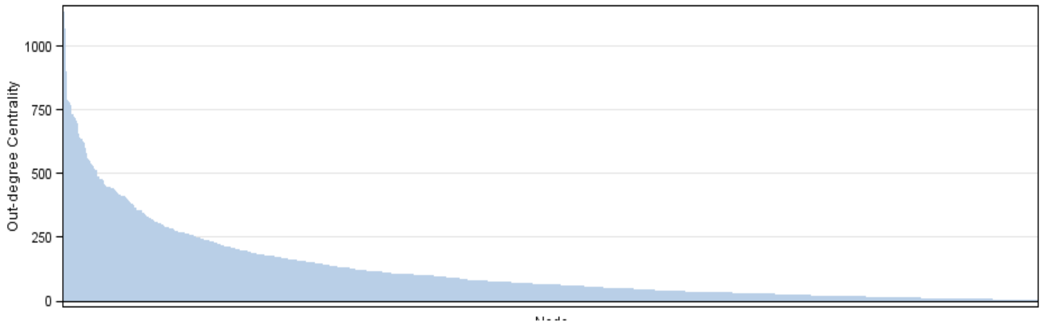


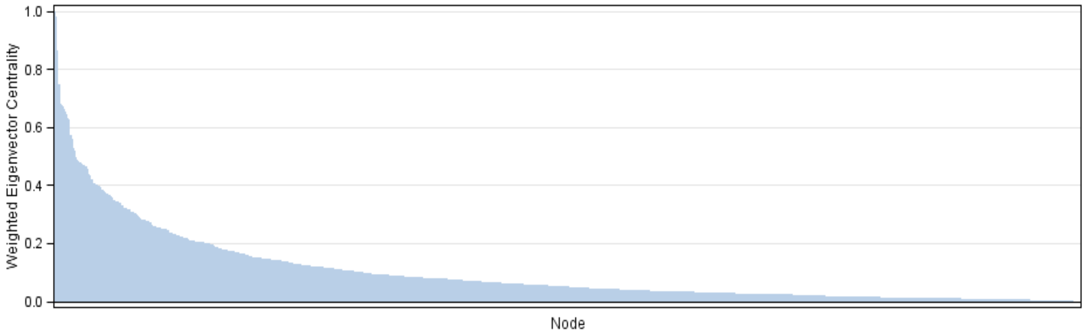


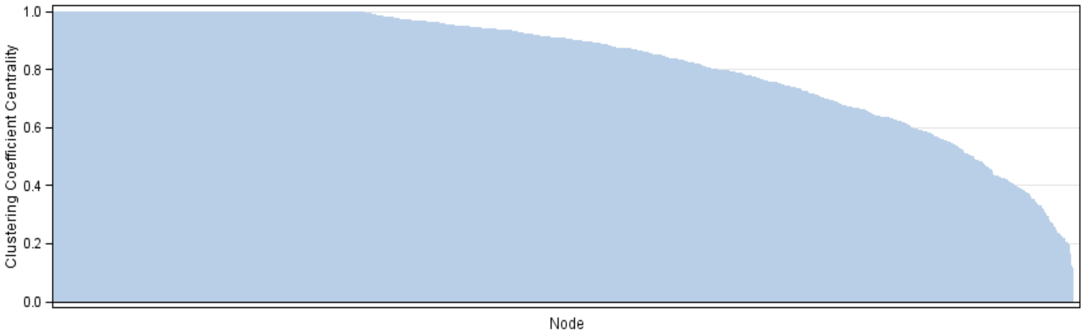


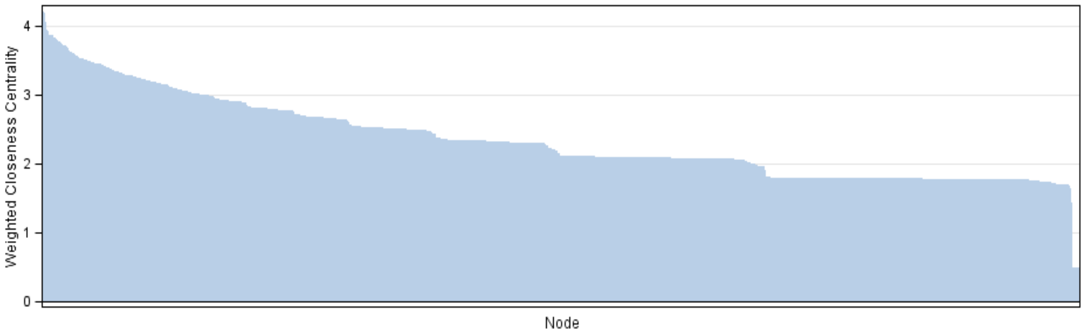
Temporal Email data T3











Temporal Email data T4

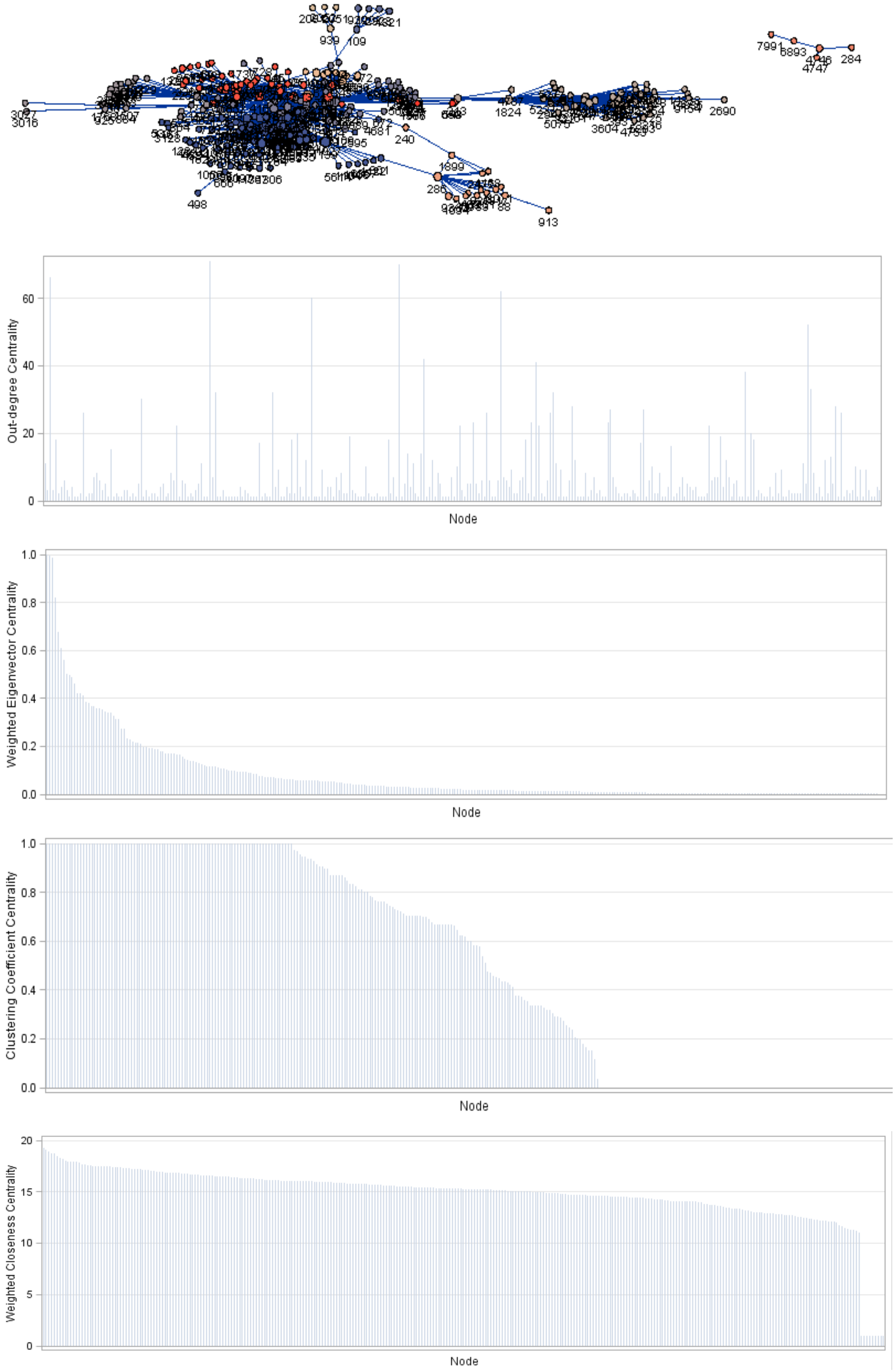


Table 7.3 Global change discovery from enron email personalised datasets

7.7 Local-level and rule level change discovery for high energy physics data

We have used high energy physics citation network data to extract local level changes. The data has been divided into 4 temporal snapshots and local level changes were discovered using the implemented change and stable patterns discovery algorithms. Moreover, important association rules based on the minimum support and confident threshold were also extracted from the results generated by applying association rule mining on the four snapshots of the high energy physics data set. Figure 55 depicts the association rules extracted at the different time intervals of the high energy physics dataset.

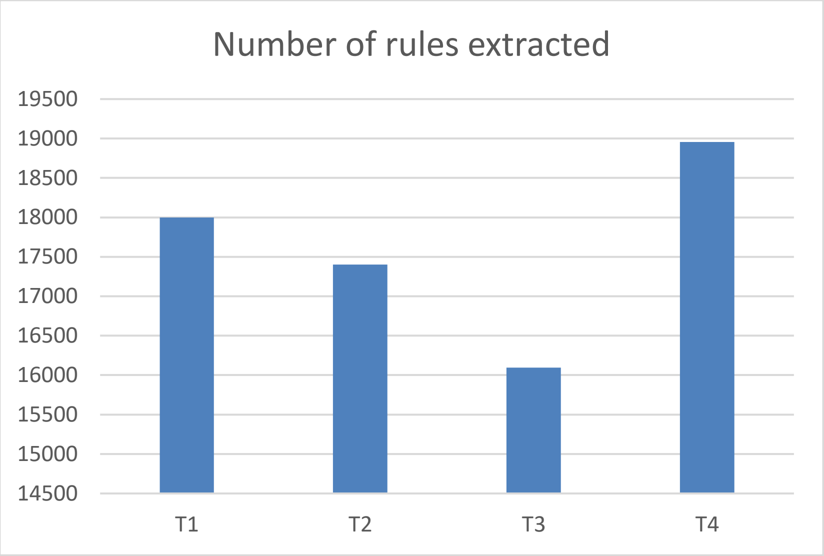


Figure 55 Number of rules extracted at different time intervals for high energy physics data

Percentage of change and stable patterns received in each case from the temporal high energy physics dataset are given below in Figure 56. It is obvious the there are a large number of change patterns in this data as the authors keep citing other works.

