D 2020

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MASSIVE SCALE STREAMING GRAPHS: EVOLVING NETWORK ANALYSIS AND MINING

SHAZIA TABASSUM
TESE DE DOUTORAMENTO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
ENGENHARIA INFORMÁTICA

Faculdade de Engenharia da Universidade do Porto

Massive Scale Streaming Graphs: Evolving Network Analysis and Mining

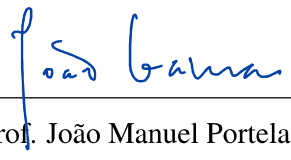
Shazia Tabassum

Dissertation
submitted to Faculdade de Engenharia da Universidade do Porto
to obtain the degree of

Doctoral Program in Informatics Engineering

Approved by:

President: Prof. Carlos Soares, University of Porto
Referee: Prof. Albert Bifet, University of Waikato
Referee: Prof. Mykola Pechenizkiy, Eindhoven University
Referee: Prof. Eduarda Mendes Rodrigues, University of Porto
Referee: Prof. João Pedro Mendes Moreira, University of Porto
Supervisor: Prof. João Manuel Portela da Gama, University of Porto



(Prof. João Manuel Portela da Gama)

May, 2020

Abstract

Social Network Analysis has become a core aspect of analyzing networks today. As statistics blended with computer science gave rise to data mining in machine learning, so is the social network analysis, which finds its roots from sociology and graphs in mathematics. In the past decades, researchers in sociology and social sciences used the data from surveys and employed graph theoretical concepts to study the patterns in the underlying networks. Nowadays, with the growth of technology following Moore's Law, we have an incredible amount of information generating per day. Most of which is a result of an interplay between individuals, entities, sensors, genes, neurons, documents, etc., or their combinations. With the emerging line of networks such as IoT, Web 2.0, Industry 4.0, smart cities and so on, the data growth is expected to be more aggressive. Analyzing and mining such rapidly generating evolving forms of networks is a real challenge. There are quite a number of research works concentrating on analytics for static and aggregated networks. Nevertheless, as the data is growing faster than computational power, those methods suffer from a number of shortcomings including constraints of space, computation and stale results. While focusing on the above challenges, this dissertation encapsulates contributions in three major perspectives: Analysis, Sampling, and Mining of streaming networks.

Stream processing is an exemplary way of treating continuously emerging temporal data. Therefore, in this dissertation, we propose algorithms that comply with single-pass and limited memory for processing. Additionally, to deal with the situations where data generation speed is higher than the processing speed, we present dynamic sampling on evolving networks. Dynamic sampling in streaming scenarios is capable of efficiently managing in-memory data for high-speed networks; This makes it a powerful means to serve many problems such as performing analytics, maintaining sufficient statistics, quantifying changes, real-time learning or running queries and applications on evolving data. However, the samples need to be representative of the structural and topological properties, changing behaviors, distributions, and patterns in the networks. Here, we present some fast and effective memoryless sampling techniques biased to recency and the strength of changing relationships in an evolving network. They are also empirically proved to be closely preserving some important properties and distributions in various evolving networks. We also exploit them with the application perspective.

Additionally, in this work, we introduce, analyze and recognize the significance of recurring links and develop a fast and scalable predictive model for recurring links in temporal network streams. Another contemporary application of network analytics is

fraud detection. Exploring the social interaction patterns of users in a network promotes the identification of different anomalous behaviors. Therefore, we exploit those patterns to identify and learn features that differentiate legitimate users from fraudsters. Eventually, we propose some novel network analysis metrics which facilitate us in quantifying and characterizing links in the above tasks.

Keywords: Graph streams. Evolving networks. Social network analysis. Sociometrics. Sampling. Forgetting. Recurring links. Link prediction. Anomaly detection. Fraud detection.

Resumo

Análise de redes sociais é um aspecto central da análise de redes. Do mesmo modo que as estatísticas combinadas com a ciência da computação deram origem ao Data Mining, o mesmo ocorre com a análise de redes sociais, que encontra suas raízes na sociologia e nos grafos. Nas últimas décadas, investigadores da àrea de sociologia e ciências sociais utilizaram dados dessa investigação e aplicaram conceitos teóricos sobre grafos para estudar os padrões nas redes subjacentes. Atualmente, com o crescimento da tecnologia e seguindo a Lei de Moore, temos uma quantidade massiva de informações geradas por dia. A maior parte é resultado de uma interação entre indivíduos, entidades, sensores, genes, neurónios, documentos, ou outras combinações. Espera-se que o crescimento de dados seja mais agressivo com a ascensão de redes como IoT, web 2.0, Indústria 4.0, cidades inteligentes etc. Analisar e extrair conhecimento destas formas de redes é um verdadeiro desafio na medida em que estão em rápida evolução. Existem vários trabalhos de investigação com foco em análises para redes estáticas e agregadas. No entanto, como os dados estão crescendo mais rápido do que o poder computacional, esses métodos sofrem de várias deficiências, incluindo restrições de espaço, computação e resultados obsoletos. Esta dissertação está centrada nos desafios acima mencionados, apresentando contribuições em três perspectivas principais: Análise, Amostragem e Extração de conhecimento de redes em *streaming*.