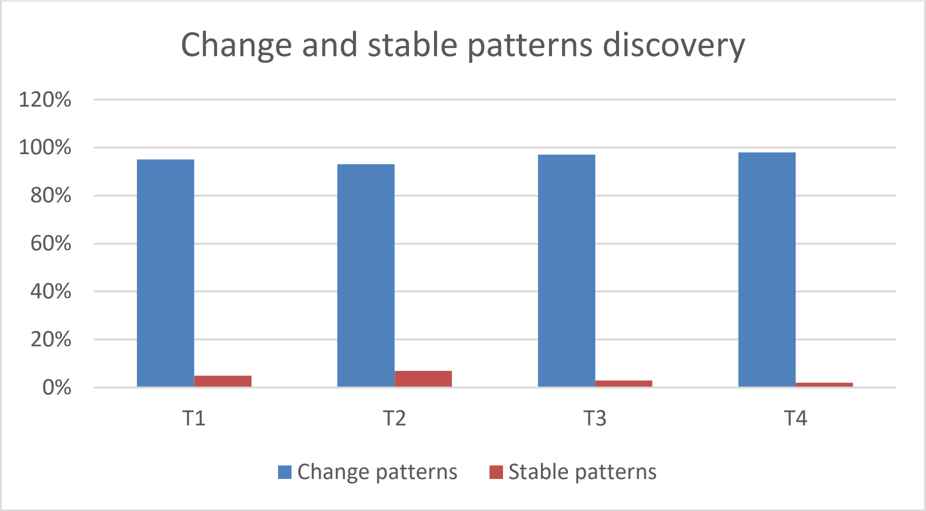


Figure 56 Change and stable patterns discovery

7.8 Empirical Evaluation with existing change discover system

To evaluate the effectiveness of the developed framework an empirical evaluation has been carried out on existing unsupervised change detection algorithms. All the unsupervised systems or algorithms CHRG, EdgeMonitoring, SizeCPD, DeltaCon, and CICPD perform change point detection instead of change discovery. Only Rule-base systems use relational mining to discover changes in dynamic heterogeneous information systems. So, an empirical evaluation was performed between our proposed method and the rule-based (Loglisci et al., 2015) system of the change discovery. The rule-based system has been implemented in Sictus-Prolog, so it needs the Sictus compiler for running. Moreover, the use is not so immediate, as the data should be formatted in Datalog formalism. The first difference between our approach and the rule-based method is that our system is not dependant on any other system for background knowledge development. The rule-based system uses SPADA inductive logic programming system to extract rules from the data that has been saved in datalogs. The second difference between our system and the rule-based system is that it is based on the research-based system and we have used java and SAS programming based systems for extracting rules from our datasets. The rule-based system also uses dissimilarity measures to extract stable and change patterns while our approach uses Jaccard similarity measures to extract stable and change patterns. Our approach works on all three types of data to extract change patterns while rule-based change discovery algorithm only works on the unstructured datalogs. The rule base system has used the concept of the target object which only select specific nodes in a dynamic heterogeneous information network, non-target object are neglected during change patterns discovery in a rule-based system. Our method in contrast give equal importance to each node and reduces information loss eventually our approach provides more patterns than the rule-based approach. Table 7.4 shows the comparative evaluation of our developed method with the relational mining (rule-based) method of change discovery. Chapter 3 related work also provides the shortcoming of rule-based approach.

Evaluation Matric	The characteristics of our developed system as compared to the rule-based algorithm	
	Rule-base	ChaMining
Quantitative number of change patterns discovered	√	√
Qualitative	X	√
Number of change rules	√	√
Clusters of changes	X	√
F1	X	√
Precision	X	√
Recall	X	√
Depend on another system to perform change discovery	√	X
Support human-understandable patterns	X	√
Change discovery from Unstructured datalogs data	√	√
Change discovery from structured data	X	√
Change discovery from semi-structured data	X	√

Table 7.4 Comparative evaluation of Rule-based changes discovery with our developed method

We will show now the comparison of the empirical results of the rule-based change discovery system with our developed system using the identical minimum support to extract change patterns from DBLP, IMDB, Enron email and high energy physics datasets. Since we have used a formatted version of the high energy physics citation[18] benchmark dataset from Stanford network data collections. The rule-based approach will not work on it as it needs completely unstructured datalogs to extract rules. We have applied our approach and rule-based algorithm approach to discover the number of change patterns and we can see in the Figure 57 that our approach has extracted more change patterns because in the ChaMining system each node is

[18] https://snap.stanford.edu/data/cit-HepPh.html

used to test if it is changed instead of using frequent patterns technique using in the rule-based method.

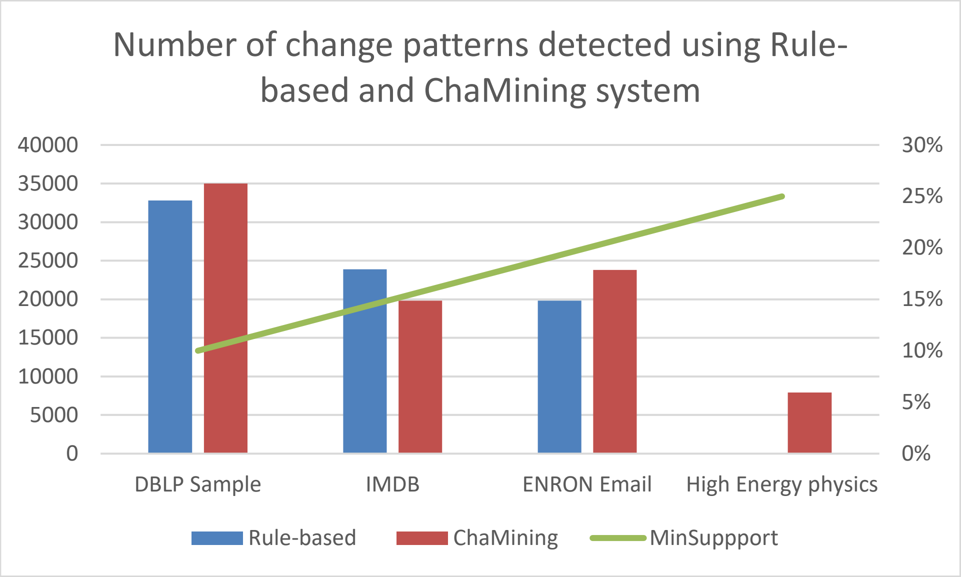


Figure 57 Number of change patterns discovered using Rule-based and ChaMining systems

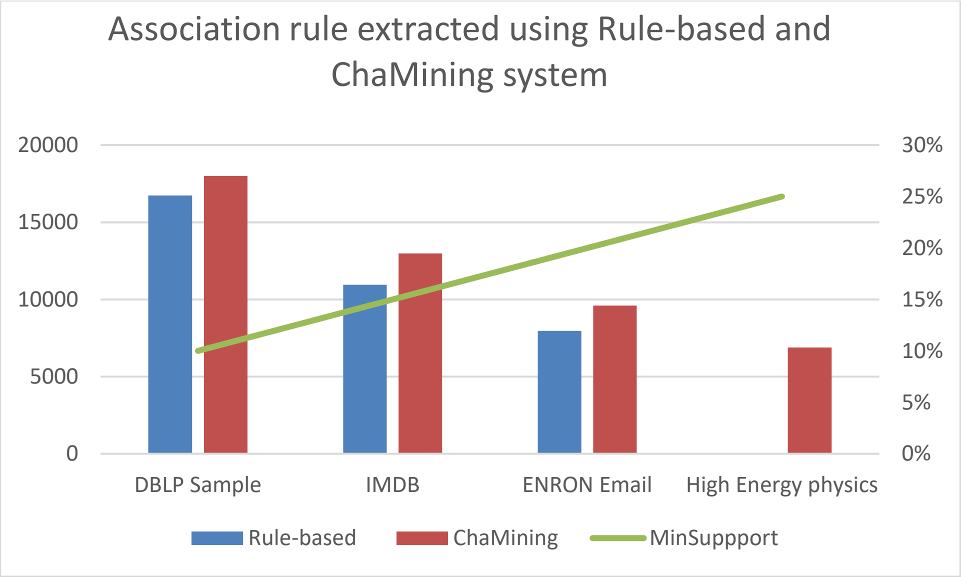


Figure 58 Number of rules extracted using Rule-based and ChaMining system

Similarly, our approach also works better on the discovery of rules between the nodes of the system because we have used employed ontology powered knowledge graphs to extract rules (see Figure 58) based on the specified minimum support and confidence. Our proposed approach also provides users with understandable communities detected from the data but rule-based does not provide them. Hence, the empirical evaluation on different datasets have produced very promising results. This empirical evaluation has shown experimentally the effectiveness of our developed system. Therefore, we can say our approach worked effectively on the change discovery in contrast to the rule-based algorithm.

7.9 Information loss calculation

The following formula has been used to calculate the percentage of information loss (PIL):

$$PIL = \frac{ChaMining\ Rules - Rule\ system}{Total\ Rules} * 100\%$$

Data set	Total No of Change Rules	No of Change Rules extracted by ChaMining System	No of Change Rules extracted by Rule-based	Reduction in information loss by ChaMining
DBLP	22560	18000	16139	8.2%
IMDB	23490	12983	10927	8.7%
Enron Email	17700	9593	7953	9.2%

Table 7.5 Information loss calculation between ChaMining and Rule-based systems

It can be seen in the table 7.5 the information loss is reduce by around 9% using ChaMining system with comparison to Rule-based system.

7.10 Comparing the Developed system with Other existing algorithms and frameworks

Although the other unsupervised algorithms discussed in related work of Chapter 3 only detect change point on a single type of data, DHINs are multi-type and contains rich semantics. We have also checked the performance of our developed system against them but all of them failed to provide results as our data is multi-type so we have compared our approach with DeltaCon by using single node based Enron email data. The codes of CHRG, SizeCPD, DeltaCon and DynSnap are available on GitHub[19]. After applying our developed method and the DeltaCon method on enron email data we can clearly see that our method performs better than DeltaCon algorithm. Table 7.6 of evaluation show that the proposed system is novel and have state of the art methodology to discover changes in DHINs.

[19] https://github.com/roger40/CINS_MLgroup/tree/master/Paper codes/ Change point detection

Method	Enron email		
	P	R	F_1
DeltaCon	38.1	13.2	19.6
ChaMining	73.34	61.3	66.78

Table 7.6 Empirical evaluation of ChaMining system with DeltaCon Algorithm

The Table 7.7 presents a comparison between the developed ChaMining system with the existing algorithms and frameworks by showing the key characteristics of each algorithms and frameworks.