O processamento de fluxo de dados é um modo exemplar de tratar continuamente dados temporais emergentes. Nesta dissertação, apresentamos algoritmos de streaming que satisfazem as propriedades mais importantes desta área, ao manterem uma passagem única sobre os dados e utilizarem memória limitada durante o processamento. Além disso, para lidar com situações em que a velocidade de geração de dados é superior à velocidade de processamento, propomos amostragem dinâmica em redes em evolução. A amostragem dinâmica em cenários de streaming é capaz de gerir com eficiência dados em memória para análises de grafos, manter estatísticas suficientes, quantificar alterações, aprender em tempo real ou executar consultas e aplicativos, etc. No entanto, as amostras precisam de ser representativas das propriedades estruturais e topológicas, comportamentos dinâmicos, distribuições e padrões nas redes. Apresentamos algumas técnicas de amostragem rápidas, eficazes e *memoryless*, influenciadas pela nova informação e pela força da mudança de relacionamentos em uma rede em evolução. Estas técnicas foram comprovadas empiricamente, preservando algumas propriedades e distribuições importantes em várias redes em evolução, que também são exploradas com a perspectiva de aplicação.

Adicionalmente, neste trabalho, apresentamos, analisamos e reconhecemos a importância dos links recorrentes. Neste caso desenvolvemos um modelo preditivo rápido e escalável para links recorrentes em fluxos de redes temporais. Uma outra aplicação contemporânea da análise de rede é a detecção de fraudes. Explorar e aprender os padrões de interação social dos utilizadores em uma rede, promove a identificação de diferentes comportamentos anómalos. Portanto, exploramos essas características para descobrir, interpretar e diferenciar os padrões estruturais de utilizadores legítimos de fraudulentos. Por fim, propusemos novas métricas de análise de rede que nos facilitaram a quantificação e caracterização de links nas tarefas anteriormente mencionadas.

Keywords: fluxo de grafos. redes dinâmicas. análise de redes sociais. amostragem. esquecimento. links recurrentes. predição de link. deteção de anómalias. Deteção de fraudes

Funding Acknowledgements

The initial works and experiments referenced in the proposal are financed by the European Commission through MAESTRA (ICT-2013-612944) from May 2015 to Jan 2017. Later works are carried out for the Project TEC4Growth-RL SMILES-Smart, Mobile, Intelligent and Large scale Sensing and analytics NORTE-01-0145-FEDER-000020 which is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement and ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project (POCI-01-0145-FEDER-006961). During this research, I was also part of Indo-Portugal Bilateral Scientific and Technological Cooperation project INT/PORTUGAL/P-15/2017.

Acknowledgements

All these years of my PhD have a significant impact on my life. I have had new experiences, new friends, a different work environment and visited many places. It helped me grow as a researcher and as a human being. I want to thank everyone whom I met in this journey, as everything would not have been so pleasant without the people around me.

Most importantly, I would like to express my wholehearted gratitude to Prof. João Gama, who laid the foundation for this Ph.D. thesis and provided me with valuable opportunities. I always admired his optimistic attitude, social skills and expertise, which made me learn a lot from him. His approach to complex things in a simple and effortless way is absolutely commendable and I also tried to follow it in this thesis. He always gave me constant encouragement with his words, "go ahead" for all the potential ideas I came up with. He often referred to a Portuguese saying, "the only way you can learn to do things is by doing," which I found very inspiring in learning things during this work.

I would also like to extend my sincere gratitude to Prof. Augusto Sousa and Prof. Eugenio Oliveira, who generously welcomed and introduced me to research. I want to thank the exceptional faculty Prof. Joao Moreira, Prof. Carlos Soares, Prof. Rui Camacho, for being my teachers, sharing their knowledge and their availability to help. Thanks to Prof. Alipio Jorge, Prof. Pedro Campos, Prof. Carlos Ferreira, Prof. Rita Ribeiro, Prof. Pedro Rebeiro for their time, questions and guidance. A special thanks to Joana Dumas, LIAAD secretary and Sandra, ProDEI secretary, for being helpful with the administrative details. I want to thank Prof. Ronita Bardhan for collaborating and inviting me to IIT-Bombay associated with the Indo-Portugal program as a Ph.D. student.