System characteristics	**Existing algorithms and Frameworks**					
	CHRG	EdgeMonitoring	SizeCPD	DeltaCon	CICPD	ChaMining
F1	Yes	No	Yes	No	Yes	Yes
Precision	Yes	No	Yes	No	Yes	Yes
Recall	Yes	No	Yes	No	Yes	Yes
Time series	No	Yes	No	No	No	No
Text Clustering	No	No	No	Yes	No	Yes
Quantitative	No	Yes	No	No	No	Yes
Qualitative	No	Yes	No	No	No	Yes
Running Time	No	No	No	No	No	N/A
Evolution rules	No	No	No	No	No	No
Silhouette Coefficient	Yes	No	No	No	Yes	No
Change Point	Yes	Yes	Yes	Yes	Yes	No
Change Discovery	No	No	No	No	No	Yes

Table 7.7 A Comparison of ChaMining system with existing systems

7.11 Summary

This chapter has presented the results and empirical evaluation of our developed system with existing algorithms. We have applied the developed system on four different domains data sets and our system have effectively captured local, community level and global changes in the given dynamic heterogenous information networks. We have empirically evaluated the developed system with rule-based and DeltaCon algorithms and the results show that our framework work better that the existing system.

The next chapter will present a detailed overview of the achievements made in this study and how the thesis objectives and research questions have been addressed.

Chapter 8 Conclusions and Future Work

The world we are living in is linked and consists of real-world applications, like interlinked social networks, engineering, scientific, medical information, e-commerce, and database systems, which can be structured into an expression of information networks. Interactions among these systems or people are dynamic and multi-type and can be presented as series of network observations, each observation providing a snapshot of the relations over a transitory period. Dynamicity, multi-types, and multi-interactions of these systems make them dynamic heterogeneous information networks. A significant task in analysing these networks is changing discovery, in which we identify the change patterns and quantify what kind of rich information has changed over the different temporal snapshots of these networks.

This research investigates the use of knowledge engineering processes and data mining techniques to discover changes in dynamic heterogeneous information networks, by developing a novel change mining framework called ChaMining. The core goal of the developed system is to construct a temporal knowledge-base for domain-specific dynamic heterogeneous information networks and then use these knowledge graphs to extract local, community and global level change by employing data mining methods.

This final chapter summarises the research conducted in this study and describes the future directions that can be followed in this research. This chapter will also evaluate the results of this study against the aims and objectives as well as revisit the main contributions of this research.

8.1 Summary

The first phase of this research has conducted a background literature review of heterogeneous information networks (HINs) because it is important to understand HINs theoretical foundations before working with the concepts of dynamic heterogeneous information networks (DHINs). Chapter 2 gives an overview of the basics of HINs and describes a range of data mining methods including similarity, classification, clustering, link prediction and ranking based analysis of heterogeneous information networks. Chapter 3 describes why DHINs change discovery is important? And what are the applications of DHINS? This chapter also provides an in-depth survey of existing dynamic heterogeneous information network changes discovery methods which include pattern-based, rule-based, link prediction, supervised learning (classification, regression etcetera), and unsupervised learning (generative, and feature extraction based) methods. Chapter 4 presents the overview of semantic web technologies from the perspective of data mining methods. Moreover, it also explains the advantages of using ontology-based data mining over data mining without knowledge-base.

Change discovery from dynamic heterogeneous information networks is performed by taking DHINs data at different time intervals. Then apply the generative, feature-based, statistical, rule-based, or data mining methods to extract the change patterns at different temporal snapshots and finally combining the obtained results to see the overall change in the network. This study has explored the change discovery from DHINs by combining knowledge-engineering and data mining methods. Chapters 5, 6 and 7 are the core chapters of this research. These chapters support the completion of the research aim and objectives stated in chapter 1.

Chapter 5 has proposed a new framework to discover local, global and community level changes from structured, semi-structured and unstructured dynamic heterogeneous network data. The proposed system has been divided into three layers (pre-processing, knowledge engineering, and change detection or discovery). Before the knowledge engineering phase can be started the data needs to be in a specific format so it could be translated into ontology. This study has proposed three algorithms to transform data into a knowledge base. The knowledge engineering phase has also used the core ontology and reuse ontology pool to map the data and then convert them into a temporal knowledge base. Then the temporal knowledge base is divided into temporal knowledge graphs based on user-specified meta-paths. These meta-paths can be divided into node or community based reduced graphs using Algorithm 4 which converts the data into a format to perform data mining tasks. Thus, the ontology powered knowledge graphs have been used in the process of data mining to detect community and global level changes. Finally, three algorithms have been introduced to detect local level changes from the node or community level graphs that have been extracted from the temporal knowledge base using user-specific meta-paths. Moreover, this chapter has helped to complete the following objectives elaborated in chapter 1:

- Construct relations from structured semi-structured and unstructured dynamic heterogeneous network data to develop knowledge-base.

- Proposed a change discovery framework based on knowledge engineering and data mining methods.
- Proposed data mining methods to be used in this framework to discover community and global changes from temporal knowledge graphs.
- Proposed three algorithms to extract stable and change (local level) patterns from developed knowledge graphs.

Chapter 6 presented the implementation of the proposed framework. It also described three datasets from different domains. These datasets were used to develop a temporal knowledge base extended from the core ontology. All the algorithms that were proposed in chapter 5 were implemented and a DBLP data mining papers data was used as a baseline to see the local, global and community level changes by applying each layer of the implemented system. This chapter has completed the following objectives specified in chapter 1.

- Implementation of the knowledge base from the DHINs dataset
- Implementation of the ChaMining system and its layers to discover community and global level changes from DHINs
- Implementation of change and stable patterns discovery algorithms to discover local level changes

Chapter 7 describes the results and evaluation of the developed system with existing change detection systems or algorithms. ChaMining system has been employed on three data sets from diverse domains and the results have also been discussed. This chapter also shows that the developed framework has been able to address the following research objective described in Chapter 1.

- Evaluated the ChaMining system by applying it on different domain-specific DHINs datasets and showed that the proposed system performed better than the existing methods in performance and information loss was also reduced.
- Develop human-understandable change discovery results
- The precision, recall and F_1 scores of our developed were better than the existing DeltaCon algorithm.

8.2 Main Findings Research Contributions

This section presents the key findings from the research conducted in this study. As discussed in Chapter 1 there were three main research questions to be addressed in this research, these questions were:

1. *How well do existing algorithms or frameworks for dynamic network change discovery perform?*
2. *Can the development of DHINs employing a novel framework embedded with knowledge engineering processes through Web Ontology Language (OWL[20]) and data mining methods work best to detect stable and change patterns in DHINs?*
3. *How does the performance of the developed framework "ChaMining" compare with other existing methods and frameworks?*

This thesis aimed to address these research questions, and the main contributions are given below.

1. *How well do existing algorithms or frameworks for dynamic network change discovery perform?*

   The research performed an in-depth survey of literature on supervised and unsupervised change discovery algorithms and frameworks applied three of the unsupervised learning methods on selected DHINs data sets. The key conclusion related to this question are:

   - Most of the change discovery methods work only on static networks.
   - Supervised and classification methods require prior information and are used to detect changes of nodes and edges between two dynamic networks, but these methods do not cover the structural and global changes of a dynamic network.
   - Unsupervised learning methods where this research also sits, use feature base and statistical methods to detect change points instead of change discovery at two-time windows of a dynamic network.
   - Most of the unsupervised algorithms and frameworks work on single type dynamic networks.

[20] https://www.w3.org/OWL/

- None of the existing methods works on the change discovery of structured, semi-structured and unstructured DHINs.

2. *Can the development of DHINs employing a novel framework embedded with knowledge engineering processes through Web Ontology Language (OWL[21]) and data mining methods work best to detect stable and change patterns in DHINs?*
    - A novel three-layered framework is developed in this study to discover changes in dynamic heterogeneous information networks employing knowledge engineering and data mining methods.
    - Three algorithms were developed to transform structured, semi-structured and unstructured data to a temporal knowledge base.
    - This framework also facilitates consistency checking of the knowledge base by using an inference engine. This research has also reused domain ontology for DHINs data. It is also new in the field of change discovery to reuse ontologies as there is no standard method to conduct ontology reuse in temporal knowledge engineering.
    - The proposed framework and methodology are better than existing methods as it detects changes at all three levels (local, community level and global).
    - Meth-path based ranking and clustering have been already studied but the use of meta-paths to develop a knowledge-base is novel and it helps to develop knowledge graphs based on user-specified constraints.
    - The community-level change discovery was obtained by implementing the data mining hierarchical and maximization clustering method.
    - The global change discovery was performed by computing out-degree centrality, eigenvector centrality, clustering coefficient centrality, betweenness centrality, influence centrality of the temporal knowledge graphs.
    - This research has calculated the similarity measure using the Jaccard similarity measuring method (Niwattanakul, Singthongchai, Naenudorn, & Wanapu, 2013) between reduced temporal snapshots of the dynamic heterogeneous information graphs at different time stamps to find the changes. The stable patterns were firstly extracted by computing knowledge graphs similarity based on a specified minimum threshold. Then local changes were detected between

[21] https://www.w3.org/OWL/

nodes if the similarity score is less than the specified minimum threshold between two graph snapshots of the data.

3. *How does the performance of the developed framework "ChaMining" compare with other existing methods and frameworks?*
    - The empirical evaluation was performed with four existing change discovery methods including Rule-based, DeltaCon, CHRG and SizeCPD.
    - The results evaluation shows that our framework outperformed existing algorithms and frameworks.
    - Information loss was reduced by approximately 9% in our approach than the rule-based change discovers method.
    - Our framework also performed better in quantitative and qualitative evaluation measures and human-understandable change detection (see details in chapter 7).

The methodology adopted in this research and the developed framework were effective in achieving the aim of the research, therefore this method and framework can be applied in future works to develop a more advanced system to generate automatic inference of change patterns.

8.3 Future work

For future work we plan to extend our proposed research in the four directions:

- Unstructured data mapping to temporal knowledge base algorithm continuously required human perception and reasoning power for feature extraction and concept identification for the development of classes and properties hierarchies. We want to develop a method that can improve the knowledge engineering process of unstructured data.
- Apply the change discovery framework to other domains for example biological data to discover changes of relations between biological structures.
- This research has developed the knowledge and understanding of the formation of change discovery using developed system. This change discovery is a key to the future work of developing a change prediction system based on the principles developed in this work.
- Apply our method on Covid vaccination record dataset and find the effect of vaccination on patient before and after the vaccination.

- Extend our method using parallel processing and big data analytics to discover changes in very large amount of data.

Bibliography

Agarwal, M. K., Ramamritham, K., & Bhide, M. (2012). Real time discovery of dense clusters in highly dynamic graphs: Identifying real world events in highly dynamic environments. *Proceedings of the VLDB Endowment, 5*(10), 980-991.

Aggarwal, C. C., Ta, N., Wang, J., Feng, J., & Zaki, M. (2007). *Xproj: a framework for projected structural clustering of xml documents.* Paper presented at the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.

Aggarwal, C. C., & Wang, H. (2010). Graph data management and mining: A survey of algorithms and applications. In *Managing and Mining Graph Data* (pp. 13-68): Springer.

Agrawal, R., Imieliński, T., & Swami, A. (1993). *Mining association rules between sets of items in large databases.* Paper presented at the ACM SIGMOD Record.

Ahmed, N. M., Chen, L., Wang, Y., Li, B., Li, Y., & Liu, W. (2018). DeepEye: link prediction in dynamic networks based on non-negative matrix factorization. *Big Data Mining and Analytics, 1*(1), 19-33.