Many thanks to Fabiola Perreira for having long discussions of work, shaping ideas and working on them together and besides that being my dearest friend. Special thanks to Bruno Veloso for helping and collaborating with me in this thesis work. Kudos to him for being an efficient researcher. A big thanks to all my friends and colleagues at LIAAD in alphabetical order, Ahmed, Conceicao, Eduarda, Diogo, Joao, Kerley, Maria, Mariana, Rita, Renata, Patricia, Paulo, Salisu, Sonia, Sofia, Susane, Yaseen. A sincere thanks to Jianpeng Zhang for visiting and sharing his work. Besides being a good researcher, he is a Kind and noble person and I did learn a lot from his work. A special thanks to Maria, Bruno and Sonia for spending time proofreading my draft and translating the abstract to Portuguese and Rita for helping me meet the deadline. A hearty thanks to all my family friends whose care, support, cordial chats, gathering, amazing food and travels made me feel no stress during my Ph.D.

Finally yet importantly, a big thanks to my Mum, who dreamed about my future right from the beginning of my life and strived hard to make me see this day; her teachings and the values instilled in me. To my Dad, whose hardships and efforts are the driving source

of all the knowledge I have acquired. To my husband, who is the one who wanted me to do this. His support, encouragement, help and care made me successfully complete this work. To my brother and sister, who had been a constant source of my emotional strength. Last but the most significant, to my son, whose presence makes me feel a stress-free and happiest person in the world. Surely this work is yours my son, for all your patience to let me work even when you accompanied me to the lab.

My heartfelt gratitude to all the people above and more, who had directly or indirectly helped me accomplish this important goal in my life. While writing this when the world is stuck with the COVID-19 pandemic, I hope we overcome this successfully and come out strong.

Contents

List of Figures

List of Tables

List of Abbreviations

ACS	Average Component Size
AD	Average Degree Centrality
AWDC	Average Weighted Degree Centrality
CDR	Call Detail Record
EVC	Eigen Vector Centrality
PCA	Principal Component Analysis
RLP	Recurring Link Prediction
RS	Reservoir Sampling
SBias	Sampling Evolving Networks with Bias
SEFF	Sampling Evolving Ego Networks with Forgetting Factor
SNA	Social Network Analysis
Socio-Metrics	Social Network Analysis Measures
SRS	Simple Reservoir Sampling
SS	Space Saving Algorithm
SSE	Space Saving using Edges
SSN	Space Saving using Nodes
TN	Telecommunication Networks

Part I

Prologue

Chapter 1

Introduction

When objects in the data are associated with relationships between themselves, the complexity of data increases and so is the information gained from it. This complexity augments when these objects and relationships change or evolve over time. Large volumes of data are being generated from real-time application domains such as web, social media, email, medical, sensors, telecommunications, etc., which have been modelled as networks. The study of such complex relationship networks, is referred to as network science (Girvan and Newman, 2002), which can yield insight into their structures, properties, and emergent behaviors (Aggarwal, 2011).

Networks can be represented mathematically by using graphs. Graphs are used to model pairwise relations between objects (He and Petoukhov, 2011). In real-world scenarios, relations or links play an important role in diffusion, transmission and propagation of information/diseases/trust etc. Therefore, graph data structures can efficiently provide an abstraction of interactions or relationships between different entities of the real-world applications in question. Thus, analyzing and mining patterns from integrated abstract social objects leverages useful and actionable insights into the real-world physical and social entities. This helps users as well as service providers, make informed decisions, decrease operational costs and increase service efficiency. Focusing on the current advancements and applications in this domain, Gartner predicts that the application of graph processing and graph databases will grow at 100% annually through 2022 to continuously accelerate data preparation and enable more complex and adaptive data science (Gar, 2019).

1.1 Problem Overview

The work in this thesis is carried out from the evolving perspective of graphs. Most of the networks in real-world are evolutionary in nature, arising from the continuous exe-

cution of activities by the objects in the network. The activities could be transactions, interactions, posts, messages, signals, collaborations, calls etc. Resulting in more than millions of nodes and billions of edges in some cases per second. Data stream processing is regarded as a speed focused approach to handle such continuous unbounded data for providing real time insights and space efficient computations. Moreover, some applications need prompt responses, that if delayed, their value diminishes. It is estimated that the global data sphere will grow from 33 zettabytes in 2018 to 175 by 2025 and almost 30% of it will require real-time processing (Sea, 2018). A typical architecture of real-time big data analytics is shown in Figure 1.1. Additionally, stream processing can also befit for offline analysis where all the data/graph does not fit in memory even considering batches.

However, stream processing is associated with a set of challenges. Incremental techniques are cumulative and can quickly run out of space in evolving and unbounded scenarios where new nodes and edges are being added continuously. Therefore, it is imperative to find solutions that are incremental and decremental at the same time. We presented such techniques in this thesis for evolution analytics using sampling. Typically sliding windows are used for this purpose, but they need to keep track of indexing or ordering. Therefore, we provided dynamic memoryless sampling approaches for data or network streams. Data sampling techniques in statistics have been fundamental in estimating the properties of population distribution. However, they did not consider the structure and dependencies in a network, their evolution, temporal order and streaming perspective. Although there exist quite a few techniques for sampling networks, most of them did not address dynamic sampling in streaming scenarios. Their approaches and limitations are detailed in chapters 4 and 5. Eventually, while addressing the above-mentioned challenges, we also specifically introduced sampling algorithms for the networks which generate recurring links over time.

Analytical network sampling makes use of social network analysis concepts to study the samples for estimating and modeling true network structure. Besides, it can be exercised for other applications such as answering queries, simulations, analyzing trends, recommendations, detecting changes and anomalies, etc. Of particular interest here was to evaluate sampling techniques on the basis of the structural properties they preserve from the true networks. Consequently, identifying the techniques that generate representative samples, which can be further exploited for applications or extrapolations.

Application areas such as mobility, sensors, communications, purchases, recommendations, etc., are composed of links that recur in a period of time. Predicting such links as quickly as possible is a pressing need in current scenarios. In this thesis, we present a fast and memoryless sampling-based model for predicting such recurring links. The eval-

uations have been carried out on a diverse set of networks revealing significant gain over competent algorithms. The systematic methodology framework of the process proposed and followed in this dissertation is given in Figure 1.2.