Ahmed, R., & Karypis, G. (2012). Algorithms for mining the evolution of conserved relational states in dynamic networks. *Knowledge and Information Systems, 33*(3), 603-630.

Ahmed, R., & Karypis, G. (2015a). Algorithms for mining the coevolving relational motifs in dynamic networks. *ACM Transactions on Knowledge Discovery from Data (TKDD), 10*(1), 4.

Ahmed, R., & Karypis, G. (2015b). *Mining coevolving induced relational motifs in dynamic networks.* Paper presented at the SDM Networks, the 2nd Workshop on Mining Graphs and Networks.

Aicher, C., Jacobs, A. Z., & Clauset, A. (2015). Learning latent block structure in weighted networks. *Journal of complex networks, 3*(2), 221-248.

Al-Feel, H., Koutb, M., & Suoror, H. (2009). Toward An Agreement on Semantic Web Architecture. *Europe, 49*(3), 806-810.

Albert, R., & Barabási, A.-L. (2000). Topology of evolving networks: local events and universality. *Physical review letters, 85*(24), 5234.

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics, 74*(1), 47.

Alkhamees, N., & Fasli, M. (2016). Event detection from social network streams using frequent pattern mining with dynamic support values.

Alkhamees, N., & Fasli, M. (2019). *The Dynamic-FPM: An Approach for Identifying Events from Social Networks Using Frequent Pattern Mining and Dynamic Support Values.* Paper presented at the 2019 IEEE International Conference on Big Data (Big Data).

Allahyari, M., Kochut, K. J., & Janik, M. (2014). *Ontology-based text classification into dynamically defined topics.* Paper presented at the 2014 IEEE international conference on semantic computing.

Angelova, R., Kasneci, G., & Weikum, G. (2012). Graffiti: graph-based classification in heterogeneous networks. *World Wide Web, 15*(2), 139-170.

Appice, A., Ceci, M., Turi, A., & Malerba, D. (2011). A parallel, distributed algorithm for relational frequent pattern discovery from very large data sets. *Intelligent Data Analysis, 15*(1), 69-88.

Asai, T., Abe, K., Kawasoe, S., Sakamoto, H., & Arikawa, S. (2001). Efficient Substructure Discovery from Large Semi-structured Data.

Asur, S., Parthasarathy, S., & Ucar, D. (2009a). An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD), 3*(4), 1-36.

Asur, S., Parthasarathy, S., & Ucar, D. (2009b). An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD), 3*(4), 16.

Badea, L. (2001). *A refinement operator for theories.* Paper presented at the International Conference on Inductive Logic Programming.

Bagrow, J. P. (2008). Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(05), P05001.

Balcan, N., Blum, A., & Mansour, Y. (2013). *Exploiting ontology structures and unlabeled data for learning.* Paper presented at the International Conference on Machine Learning.

Barabâsi, A.-L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., & Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications, 311*(3-4), 590-614.

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

Bellandi, A., Furletti, B., Grossi, V., & Romei, A. (2007). Ontology-driven association rule extraction: A case study. *Contexts and Ontologies Representation and Reasoning, 10*.

Berahmand, K., Haghani, S., Rostami, M., & Li, Y. (2020). A new attributed graph clustering by using label propagation in complex networks. *Journal of King Saud University-Computer and Information Sciences*.

Berger-Wolf, T. Y., & Saia, J. (2006). *A framework for analysis of dynamic social networks.* Paper presented at the Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25-71): Springer.

Berlingerio, M., Bonchi, F., Bringmann, B., & Gionis, A. (2009). Mining graph evolution rules. In *Machine learning and knowledge discovery in databases* (pp. 115-130): Springer.

Berlingerio, M., Coscia, M., Giannotti, F., Monreale, A., & Pedreschi, D. (2013). Evolving networks: Eras and turning points. *Intell. Data Anal., 17*(1), 27-48.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). Scientific American: Feature Article: The Semantic Web: May 2001. *Scientific American, 4*.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American, 284*(5), 34-43.

Besson, J., Robardet, C., Boulicaut, J.-F., & Rome, S. (2005). Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis, 9*(1), 59-82.

Bian, R., Koh, Y. S., Dobbie, G., & Divoli, A. (2019). *Network embedding and change modeling in dynamic heterogeneous networks.* Paper presented at the Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.

Bienenstock, E. J., Bonacich, P., & Oliver, M. (1990). The effect of network density and homogeneity on attitude polarization. *Social Networks, 12*(2), 153-172.

Biggs, N. (1993). *Algebraic graph theory*: Cambridge university press.

Birand, B., Zafer, M., Zussman, G., & Lee, K.-W. (2011). *Dynamic graph properties of mobile networks under levy walk mobility.* Paper presented at the 2011 IEEE eighth international conference on mobile ad-hoc and sensor systems.

Bird, S. (2006). *NLTK: the natural language toolkit.* Paper presented at the Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions.

Bliss, C. A., Frank, M. R., Danforth, C. M., & Dodds, P. S. (2014). An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science, 5*(5), 750-764.

Bollen, J., Rodriguez, M. A., Van de Sompel, H., Balakireva, L. L., & Hagberg, A. (2007). *The largest scholarly semantic network... ever.* Paper presented at the Proceedings of the 16th international conference on World Wide Web.

Bonchi, F., Castillo, C., Gionis, A., & Jaimes, A. (2011). Social network analysis and mining for business applications. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 22.

Boobalan, M. P., Lopez, D., & Gao, X. (2016). Graph clustering using k-Neighbourhood Attribute Structural similarity. *Applied Soft Computing*.

Borgwardt, K. M., Kriegel, H.-P., & Wackersreuther, P. (2006). *Pattern mining in frequent dynamic subgraphs.* Paper presented at the Data Mining, 2006. ICDM'06. Sixth International Conference on.

Boyd, D. M., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of computer-mediated communication, 13*(1), 210-230.

Brin, S., Motwani, R., Page, L., & Winograd, T. (1998). What can you do with a web in your pocket? *IEEE Data Eng. Bull., 21*(2), 37-47.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine.

Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks, 56*(18), 3825-3833.

Broumi, S., & Smarandache, F. (2014). *Cosine similarity measure of interval valued neutrosophic sets*: Infinite Study.

Buckland, M., & Gey, F. (1994). The relationship between recall and precision. *Journal of the American society for information science, 45*(1), 12-19.

Bunke, H., Münger, A., & Jiang, X. (1999). Combinatorial search versus genetic algorithms: A case study based on the generalized median graph problem. *Pattern recognition letters, 20*(11-13), 1271-1277.

Busarov, V., Grafeeva, N., & Mikhailova, E. (2016). *A Comparative Analysis of Algorithms for Mining Frequent Itemsets.* Paper presented at the International Baltic Conference on Databases and Information Systems.

Campello, R. J., Moulavi, D., & Sander, J. (2013). *Density-based clustering based on hierarchical density estimates.* Paper presented at the Pacific-Asia conference on knowledge discovery and data mining.

Cao, J., Wang, S., Wen, D., Peng, Z., Philip, S. Y., & Wang, F.-y. (2020). Mutual clustering on comparative texts via heterogeneous information networks. *Knowledge and Information Systems, 62*(1), 175-202.

Cao, X., Zheng, Y., Shi, C., Li, J., & Wu, B. (2016). *Link prediction in schema-rich heterogeneous information network.* Paper presented at the Pacific-asia conference on knowledge discovery and data mining.

Casteigts, A., Flocchini, P., Quattrociocchi, W., & Santoro, N. (2012). Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems, 27*(5), 387-408.

Cerf, L., Nguyen, T. B. N., & Boulicaut, J.-F. (2009). Discovering relevant cross-graph cliques in dynamic networks. In *Foundations of Intelligent Systems* (pp. 513-522): Springer.

Chakrabarti, D., Kumar, R., & Tomkins, A. (2006). *Evolutionary clustering.* Paper presented at the Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.

Chapanond, A., Krishnamoorthy, M. S., & Yener, B. (2005). Graph theoretic and spectral analysis of Enron email data. *Computational & Mathematical Organization Theory, 11*(3), 265-281.

Chen, D., Yuan, Z., Chen, B., & Zheng, N. (2016). *Similarity learning with spatial constraints for person re-identification.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Chen, K.-J., Chen, Y., Li, Y., & Han, J. (2016). A supervised link prediction method for dynamic networks. *Journal of Intelligent & Fuzzy Systems, 31*(1), 291-299.

Chen, X., Jiang, Y., Wu, Y., Wei, X., & Lu, X. (2020). *A Meta Graph-Based Top-k Similarity Measure for Heterogeneous Information Networks.* Paper presented at the International Conference on Intelligent Computing.

Chen, Y., Sanghavi, S., & Xu, H. (2014). Improved graph clustering. *IEEE Transactions on Information Theory, 60*(10), 6440-6455.

Chi, Y., Song, X., Zhou, D., Hino, K., & Tseng, B. L. (2007). *Evolutionary spectral clustering by incorporating temporal smoothness.* Paper presented at the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData mining, 10*(1), 1-17.

Chiu, C., & Zhan, J. (2018). Deep learning for link prediction in dynamic networks using weak estimators. *IEEE Access, 6*, 35937-35945.

Chuang, H. Y., Lee, E., Liu, Y. T., Lee, D., & Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular systems biology, 3*(1), 140.

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

Chung, F. R. (1997). *Spectral graph theory* (Vol. 92): American Mathematical Soc.

Condon, A., & Karp, R. M. (2001). Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms, 18*(2), 116-140.

Consortium, G. O. (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic acids research, 32*(suppl_1), D258-D261.

Cortes, C., & Pregibon, D. (1998). *Giga-Mining.* Paper presented at the KDD.

Cunningham, D. M. H., & Bontcheva, K. (2011). *Text Processing with GATE (Version 6)*: University of Sheffield D.

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities, 36*(2), 223-254.

Dakiche, N., Tayeb, F. B.-S., Slimani, Y., & Benatchba, K. (2019). Tracking community evolution in social networks: A survey. *Information processing & management, 56*(3), 1084-1102.

Dalamagas, T., Cheng, T., Winkel, K.-J., & Sellis, T. (2005). *Clustering XML documents using structural summaries.* Paper presented at the Current Trends in Database Technology-EDBT 2004 Workshops.

Dalwadi, N., Nagar, B., & Makwana, A. (2012). Semantic Web And Comparative Analysis of Inference Engines. *Int. J. of Computer Science and Information Technologies, 3*(3), 3843-3847.

De Raedt, L., & Kramer, S. (2001). *The levelwise version space algorithm and its application to molecular fragment finding.* Paper presented at the International Joint Conference on Artificial Intelligence.

de Vergara, J. E. L., Villagrá, V. A., & Berrocal, J. (2004). *Application of the Web Ontology Language to define management information specifications.* Paper presented at the Proceedings of the of the HP Openview University Association Eleventh Plenary Workshop, Paris, France.