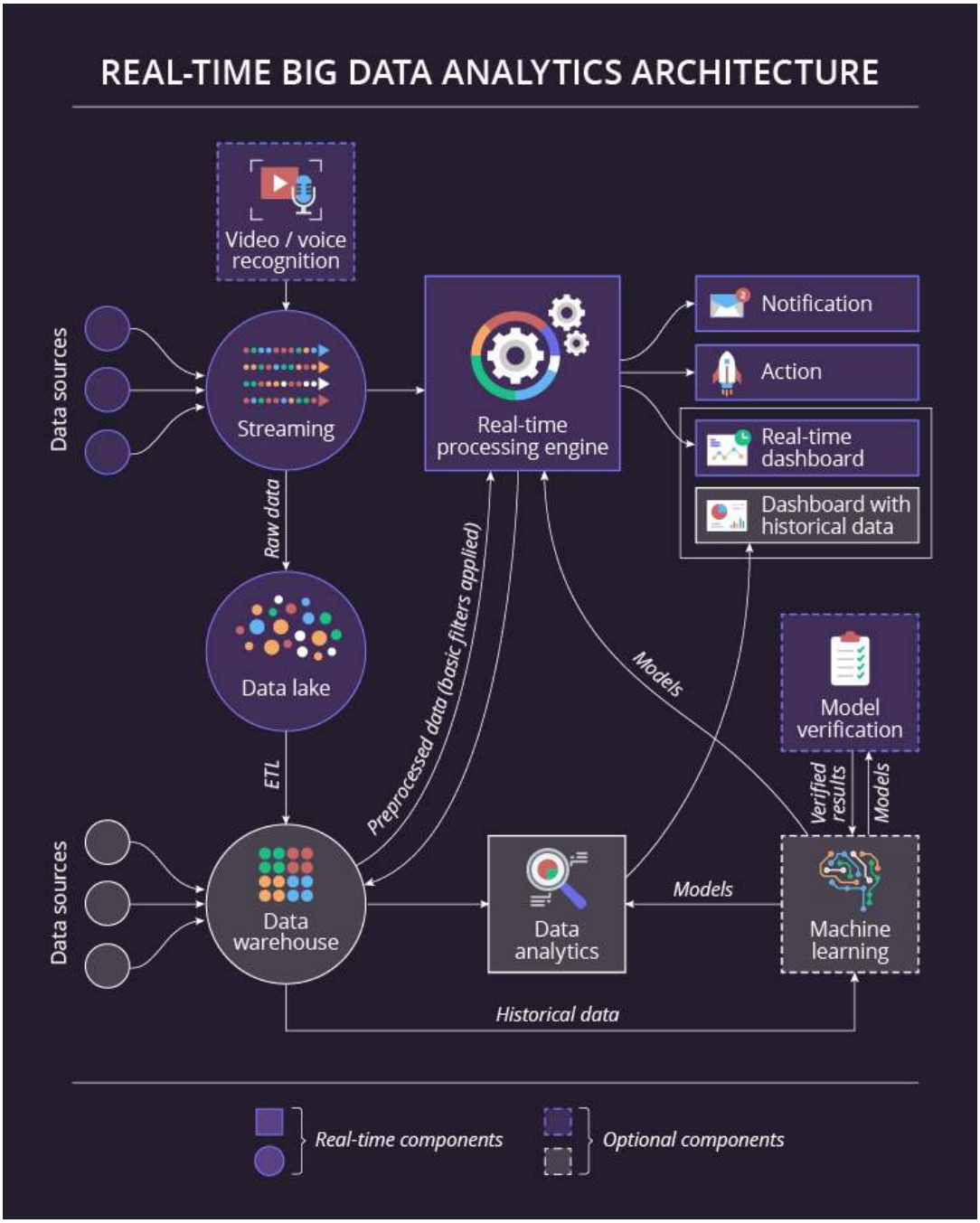


Figure 1.1: A typical real-time bigdata analytics architechture excerpt from Sci (2018)

Some typical applications of graph analytics also include finding friends, identifying groups and influential nodes, targeted marketing, finding frauds etc. As an example, Ora-

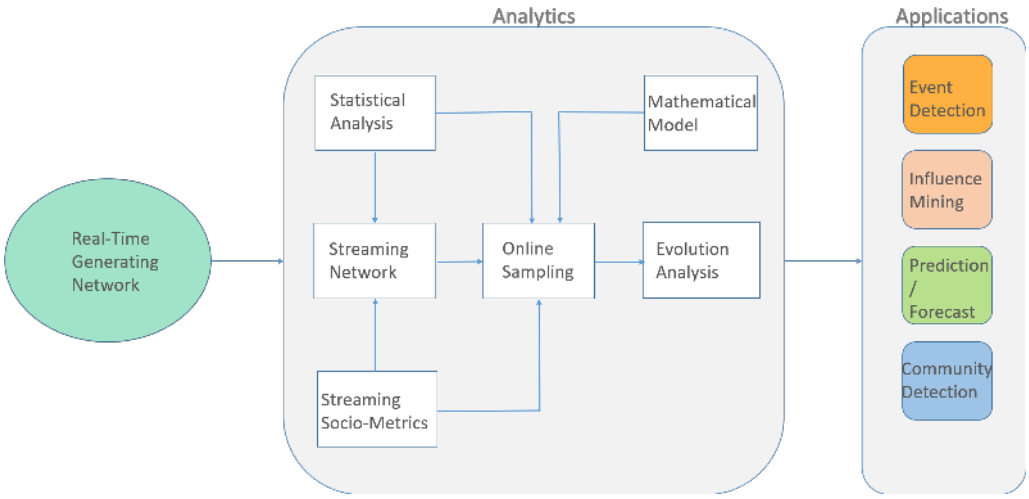


Figure 1.2: Systematic Methodology Framework

cle (Ora, 2018) uses graph analytics for fraud detection, where they identify bot accounts that retweet certain target accounts to make them look popular. However, different frauds have distinct characteristics that can be identified by the features that characterize them. As part of this dissertation, we also explore fraud in terms of telephony abuse by leveraging social network analytics. As a result, we presented a model for identifying and profiling fraudulent users from a phone call network.

1.2 Research Questions

With the real-time content generating and data collecting applications, there is an ever-increasing demand for real-time handling and processing of data. The classical methods usually treat data in an aggregate fashion, which leads to major issues like:

1. *Scale of data:* The size of data is too large to fit in the conventional main memories and I/O operations on secondary storage would increase computational complexity.

2. *Obsolete results:* Most of the applications need latest information to predict outcomes. For example, recommendations, social media, traffic, sensors, frauds etc. The results based on very old information are impractical in transient or time-ordered scenarios.

3. *Bounded size:* Static methods assume data to be bounded size and fixed distributions. Thus lacking incremental techniques and always having to start from scratch. Consequently, hindering them from providing prompt online results and difficulty in tracking change or evolution.

The above issues are being addressed in the field of data stream processing for real-time big data analytics. Moreover, distributed stream processing platforms are an added advantage for increasing latency and throughput (Nasiri et al., 2019). But what about networked data with the above bigdata-focused challenges? Classical data stream techniques disregard the cross-linking between data items, which is one of the main sources of information in networks. Distributed computing cannot be straightforwardly applied to it with the overhead of connections. The data needs to be virtually segmented before injecting into distributed engines or as an internal process to reduce communication overhead between different processors or servers. This could be accomplished by exploiting ego-networks, communities, random walks, sampling etc. Which calls for another real-time component in the processing framework. Therefore, considering the above issues, the main purpose of this thesis is to

devise simple, fast and efficient methods and algorithms for analyzing and mining massive evolving networks without having to store all the data over time.

In order to achieve this, we addressed the following research questions in this thesis.

RQ1: Evolving Graph Analysis: How do you process streaming graphs and compute incremental and decremental socio-metrics from a temporally evolving network?
To address this question, in Chapter 3, we outlined the fundamental methods and approaches for maintaining updates on socio-metrics as the stream progresses. Moreover, we used these metrics throughout the thesis in all chapters in an incremental fashion and also for evaluation purposes. We also discussed in Chapter 3 an approach for processing streaming networks.

RQ2: Dynamic Sampling: How to deal with the volume of complex graphs which are evolving at high speeds in real-time?
Fast Dynamic Sampling is the term that incorporates the three issues mentioned in this question. In this thesis, we presented algorithms in Chapter 4, which are fast to deal with high-speed network data, dynamic to cope up with the evolving nature and constant size to handle the volume. Processing data, as it streams, is a well known approach for instantaneous or timely results. This has been done sequentially considering network stream, as explained in Chapter 3. The proposed sampling method in 4 is inherently biased to the latest data.

RQ3: Sampling Recurring Links: In case of networks with recurring links, how do you update evolving samples with bounded space and time complexities while preserving network structure?
To address this problem, we applied a memoryless exponential forgetting technique that introduces bias on the network based on time and recurrence to dynamically sample edges in Chapter 5. The generating samples have been evaluated in comparison with the true network and other sampling methods. The results show the samples to be closest to the real networks than other sampling methods and possess other important structural properties that have been exploited in 7. Besides that, it preserves better cluster and component structure with least bias to low degree nodes by decreasing redundant edges and drawing samples close to spanning tree.

RQ4: Network Stream Mining: How do you leverage the above dynamic stream sampling computational models for learning and predicting recurring links over time?
From our analysis of recurring link networks, we found that recurrence probability of links decreases exponentially with the distance in time. Therefore, in Chapter 7, we applied the above memoryless sampling technique to predict links based on their strength, stability and time of occurrence. The applied approach is not only fast and scalable but also exhibits better prediction efficiency over previous recurring link prediction and sampling methods.

RQ5: Social Network Analytics: Which social network features are helpful in defining the variability of data in a call network for detecting fraudulent users?
Fraudulent users manifest patterns of interactions that are not common with the other users in the network. These patterns of interactions influence social network properties. Therefore, we use such properties as features for detecting frauds in telephony. The importance of contributing variables is shown in Chapter 8. The proposed recurring links property is proved to be more significant than call duration, which is the extensively used variable in fraud detection.

RQ6: Anomaly Detection: How do you find nodes with anomalous behavior from a telecommunication network without having to find clusters in all the data?
Clustering large data as a whole has increased computational complexity along with the number of clusters and variables. Therefore, we apply unsupervised learning methods on the extreme outliers in Chapter 8. Since finding anomalous users with standard techniques such as box plots results in more than 50% of the nodes as outliers in real-world networks following power law distributions. Hence, it is crucial to detect high leverage points

instead. These points are though outliers can have different profiles. By clustering these points, we found user profiles that match fraudulent users.

1.3 Main Contributions

In this section, we present the main contributions of this dissertation accomplished in the research area of very large and high-speed data networks that are evolving over time. We also reference below the research papers published in peer-reviewed journals and conferences, which are associated with the respective contributions.

- We compiled a succinct review from the contemporary methods used in graph analytics, which provides a guided pathway from analysis to mining of networks.
- We also presented a streaming perspective of processing and analyzing evolving networks over time.

Tabassum, S., Pereira, F. S., Fernandes, S., & Gama, J. (2018). Social network analysis: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(5), e1256.

Tabassum, S., Pereira, F. S., Fernandes, S., & Gama, J. (2018). Cover Image, Volume 8, Issue 5. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(5), e1281.

- We proposed and implemented bounded size dynamic sampling algorithms that are capable of handling high-speed network streams such as phone calls.
- We explored the statistical and structural biases of samples, also with the contribution of visualization techniques to interpret them.

Tabassum, S., & Gama, J. (2016, April). Sampling massive streaming call graphs. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 923-928). ACM.

Tabassum, S. (2016, June). Social network analysis of mobile streaming networks. In 2016 Workshop on High Velocity Mobile Data Mining in 17th IEEE International Conference on Mobile Data Management (MDM) (Vol. 2, pp. 20-25). IEEE.

- We proposed another novel sampling methodology called 'SBias', with a special focus on recurring and weighted temporal links. The edges that maximize a memoryless forgetting function remain in the evolving sample. Eventually, requiring constant update time per edge.