Deng, H., Han, J., Zhao, B., Yu, Y., & Lin, C. X. (2011). *Probabilistic topic models with biased propagation on heterogeneous information networks.* Paper presented at the Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.

Deng, H., Zhao, B., & Han, J. (2011). *Collective topic modeling for heterogeneous networks.* Paper presented at the Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval.

Desikan, P., & Srivastava, J. (2006). Mining temporally changing web usage graphs. In *Advances in Web Mining and Web Usage Analysis* (pp. 1-17): Springer.

Diao, K., Farmani, R., Fu, G., & Butler, D. (2014). Vulnerability assessment of water distribution systems using directed and undirected graph theory.

Diesner, J., Frantz, T. L., & Carley, K. M. (2005). Communication networks from the Enron email corpus "It's always about the people. Enron is no different". *Computational & Mathematical Organization Theory, 11*(3), 201-228.

Dikaiakos, M. D., Katsaros, D., Mehra, P., Pallis, G., & Vakali, A. (2009). Cloud computing: Distributed internet computing for IT and scientific research. *IEEE Internet computing, 13*(5), 10-13.

Dillon, M. (1983). Introduction to modern information retrieval: G. Salton and M. McGill. McGraw-Hill, New York (1983). xv+ 448 pp., $32.95 ISBN 0-07-054484-0. In: Pergamon.

Ding, C. H., He, X., Zha, H., Gu, M., & Simon, H. D. (2001). *A min-max cut algorithm for graph partitioning and data clustering.* Paper presented at the Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on.

Ding, L., Kolari, P., Ding, Z., & Avancha, S. (2007). Using ontologies in the semantic web: A survey. In *Ontologies* (pp. 79-113): Springer.

do Nascimento, A. (2003). *Quilombo: vida, problemas e aspirações do negro*: Editora 34.

Domingue, J., Fensel, D., & Hendler, J. A. (2011). *Handbook of semantic web technologies*: Springer Science & Business Media.

Dou, D., Wang, H., & Liu, H. (2015). *Semantic data mining: A survey of ontology-based approaches.* Paper presented at the Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015).

Du, Y., Guo, W., Liu, J., & Yao, C. (2019). Classification by multi-semantic meta path and active weight learning in heterogeneous information networks. *Expert Systems with Applications, 123,* 227-236.

Duan, D., Li, Y., Li, R., & Lu, Z. (2012). Incremental K-clique clustering in dynamic social networks. *Artificial Intelligence Review, 38*(2), 129-147.

Elmacioglu, E., & Lee, D. (2005). On six degrees of separation in DBLP-DB and more. *ACM SIGMOD Record, 34*(2), 33-40.

Erdős, P., & Rényi, A. (1959). Some further statistical properties of the digits in Cantor's series. *Acta Mathematica Hungarica, 10*(1-2), 21-29.

Ertoz, L., Steinbach, M., & Kumar, V. (2002). *A new shared nearest neighbor clustering algorithm and its applications.* Paper presented at the Workshop on clustering high dimensional data and its applications at 2nd SIAM international conference on data mining.

Eubank, S., Guclu, H., Kumar, V. A., Marathe, M. V., Srinivasan, A., Toroczkai, Z., & Wang, N. (2004). Modelling disease outbreaks in realistic urban social networks. *Nature, 429*(6988), 180-184.

Ferlez, J., Faloutsos, C., Leskovec, J., Mladenic, D., & Grobelnik, M. (2008). *Monitoring network evolution using MDL.* Paper presented at the Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on.

Fjällström, P.-O. (1998). Algorithms for graph partitioning: A survey. *Linköping electronic articles in computer and information science, 3*(10).

Fodeh, S., Punch, B., & Tan, P.-N. (2011). On ontology-driven document clustering using core semantic features. *Knowledge and Information Systems, 28*(2), 395-421.

Franklin, B. J. (1971). *Research methods: Issues and insights*: Wadsworth Publishing Company.

Friedman, T. L. (2005). *The world is flat: A brief history of the twenty-first century*: Macmillan.

Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., & Han, J. (2010). *On community outliers and their efficient detection in information networks.* Paper presented at the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.

Gao, X., Chen, J., Zhan, Z., & Yang, S. (2020). Learning heterogeneous information network embeddings via relational triplet network. *Neurocomputing, 412*, 31-41.

Gaston, M., Kraetzl, M., & Wallis, W. (2006). Using graph diameter for change detection in dynamic networks. *Australas. J Comb., 35*, 299-312.

Getoor, L. (2005). Link-based classification. In *Advanced methods for knowledge discovery from complex data* (pp. 189-207): Springer.

Görke, R., Maillard, P., Schumm, A., Staudt, C., & Wagner, D. (2013). Dynamic graph clustering combining modularity and smoothness. *Journal of Experimental Algorithmics (JEA), 18*, 1.5.

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., & Sattler, U. (2008). OWL 2: The next step for OWL. *Journal of Web Semantics, 6*(4), 309-322.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition, 5*(2), 199-220.

Grus, W. E., Shi, P., & Zhang, J. (2007). Largest vertebrate vomeronasal type 1 receptor gene repertoire in the semiaquatic platypus. *Molecular biology and evolution, 24*(10), 2153-2157.

Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy* (Vol. 46): IOS press.

Gupta, A., Thakur, H. K., Garg, A., & Garg, D. (2016). Mining and analysis of periodic patterns in weighted directed dynamic network. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), 7*(1), 1-26.

Gupta, A., Thakur, H. K., Jain, B., & Garg, S. (2015). Framework for mining regular patterns in dynamic networks. *International Journal of Knowledge Engineering and Data Mining, 3*(3-4), 299-314.

Gupta, M., Gao, J., & Han, J. (2013). *Community distribution outlier detection in heterogeneous information networks.* Paper presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases.

Gupta, M., Kumar, P., & Bhasker, B. (2017). HeteClass: A Meta-path based framework for transductive classification of objects in heterogeneous information networks. *Expert Systems with Applications, 68*, 106-122.

Halder, S., Samiullah, M., & Lee, Y.-K. (2017). Supergraph based periodic pattern mining in dynamic social networks. *Expert Systems with Applications, 72*, 430-442.

Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems, 5*(4), 83-124.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*: Elsevier.

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. (2001). *Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth.* Paper presented at the proceedings of the 17th international conference on data engineering.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record, 29*(2), 1-12.

Hartmann, T., Kappes, A., & Wagner, D. (2014). Clustering evolving networks. *arXiv preprint arXiv:1401.3516*.

Hartuv, E., & Shamir, R. (2000). A clustering algorithm based on graph connectivity. *Information processing letters, 76*(4-6), 175-181.

He, C., Liu, S., Zhang, L., & Zheng, J. (2019). A fuzzy clustering based method for attributed graph partitioning. *Journal of Ambient Intelligence and Humanized Computing, 10*(9), 3399-3407.

He, J., Bailey, J., & Zhang, R. (2014). *Exploiting transitive similarity and temporal dynamics for similarity search in heterogeneous information networks.* Paper presented at the International Conference on Database Systems for Advanced Applications.

Heard, N. A., Weston, D. J., Platanioti, K., & Hand, D. J. (2010). Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics, 4*(2), 645-662.

Held, P., & Kruse, R. (2013). *Analysis and visualization of dynamic clusterings.* Paper presented at the System sciences (hicss), 2013 46th hawaii international conference on.

Hochstenbach, M. E., & Notay, Y. (2006). The Jacobi–Davidson method. *GAMM-Mitteilungen, 29*(2), 368-382.

Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks, 5*(2), 109-137.

Holme, P., Edling, C. R., & Liljeros, F. (2004). Structure and time evolution of an Internet dating community. *Social Networks, 26*(2), 155-174.

Hopcroft, J., Khan, O., Kulis, B., & Selman, B. (2004). Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences, 101*(suppl 1), 5249-5253.

Horridge, M., & Bechhofer, S. (2011). The owl api: A java api for owl ontologies. *Semantic web, 2*(1), 11-21.

Horridge, M., Drummond, N., Goodwin, J., Rector, A. L., Stevens, R., & Wang, H. (2006). *The Manchester OWL syntax.* Paper presented at the OWLed.

Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2009). A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 2. *The university of Manchester, 107*.

Horridge, M., Knublauch, H., Rector, A., Stevens, R., & Wroe, C. (2004). A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools edition 1.0. *University of Manchester*.

Hotho, A., Staab, S., & Stumme, G. (2003). *Ontologies improve text document clustering.* Paper presented at the Third IEEE international conference on data mining.

Hramov, A. E., Frolov, N. S., Maksimenko, V. A., Makarov, V. V., Koronovskii, A. A., Garcia-Prieto, J., . . . Pisarchik, A. N. (2018). Artificial neural network detects human uncertainty. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 28*(3), 033607.

Hu, H.-B., & Wang, X.-F. (2009). Disassortative mixing in online social networks. *EPL (Europhysics Letters), 86*(1), 18003.

Huan, J., Wang, W., & Prins, J. (2003). *Efficient mining of frequent subgraphs in the presence of isomorphism.* Paper presented at the Data Mining, 2003. ICDM 2003. Third IEEE International Conference on.

Huan, J., Wang, W., Prins, J., & Yang, J. (2004). *Spin: mining maximal frequent subgraphs from graph databases.* Paper presented at the Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.

Huang, M., Zou, G., Zhang, B., Liu, Y., Gu, Y., & Jiang, K. (2018). Overlapping community detection in heterogeneous social networks via the user model. *Information Sciences, 432*, 164-184.

Ibrahim, N. M. A., & Chen, L. (2015). Link prediction in dynamic social networks by integrating different types of information. *Applied Intelligence, 42*(4), 738-750.

Idehen, K. U. (2017). Semantic Web Layer Cake Tweak, Explained. *OpenLink Software, via Medium, July, 14*.

Inokuchi, A., & Washio, T. (2010). *Mining Frequent Graph Sequence Patterns Induced by Vertices.* Paper presented at the SDM.

Inokuchi, A., Washio, T., & Motoda, H. (2000a). An apriori-based algorithm for mining frequent substructures from graph data. In *Principles of Data Mining and Knowledge Discovery* (pp. 13-23): Springer.

Inokuchi, A., Washio, T., & Motoda, H. (2000b). *An apriori-based algorithm for mining frequent substructures from graph data.* Paper presented at the European Conference on Principles of Data Mining and Knowledge Discovery.

Ivchenko, G., & Honov, S. (1998). On the jaccard similarity test. *Journal of Mathematical Sciences, 88*(6), 789-794.

Izenman, A. J. (2013). Linear discriminant analysis. In *Modern multivariate statistical techniques* (pp. 237-280): Springer.

Jacob, Y., Denoyer, L., & Gallinari, P. (2014). *Learning latent representations of nodes for classifying in heterogeneous social networks.* Paper presented at the Proceedings of the 7th ACM international conference on Web search and data mining.

Jacobs, K. (1992). Independent identically distributed (iid) random variables. In *Discrete Stochastics* (pp. 65-101): Springer.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters, 31*(8), 651-666.

Jendoubi, S., Martin, A., Liétard, L., & Yaghlane, B. B. (2014). *Classification of message spreading in a heterogeneous social network.* Paper presented at the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems.

Ji, M., Han, J., & Danilevsky, M. (2011). *Ranking-based classification of heterogeneous information networks.* Paper presented at the Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.

Ji, M., Sun, Y., Danilevsky, M., Han, J., & Gao, J. (2010). *Graph regularized transductive classification on heterogeneous information networks.* Paper presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases.

Jia, Y., Wang, Y., Jin, X., Zhao, Z., & Cheng, X. (2017). Link inference in dynamic heterogeneous information network: A knapsack-based approach. *IEEE Transactions on Computational Social Systems, 4*(3), 80-92.

Jiang, X., Munger, A., & Bunke, H. (2001). An median graphs: properties, algorithms, and applications. *IEEE Transactions on pattern analysis and machine intelligence, 23*(10), 1144-1151.

Jin, R., McCallen, S., & Almaas, E. (2007). *Trend motif: A graph mining approach for analysis of dynamic complex networks.* Paper presented at the Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on.

Jing, Y., Covell, M., & Rowley, H. A. (2010). *Comparison of clustering approaches for summarizing large populations of images.* Paper presented at the 2010 IEEE International Conference on Multimedia and Expo.

Juszczyszyn, K., Musial, K., & Budka, M. (2011). *Link prediction based on subgraph evolution in dynamic social networks.* Paper presented at the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing.

Kamis, N. H., Chiclana, F., & Levesley, J. (2018). Preference similarity network structural equivalence clustering based consensus group decision making model. *Applied Soft Computing, 67*, 706-720.

Kataria, A., & Singh, M. (2013). A review of data classification using k-nearest neighbour algorithm. *International Journal of Emerging Technology and Advanced Engineering, 3*(6), 354-360.

Katsaros, D., Pallis, G., Stamos, K., Vakali, A., Sidiropoulos, A., & Manolopoulos, Y. (2009). CDNs content outsourcing via generalized communities. *IEEE Transactions on Knowledge and Data Engineering, 21*(1), 137-151.

Kaur, R., & Singh, S. (2016). A survey of data mining and social network analysis based anomaly detection techniques. *Egyptian informatics journal, 17*(2), 199-216.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). *Maximizing the spread of influence through a social network.* Paper presented at the Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.

Kendal, S. L., & Creen, M. (2007). *An introduction to knowledge engineering*: Springer.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). *Logistic regression*: Springer.

Kleinberg, J. (2000). *The small-world phenomenon: An algorithmic perspective.* Paper presented at the Proceedings of the thirty-second annual ACM symposium on Theory of computing.

Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A. S. (1999). *The web as a graph: Measurements, models, and methods.* Paper presented at the International Computing and Combinatorics Conference.

Kong, C., Li, H., Zhang, L., Zhu, H., & Liu, T. (2019). *Link Prediction on Dynamic Heterogeneous Information Networks.* Paper presented at the International Conference on Computational Data and Social Networks.

Koren, Y., North, S. C., & Volinsky, C. (2007). Measuring and extracting proximity graphs in networks. *ACM Transactions on Knowledge Discovery from Data (TKDD), 1*(3), 12.

Kossinets, G., & Watts, D. J. (2006). Empirical analysis of an evolving social network. *Science, 311*(5757), 88-90.

Kostakis, O., & Gionis, A. (2017). *On mining temporal patterns in dynamic graphs, and other unrelated problems.* Paper presented at the International Conference on Complex Networks and their Applications.

Kothari, C. R. (2004). *Research methodology: Methods and techniques*: New Age International.

Koutra, D., Ke, T.-Y., Kang, U., Chau, D. H. P., Pao, H.-K. K., & Faloutsos, C. (2011). *Unifying guilt-by-association approaches: Theorems and fast algorithms.* Paper presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases.

Koutra, D., Parikh, A., Ramdas, A., & Xiang, J. (2011). *Algorithms for graph similarity and subgraph matching.* Paper presented at the Proc. Ecol. Inference Conf.

Koutra, D., Shah, N., Vogelstein, J. T., Gallagher, B., & Faloutsos, C. (2016). Deltacon: Principled massive-graph similarity function with attribution. *ACM Transactions on Knowledge Discovery from Data (TKDD), 10*(3), 1-43.

Koutra, D., Vogelstein, J. T., & Faloutsos, C. (2013). *Deltacon: A principled massive-graph similarity function.* Paper presented at the Proceedings of the 2013 SIAM International Conference on Data Mining.

Kuck, J., Zhuang, H., Yan, X., Cam, H., & Han, J. (2015). *Query-based outlier detection in heterogeneous information networks.* Paper presented at the Advances in database technology: proceedings. International Conference on Extending Database Technology.

Kuramochi, M., & Karypis, G. (2004a). An efficient algorithm for discovering frequent subgraphs. *Knowledge and Data Engineering, IEEE Transactions on, 16*(9), 1038-1051.

Kuramochi, M., & Karypis, G. (2004b). *Finding Frequent Patterns in a Large Sparse Graph.* Paper presented at the SDM.

Kuramochi, M., & Karypis, G. (2004c). *Grew-a scalable frequent subgraph discovery algorithm.* Paper presented at the Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on.

Kuramochi, M., & Karypis, G. (2005). Finding frequent patterns in a large sparse graph*. *Data Mining and Knowledge Discovery, 11*(3), 243-271.

Lahiri, M., & Berger-Wolf, T. Y. (2008). *Mining periodic behavior in dynamic social networks.* Paper presented at the 2008 Eighth IEEE International Conference on Data Mining.

Langville, A. N., Meyer, C. D., & FernÁndez, P. (2008). Google's pagerank and beyond: The science of search engine rankings. *The Mathematical Intelligencer, 30*(1), 68-69.

Lao, N., & Cohen, W. W. (2010). Relational retrieval using a combination of path-constrained random walks. *Machine Learning, 81*(1), 53-67.

Latapy, M., Magnien, C., & Del Vecchio, N. (2008). Basic notions for the analysis of large two-mode networks. *Social Networks, 30*(1), 31-48.

Latora, V., & Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical review letters, 87*(19), 198701.

Lawrence, S., Giles, C. L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *Computer, 32*(6), 67-71.

Lee, C., & Cunningham, P. (2013). Benchmarking community detection methods on social media data. *arXiv preprint arXiv:1302.0739*.

Lee, M. L., Yang, L. H., Hsu, W., & Yang, X. (2002). *XClust: clustering XML schemas for effective integration.* Paper presented at the Proceedings of the eleventh international conference on Information and knowledge management.

Leskovec, J., Chakrabarti, D., Kleinberg, J., & Faloutsos, C. (2005). *Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication.* Paper presented at the European conference on principles of data mining and knowledge discovery.

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2005). *Graphs over time: densification laws, shrinking diameters and possible explanations.* Paper presented at the Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD), 1*(1), 2.

Lewis, K., Gonzalez, M., & Kaufman, J. (2012). Social selection and peer influence in an online social network. *Proceedings of the National Academy of Sciences, 109*(1), 68-72.

Li, L., Yang, Y., & Wu, B. (2006). *Implementation of agent-based ontology mapping and integration.* Paper presented at the 2006 IEEE International Conference on e-Business Engineering (ICEBE'06).

Li, S., Armstrong, C. M., Bertin, N., Ge, H., Milstein, S., Boxem, M., . . . Hao, T. (2004). A map of the interactome network of the metazoan C. elegans. *Science, 303*(5657), 540-543.

Li, T., Wang, B., Jiang, Y., Zhang, Y., & Yan, Y. (2018). Restricted Boltzmann machine-based approaches for link prediction in dynamic networks. *IEEE Access, 6*, 29940-29951.

Li, X., Ding, D., Kao, B., Sun, Y., & Mamoulis, N. (2020). Leveraging Meta-path Contexts for Classification in Heterogeneous Information Networks. *arXiv preprint arXiv:2012.10024*.

Li, X., Kao, B., Ren, Z., & Yin, D. (2019). *Spectral clustering in heterogeneous information networks.* Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

Li, Z., Sun, D., Zhu, R., & Lin, Z. (2017). Detecting event-related changes in organizational networks using optimized neural network models. *PLoS ONE, 12*(11), e0188733.

Li, Z., & Wang, H. (2018). *A Weighted Similarity Measure Based on Meta Structure in Heterogeneous Information Networks.* Paper presented at the Pacific Rim Knowledge Acquisition Workshop.

Lian, W., Cheung, D. W.-L., Mamoulis, N., & Yiu, S.-M. (2004). An efficient and scalable algorithm for clustering XML documents by structure. *Knowledge and Data Engineering, IEEE Transactions on, 16*(1), 82-96.

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

Lisi, F. A., & Malerba, D. (2004). Inducing multi-level association rules from multiple relations. *Machine Learning, 55*(2), 175-210.

Liu, F., Choi, D., Xie, L., & Roeder, K. (2018). Global spectral clustering in dynamic networks. *Proceedings of the National Academy of Sciences, 115*(5), 927-932.

Liu, L., & Wang, S. (2020). Meta-path-based outlier detection in heterogeneous information network. *Frontiers of Computer Science, 14*(2), 388-403.

Liu, X., & Yang, J. (2012). Mining Buyer Behavior Patterns Based on Dynamic Group-Buying Network. In *Computational Social Networks* (pp. 161-181): Springer.

Loglisci, C., Ceci, M., & Malerba, D. (2013). Discovering evolution chains in dynamic networks. In *New Frontiers in Mining Complex Patterns* (pp. 185-199): Springer.

Loglisci, C., Ceci, M., & Malerba, D. (2015). Relational mining for discovering changes in evolving networks. *Neurocomputing, 150*, 265-288.

Loglisci, C., & Malerba, D. (2015). *Mining periodic changes in complex dynamic data through relational pattern discovery.* Paper presented at the International Workshop on New Frontiers in Mining Complex Patterns.

Luo, C., Guan, R., Wang, Z., & Lin, C. (2014). *Hetpathmine: A novel transductive classification algorithm on heterogeneous information networks.* Paper presented at the European Conference on Information Retrieval.

Luo, C., Pang, W., & Wang, Z. (2014). *Semi-supervised clustering on heterogeneous information networks.* Paper presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining.

Lusseau, D., & Newman, M. E. (2004). Identifying the role that animals play in their social networks. *Proceedings of the Royal Society of London B: Biological Sciences, 271*(Suppl 6), S477-S481.

Ma, G., He, L., Cao, B., Zhang, J., Philip, S. Y., & Ragin, A. B. (2016). *Multi-graph clustering based on interior-node topology with applications to brain networks.* Paper presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases.

Ma, X., & Dong, D. (2017). Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering, 29*(5), 1045-1058.

Malerba, D., & Lisi, F. A. (2001). *Discovering associations between spatial objects: An ILP application.* Paper presented at the International Conference on Inductive Logic Programming.