- Besides the properties considered so far in the literature, we presented some interesting ones that the online samples need to preserve, such as efficiency and temporal distributions.

- SBias represents highly correlated samples to the true network in comparison with the state of the art methods on diverse network domains.

Tabassum, S., & Gama, J. (2018, December). Biased dynamic sampling for temporal network streams. In International Conference on Complex Networks and their Applications (pp. 512-523). Springer, Cham.

Tabassum, S., Pereira, F. S., & Gama, J. (2018). Knowledge Discovery from Temporal Social Networks. Intelligent Informatics, 10.

- In addition to global networks, we concentrated on local network properties such as ego networks and discovered how they evolve in a telecom network.

- As we see the ego networks densify similarly as the global network we proposed a forgetting function for ego networks' dynamic sampling and observe their representatives with ground truth.

Tabassum, S., & Gama, J. (2016, October). Evolution analysis of call ego-networks. In International conference on discovery science (pp. 213-225). Springer, Cham.

Tabassum, S., & Gama, J. (2016, June). Sampling evolving ego-networks with forgetting factor. In 2016 Workshop on High Velocity Mobile Data Mining in 17th IEEE international conference on mobile data management (MDM) (Vol. 2, pp. 55-59). IEEE.

- We formalized the problem of recurring links in a stream setting and analyzed the importance of recurring links and their recurrence probability over time.

- We designed a fast and scalable heuristics-based streaming model for predicting recurring links in evolving networks with higher predictive efficiency over baselines, in real-world massive data sets.

Tabassum, S., Veloso, B., & Gama, J. (2020). On fast and scalable recurring link's prediction in evolving multi-graph streams. Network Science. Cambridge University Press.

- We proposed a novel approach for detecting fraudulent users in telecommunication networks using unsupervised learning from high leverage points.

- We also observed that the outliers detected using Mahalanobis distance are not all anomalies in the same sense (have dissimilar profiles), especially when the variables are inversely correlated.
- We identified and extracted the graph-based structural properties of nodes in the network whose values are uncommon to the common users and evaluated them based on their contribution in characterizing user behaviors.

Tabassum, S., Ajmal, M., & Gama, J. Profiling High Leverage Users for Fraud Detection in Telecom Data Networks. Manuscript submitted for a journal review.

1.4 Dissertation Organization

After providing the preliminaries in this unit, we here discuss the structure and organization in the rest of the dissertation. This dissertation is divided into four parts. The first part **Prolog:**, which introduces the problem and highlights contributions. The last part **Epilogue:** concludes the thesis. Based on the focus and specificity of individual chapters, the body of the thesis is divided into two main parts, i.e., **part II and part III**. These two parts are major components in our methodological framework. Part II: emphasizes the analytical methods for static and evolving networks. Part III: exploits the analytics from the perspective of applications. We compile them using scientific articles that are already published in peer-reviewed international scientific journals and conferences.

Part-II In this part, we focus on the problem of analyzing very large streaming networks using incremental and decremental methods. Consequently, we study fast dynamic sampling methodologies for networks.

Chapter 2 and 3 In these chapters, we provide an overview of basic methods, approaches and techniques for static and evolving graph analytics. These chapters set a background for the following chapters in this thesis. Firstly, we present the way we process streaming graphs for the purpose of the methods introduced in this thesis. Some of the fundamental socio-metrics need simple updates and low computational complexity to make them incremental, which is also elucidated in this chapter. Additionally, continuously streaming graphs (which cannot be accommodated by memory in their entirety) also require decremental methods with one pass constraint. Besides discussing the background for these methods, we also present some streaming data sampling methods and illustrate their basic differences. Particularly, state of the art for graph sampling methods

and their limitations are discussed in the respective chapters 4 and 5. Lastly, we explain the importance of visualization in this domain and also refer to the tools employed in this work.

Chapter 4 This chapter introduces algorithms for sampling high-speed network streams. While reviewing the limitations in the literature for sampling streams, we present a novel conceptually simple algorithm. It not only preserves structures better than a classic algorithm but maintains the latest information without storing ordering information. The evaluations of samples are based on their ability to preserve structural properties.

Chapter 5 As there are no sampling methods proposed for dynamically recurring links' networks so far, we introduce one in this chapter. For this, we exploit the weight of links based on their recurrence, which symbolizes the strength of relationship between nodes. The samples were evaluated based on their structural and statistical proximity to the true network in the data sets from diverse domains.

Chapter 6 In this chapter, we perform an analysis of ego networks over time. Firstly, we introduce the concept of ego networks and related works in the area. Then we find that they also satisfy densification power law and propose a dynamic sampling strategy focusing on ego networks with recurring links. Finally, we notice some important properties achieved by the sample, such as decreasing redundant edges.