Martínez-Romero, M., O'Connor, M. J., Egyedi, A. L., Willrett, D., Hardi, J., Graybeal, J., & Musen, M. A. (2019). Using association rule mining and ontologies to generate metadata recommendations from multiple biomedical databases. *Database, 2019*.

Maslennikov, O. V., & Nekorkin, V. I. (2015). Evolving dynamical networks with transient cluster activity. *Communications in Nonlinear Science and Numerical Simulation, 23*(1), 10-16.

Mei, J.-P., Lv, H., Yang, L., & Li, Y. (2019). Clustering for heterogeneous information networks with extended star-structure. *Data Mining and Knowledge Discovery, 33*(4), 1059-1087.

Menezes, G. V., Ziviani, N., Laender, A. H., & Almeida, V. (2009). *A geographical analysis of knowledge production in computer science.* Paper presented at the Proceedings of the 18th international conference on World wide web.

Menichetti, G., Dall'Asta, L., & Bianconi, G. (2014). Network controllability is determined by the density of low in-degree and out-degree nodes. *Physical review letters, 113*(7), 078701.

Merigó, J. M., & Gil-Lafuente, A. M. (2008). Using the OWA operator in the Minkowski distance. *International Journal of Computer Science, 3*(3), 149-157.

Messmer, B. T., & Bunke, H. (1998). A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on pattern analysis and machine intelligence, 20*(5), 493-504.

Miller, E. (1998). An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology, 25*(1), 15-19.

Miller, H., & Mokryn, O. (2020). Size agnostic change point detection framework for evolving networks. *PLoS ONE, 15*(4), e0231035.

Mina, M., & Guzzi, P. H. (2014). Improving the robustness of local network alignment: Design and extensive assessmentof a Markov clustering-based approach. *IEEE/ACM transactions on computational biology and bioinformatics, 11*(3), 561-572.

Mitchell, T. M. (2006). *The discipline of machine learning* (Vol. 9): Carnegie Mellon University, School of Computer Science, Machine Learning ….

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to linear regression analysis* (Vol. 821): John Wiley & Sons.

Moody, J., McFarland, D., & Bender-deMoll, S. (2005). Dynamic network visualization1. *American Journal of Sociology, 110*(4), 1206-1241.

Moradi, P., & Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. *Knowledge-Based Systems, 84*, 144-161.

Moreno, S., & Neville, J. (2013). *Network hypothesis testing using mixed Kronecker product graph models.* Paper presented at the 2013 IEEE 13th International Conference on Data Mining.

Motik, B., Patel-Schneider, P. F., Parsia, B., Bock, C., Fokoue, A., Haase, P., . . . Sattler, U. (2009). OWL 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation, 27*(65), 159.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM Review, 45*(2), 167-256.

Newman, M. E. (2004). Analysis of weighted networks. *Physical Review E, 70*(5), 056131.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E, 69*(2), 026113.

Newman, M. E., Watts, D. J., & Strogatz, S. H. (2002). Random graph models of social networks. *Proceedings of the National Academy of Sciences, 99*(suppl 1), 2566-2572.

Nijssen, S., & Kok, J. N. (2004). *A quickstart in frequent structure mining can make a difference.* Paper presented at the Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). *Using of Jaccard coefficient for keywords similarity.* Paper presented at the Proceedings of the international multiconference of engineers and computer scientists.

Nowicki, K., & Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical association, 96*(455), 1077-1087.

Ovaska, K., Laakso, M., & Hautaniemi, S. (2008). Fast gene ontology based clustering for microarray experiments. *BioData mining, 1*(1), 1-8.

Pandit, S., & Gupta, S. (2011). A comparative study on distance measuring approaches for clustering. *International Journal of Research in Computer Science, 2*(1), 29-31.

Parimala, M., & Lopez, D. (2015). *Graph clustering based on structural attribute neighborhood similarity (SANS).* Paper presented at the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT).

Park, S.-T., & Pennock, D. M. (2007). *Applying collaborative filtering techniques to movie search for better ranking and browsing.* Paper presented at the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.

Parsia, B., Matentzoglu, N., Gonçalves, R. S., Glimm, B., & Steigmiller, A. (2017). The OWL reasoner evaluation (ORE) 2015 competition report. *Journal of Automated Reasoning, 59*(4), 455-482.

Peel, L., & Clauset, A. (2015). *Detecting change points in the large-scale structure of evolving networks.* Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

Pei, J., Han, J., Lu†, H., Nishio, S., Tang, S., & Yang, D. (2007). H-Mine: Fast and space-preserving frequent pattern mining in large databases. *IIE transactions, 39*(6), 593-605.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., . . . Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *Knowledge and Data Engineering, IEEE Transactions on, 16*(11), 1424-1440.

Pei, J., Jiang, D., & Zhang, A. (2005). *Mining cross-graph quasi-cliques in gene expression and protein interaction data.* Paper presented at the Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on.

Phyu, T. N. (2009). *Survey of classification techniques in data mining.* Paper presented at the Proceedings of the International MultiConference of Engineers and Computer Scientists.

Qi, G.-J., Aggarwal, C. C., & Huang, T. S. (2012). *On clustering heterogeneous social media objects with outlier links.* Paper presented at the Proceedings of the fifth ACM international conference on Web search and data mining.

Qiu, C., Chen, W., Wang, T., & Lei, K. (2015). *Overlapping community detection in directed heterogeneous social network.* Paper presented at the International Conference on Web-Age Information Management.

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E, 76*(3), 036106.

Raghavan, V., Galstyan, A., & Tartakovsky, A. G. (2013). Hidden Markov models for the activity profile of terrorist groups. *The Annals of Applied Statistics*, 2402-2430.

Ramzan, A., Wang, H., & Buckingham, C. (2014). Representing Human Expertise by the OWL Web Ontology Language to Support Knowledge Engineering in Decision Support Systems. *Studies in health technology and informatics, 207*, 290-299.

Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., . . . Wroe, C. (2004). *OWL pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns.* Paper presented at the International Conference on Knowledge Engineering and Knowledge Management.

Riedy, E. J., Meyerhenke, H., Ediger, D., & Bader, D. A. (2012). Parallel community detection for massive graphs. In *Parallel Processing and Applied Mathematics* (pp. 286-296): Springer.

Rish, I. (2001). *An empirical study of the naive Bayes classifier.* Paper presented at the IJCAI 2001 workshop on empirical methods in artificial intelligence.

Robardet, C. (2009). *Constraint-based pattern mining in dynamic graphs.* Paper presented at the Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on.

Rodriguez, M. A., Bollen, J., & Van de Sompel, H. (2007). *A practical ontology for the large-scale modeling of scholarly artifacts and their usage.* Paper presented at the Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries.

Rodriguez, M. A., & Shinavier, J. (2010). Exposing multi-relational networks to single-relational network analysis algorithms. *Journal of Informetrics, 4*(1), 29-41.

Rokach, L., & Maimon, O. (2005). Decision trees. In *Data mining and knowledge discovery handbook* (pp. 165-192): Springer.

Rossetti, G., & Cazabet, R. (2018). Community discovery in dynamic networks: a survey. *ACM computing surveys (CSUR), 51*(2), 1-37.

Rossi, R. G., de Paulo Faleiros, T., de Andrade Lopes, A., & Rezende, S. O. (2012). *Inductive model generation for text categorization using a bipartite heterogeneous network.* Paper presented at the 2012 IEEE 12th International Conference on Data Mining.

Ruhnau, B. (2000). Eigenvector-centrality—a node-centrality? *Social Networks, 22*(4), 357-365.

Rysz, M., Pajouh, F. M., & Pasiliao, E. L. (2018). Finding clique clusters with the highest betweenness centrality. *European Journal of Operational Research, 271*(1), 155-164.

Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., & Tsuda, K. (2009). gBoost: a mathematical programming approach to graph classification and regression. *Machine Learning, 75*(1), 69-89.

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

Schwikowski, B., Uetz, P., & Fields, S. (2000). A network of protein–protein interactions in yeast. *Nature biotechnology, 18*(12), 1257-1261.

Shi, C., Li, Y., Zhang, J., Sun, Y., & Philip, S. Y. (2016). A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering, 29*(1), 17-37.

Shi, C., & Philip, S. Y. (2017). *Heterogeneous information network analysis and applications*: Springer.

Shi, Z. (2021). *Intelligence Science: Leading the Age of Intelligence*: Elsevier.

Shoubridge, P., Kraetzl, M., Wallis, W., & Bunke, H. (2002). Detection of abnormal change in a time series of graphs. *Journal of Interconnection Networks, 3*(01n02), 85-101.

Showbridge, P., Kraetzl, M., & Ray, D. (1999). *Detection of abnormal change in dynamic networks.* Paper presented at the 1999 Information, Decision and Control. Data and Information Fusion Symposium, Signal Processing and Communications Symposium and Decision and Control Symposium. Proceedings (Cat. No. 99EX251).

Silva, J., & Willett, R. (2008). *Detection of anomalous meetings in a social network.* Paper presented at the 2008 42nd Annual Conference on Information Sciences and Systems.

Simperl, E. (2009). Reusing ontologies on the Semantic Web: A feasibility study. *Data & Knowledge Engineering, 68*(10), 905-925.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Journal of Web Semantics, 5*(2), 51-53.

Sliva, A., Subrahmanian, V., Martinez, V., & Simari, G. (2009). Cape: Automatically predicting changes in group behavior. In *Mathematical Methods in Counterterrorism* (pp. 253-269): Springer.

Soffer, S. N., & Vazquez, A. (2005). Network clustering coefficient without degree-correlation biases. *Physical Review E, 71*(5), 057101.

Song, W., & Park, S. C. (2009). Genetic algorithm for text clustering based on latent semantic indexing. *Computers & Mathematics with Applications, 57*(11-12), 1901-1907.

Sowa, J. F. (2014). *Principles of semantic networks: Explorations in the representation of knowledge*: Morgan Kaufmann.

Stanton, I., & Kliot, G. (2012). *Streaming graph partitioning for large distributed graphs.* Paper presented at the Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining.

StröEle, V., ZimbrãO, G., & Souza, J. M. (2013). Group and link analysis of multi-relational scientific social networks. *Journal of Systems and Software, 86*(7), 1819-1830.

Subrahmanian, V., Mannes, A., Roul, A., & Raghavan, R. (2013). *Indian Mujahideen: Computational analysis and public policy*: Springer Science & Business Media.

Subrahmanian, V., Mannes, A., Sliva, A., Shakarian, J., & Dickerson, J. P. (2012). *Computational analysis of terrorist groups: Lashkar-e-Taiba*: Springer Science & Business Media.

Sun, J., Faloutsos, C., Papadimitriou, S., & Yu, P. S. (2007). *Graphscope: parameter-free mining of large time-evolving graphs.* Paper presented at the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.

Sun, J., Tao, D., & Faloutsos, C. (2006). *Beyond streams and graphs: dynamic tensor analysis.* Paper presented at the Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.

Sun, X., Tong, M., Yang, J., Xinran, L., & Heng, L. (2019). *Hindom: A robust malicious domain detection system based on heterogeneous information network with transductive classification.* Paper presented at the 22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019).

Sun, Y., Aggarwal, C. C., & Han, J. (2012). Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *Proceedings of the VLDB Endowment, 5*(5), 394-405.

Sun, Y., & Han, J. (2013). Mining heterogeneous information networks: a structural analysis approach. *Acm Sigkdd Explorations Newsletter, 14*(2), 20-28.

Sun, Y., Han, J., Yan, X., Yu, P. S., & Wu, T. (2011). Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment, 4*(11), 992-1003.

Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., & Wu, T. (2009). *Rankclus: integrating clustering with ranking for heterogeneous information network analysis.* Paper presented at the Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology.

Sun, Y., Norick, B., Han, J., Yan, X., Yu, P. S., & Yu, X. (2013). Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD), 7*(3), 1-23.

Sun, Y., Tang, J., Han, J., Gupta, M., & Zhao, B. (2010). *Community evolution detection in dynamic heterogeneous information networks.* Paper presented at the Proceedings of the Eighth Workshop on Mining and Learning with Graphs.

Sun, Y., Yu, Y., & Han, J. (2009). *Ranking-based clustering of heterogeneous information networks with star network schema.* Paper presented at the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.

Sundar, C., Chitradevi, M., & Geetharamani, G. (2012). Classification of cardiotocogram data using neural network based machine learning technique. *International Journal of Computer Applications, 47*(14).

Szmrecsanyi, B. (2012). *Grammatical variation in British English dialects: A study in corpus-based dialectometry*: Cambridge University Press.

Tang, L., Liu, H., Zhang, J., & Nazeri, Z. (2008). *Community evolution in dynamic multi-mode networks.* Paper presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.

Taskar, B., Abbeel, P., & Koller, D. (2002). *Discriminative probabilistic models for relational data.* Paper presented at the Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence.

Tong, H., Papadimitriou, S., Sun, J., Yu, P. S., & Faloutsos, C. (2008). *Colibri: fast mining of large static and dynamic graphs.* Paper presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.

Trivedi, R., Dai, H., Wang, Y., & Song, L. (2017). *Know-evolve: Deep temporal reasoning for dynamic knowledge graphs.* Paper presented at the international conference on machine learning.

Vigliotti, M. G., & Hankin, C. (2015). Discovery of anomalous behaviour in temporal networks. *Social Networks, 41*, 18-25.

Wackersreuther, B., Wackersreuther, P., Oswald, A., Böhm, C., & Borgwardt, K. M. (2010). *Frequent subgraph discovery in dynamic networks.* Paper presented at the Proceedings of the Eighth Workshop on Mining and Learning with Graphs.

Wallis, W. D., Shoubridge, P., Kraetz, M., & Ray, D. (2001). Graph distances using graph union. *Pattern recognition letters, 22*(6-7), 701-704.

Wang, C.-D., Lai, J.-H., & Yu, P. (2013). *Dynamic community detection in weighted graph streams.* Paper presented at the Proc. of SDM.

Wang, C., Pan, S., Long, G., Zhu, X., & Jiang, J. (2017). *Mgae: Marginalized graph autoencoder for graph clustering.* Paper presented at the Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.

Wang, C., Song, Y., Li, H., Zhang, M., & Han, J. (2016). *Text classification with heterogeneous information network kernels.* Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

Wang, C., Sun, Y., Song, Y., Han, J., Song, Y., Wang, L., & Zhang, M. (2016). *Relsim: relation similarity search in schema-rich heterogeneous information networks.* Paper presented at the Proceedings of the 2016 SIAM International Conference on Data Mining.

Wang, L., Zhang, Y., & Feng, J. (2005). On the Euclidean distance of images. *IEEE Transactions on pattern analysis and machine intelligence, 27*(8), 1334-1339.

Wang, Q., Peng, Z., Wang, S., Philip, S. Y., Li, Q., & Hong, X. (2015). *cluTM: Content and link integrated topic model on heterogeneous information networks.* Paper presented at the International Conference on Web-Age Information Management.

Wang, R., Shi, C., Philip, S. Y., & Wu, B. (2013). *Integrating clustering and ranking on hybrid heterogeneous information network.* Paper presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining.

Wang, X., Gorlitsky, R., & Almeida, J. S. (2005). From XML to RDF: how semantic web technologies will change the design of'omic'standards. *Nature biotechnology, 23*(9), 1099-1103.

Wang, Y., Chakrabarti, A., Sivakoff, D., & Parthasarathy, S. (2017). Fast change point detection on dynamic social networks. *arXiv preprint arXiv:1705.07325*.

Wang, Y., Wang, Z., Zhao, Z., Li, Z., Jian, X., Chen, L., & Song, J. (2020). *HowSim: A General and Effective Similarity Measure on Heterogeneous Information Networks.* Paper presented at the 2020 IEEE 36th International Conference on Data Engineering (ICDE).

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications* (Vol. 8): Cambridge university press.

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

Wei, W., Barnaghi, P. M., & Bargiela, A. (2008). Search with meanings: an overview of semantic search systems. *International journal of Communications of SIWN, 3,* 76-82.

Whorton, M. R., Bokoch, M. P., Rasmussen, S. G., Huang, B., Zare, R. N., Kobilka, B., & Sunahara, R. K. (2007). A monomeric G protein-coupled receptor isolated in a high-density lipoprotein particle efficiently activates its G protein. *Proceedings of the National Academy of Sciences, 104*(18), 7682-7687.

Wielinga, B. J., Schreiber, A. T., & Breuker, J. A. (1992). KADS: A modelling approach to knowledge engineering. *Knowledge acquisition, 4*(1), 5-53.

Wu, C., Gu, Y., & Yu, G. (2019). *Dpscan: Structural graph clustering based on density peaks.* Paper presented at the International Conference on Database Systems for Advanced Applications.

Wu, X., Zhu, X., Wu, G.-Q., & Ding, W. (2013). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering, 26*(1), 97-107.

Xiong, Y., Zhu, Y., & Philip, S. Y. (2014). Top-k similarity join in heterogeneous information networks. *IEEE Transactions on Knowledge and Data Engineering, 27*(6), 1710-1723.

Xue, A., Wang, W., & Zhang, M. (2011). *Terrorist organization behavior prediction algorithm based on context subspace.* Paper presented at the International Conference on Advanced Data Mining and Applications.

Yan, X., Han, J., & Afshar, R. (2003). *CloSpan: Mining: Closed sequential patterns in large datasets.* Paper presented at the Proceedings of the 2003 SIAM international conference on data mining.

Yan, X., Zhou, X., & Han, J. (2005). *Mining closed relational graphs with connectivity constraints.* Paper presented at the Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.

Yang, C., Zhao, C., Wang, H., Qiu, R., Li, Y., & Mu, K. (2018). *A semantic path-based similarity measure for weighted heterogeneous information networks.* Paper presented at the International Conference on Knowledge Science, Engineering and Management.

Yang, M., Liu, J., Chen, L., Zhao, Z., Chen, X., & Shen, Y. (2019). An advanced deep generative framework for temporal link prediction in dynamic networks. *IEEE Transactions on Cybernetics*.

Yao, K., & Mak, H. F. (2014). *Pathsimext: revisiting pathsim in heterogeneous information networks.* Paper presented at the International Conference on Web-Age Information Management.

Yin, H., Benson, A. R., Leskovec, J., & Gleich, D. F. (2017). *Local higher-order graph clustering.* Paper presented at the Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining.

Yoshida, K. i., & Motoda, H. (1995). CLIP: Concept learning from inference patterns. *Artificial Intelligence, 75*(1), 63-92.

Zaki, M. J. (2002). *Efficiently mining frequent trees in a forest.* Paper presented at the Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.

Zaki, M. J. (2005). Efficiently mining frequent trees in a forest: Algorithms and applications. *Knowledge and Data Engineering, IEEE Transactions on, 17*(8), 1021-1035.

Zayani, M.-H., Gauthier, V., Slama, I., & Zeghlache, D. (2012). *Tracking topology dynamicity for link prediction in intermittently connected wireless networks.* Paper presented at the 2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC).

Zeng, Z., Wang, J., Zhou, L., & Karypis, G. (2007). Out-of-core coherent closed quasi-clique mining from large dense graph databases. *ACM Transactions on Database Systems (TODS), 32*(2), 13.

Zhang, D., Dai, M., Li, L., & Zhang, C. (2015). Distribution characteristics of weighted bipartite evolving networks. *Physica A: Statistical Mechanics and its Applications, 428*, 340-350.

Zhang, J., Jiang, Z., & Li, T. (2018). *CHIN: Classification with META-PATH in heterogeneous information networks.* Paper presented at the International Conference on Applied Informatics.

Zhang, M., Wang, J., & Wang, W. (2018). HeteRank: A general similarity measure in heterogeneous information networks by integrating multi-type relationships. *Information Sciences, 453*, 389-407.

Zhang, Y., Xiong, Y., Kong, X., Li, S., Mi, J., & Zhu, Y. (2018). *Deep collective classification in heterogeneous information networks.* Paper presented at the Proceedings of the 2018 World Wide Web Conference.

Zhang, Y., Yang, X., Wang, L., & Li, K. (2020). *WMPEClus: Clustering via Weighted Meta-Path Embedding for Heterogeneous Information Networks.* Paper presented at the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI).

Zhao, J., Xiao, D., Hu, L., & Shi, C. (2019). *Coupled Semi-supervised Clustering: Exploring Attribute Correlations in Heterogeneous Information Networks.* Paper presented at the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data.

Zhao, Z., Li, C., Zhang, X., Chiclana, F., & Viedma, E. H. (2019). An incremental method to detect communities in dynamic evolving social networks. *Knowledge-Based Systems, 163,* 404-415.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in neural information processing systems, 16*(16), 321-328.

Zhou, D., Huang, J., & Schölkopf, B. (2005). *Learning from labeled and unlabeled data on a directed graph.* Paper presented at the Proceedings of the 22nd international conference on Machine learning.

Zhou, Y., Cheng, H., & Yu, J. X. (2009). Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment*, *2*(1), 718-729.

Zhou, Y., Huang, J., Li, H., Sun, H., Peng, Y., & Xu, Y. (2018). A semantic-rich similarity measure in heterogeneous information networks. *Knowledge-Based Systems, 154*, 32-42.

Zhou, Y., Huang, J., Sun, H., Sun, Y., Qiao, S., & Wambura, S. (2019). Recurrent meta-structure for robust similarity measure in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD), 13*(6), 1-33.

Zhou, Y., & Liu, L. (2014). *Activity-edge centric multi-label classification for mining heterogeneous information networks.* Paper presented at the Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.

Zhu, L., Guo, D., Yin, J., Ver Steeg, G., & Galstyan, A. (2016). Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Transactions on Knowledge and Data Engineering, 28*(10), 2765-2777.

Zhu, T., Li, P., Yu, L., Chen, K., & Chen, Y. (2020). Change Point Detection in Dynamic Networks based on Community Identification. *IEEE Transactions on Network Science and Engineering*.