Part-III This part encompasses two chapters on the applications and models for graph mining on evolving network streams.

Chapter 7 The sampling methodologies introduced in part-I are further explored and adapted for recurring link's prediction in evolving network streams. This leads to fast and scalable algorithms for the above task. Firstly, we present the motivation, background, and limitations of the literature in predicting links. We also carry out an analysis to understand the significance of recurring links and their temporal relation while introducing new metrics. Lastly, we present the predictive performance of methods using prequential analysis.

Chapter 8 The overview of the problem is followed by recognizing the characteristics of telephony abuse in telecommunication networks. The graph analytics studied in part-I are extracted as features from the network over time in this chapter. These features are leveraged to detect extreme anomalous nodes for fraud detection. We also present

some novel socio-metrics which define the variability of users for specific frauds in this scenario. Additionally, we use social network visualization to understand patterns of interactions of legitimate users and fraudulent users.

Part-IV Finally, the last part concludes this thesis. This part summarizes the thesis, the data sets used for experiments and the source code availability. Lastly, we reference our scientific contributions in the field during the work of this dissertation.

Part II

Evolving Network Analytics

Chapter 2

Networked Data Analytics : An Overview

Social Network Analysis (SNA) is the core aspect of analyzing data networks today. In addition to the usual statistical techniques for data analysis, the networks are investigated using SNA measures. It helps in understanding the dependencies between social entities in the data, characterizing their behaviors and their effect on the network as a whole and over time. Therefore, this chapter attempts to provide a succinct overview of SNA in diverse topological networks (static, temporal and evolving networks) and perspective (ego-networks). As one of the primary applicability of SNA is in networked data mining, we provide a brief overview of network mining models as well; by this, we try to present the readers with a concise guided tour from analysis to mining of networks.

2.1 Introduction

Nowadays the data generated from many of the real-world applications are represented as a network of interconnected objects. The main objective is to extract more information than the traditional way of investigating independent objects. Of course, it increases the complexity of handling data as well. One of the major class of data networks is social networks. A social network can be constructed from relational data and can be defined as a set of social entities, such as people, groups, and organizations, with some relationships or interactions between them. These networks are usually modeled by **graphs**, where vertices represent the social entities and edges represent the ties established between them.

Some of the common examples of social networks are given in Table 2.1. The underlying structure of such networks is the object of study of Social Network Analysis (SNA). SNA methods and techniques were thus designed to discover patterns of interaction be-

tween social actors in social networks. Hence, the focus of SNA is on the relationships established between social entities rather than the social entities themselves. In fact, the main goal of this technique is to examine both the contents and patterns of relationships in social networks in order to understand the relations among actors and the implications of these relationships.

Table 2.1: Some examples of data networks

Examples	**Applications**
Friendship networks	College/school students, Organizations or Web(Facebook, MySpace, etc.)
Follower networks	Twitter, LinkedIn, Pinterest, etc.
Preference similarity networks	Pinterest, Instagram, Twitter, etc.
Interaction networks	Phone calls, Messages, Emails, Whatsapp, Snapchat, etc.
Co-authorship networks	Dblp, Science direct, Wikibooks, other scientific databases, etc.
Spread networks	Epidemics, Information, Rumors, etc.
Co-actor networks	IMDB, etc.

A network is defined by the relation/link between the nodes in it, as given in the examples above. There can be distinct relations between a single set of nodes in a network. For example, in a product network, the relation could be based on "similarity" or "brought together" in a set of products. Similarly, there can be unique/distinct relations between multiple sets of nodes, for example user-product networks. These type of networks are heterogeneous networks. When the network comprises of two sets of nodes, it is called a two-mode network. Some examples of two-mode networks include user-product (Amazon, eBay etc), membership or affiliation (actor-movies (IMDB), user-group (youtube), user-channel (youtube), user-project (GitHub), user-organization etc), user-preference (Pinterest, Instagram, twitter) and user-citation networks. These two-mode networks can be transformed into single-mode networks between a single set of nodes like the examples given above and then analyzed. However, two-mode networks can also be analyzed using methods discussed by Borgatti and Everett (1997) and Latapy et al. (2008).

Apart from social networks, numerous data networks are also formed between objects other than social entities, like sensors, products, words/texts, brain neurons, proteins, genes, geographical locations, predators and preys, and web-pages etc. Though the SNA measures were primarily designed to analyze social networks, they can also be employed to analyze data networks like these.

Common tasks of SNA include the identification of the most influential, prestigious or central actors, using statistical measures; the identification of hubs and authorities, using link analysis algorithms; discovering communities, using community detection techniques; and how information propagates in the network, using diffusion algorithms. These tasks are extremely useful in the process of extracting knowledge from networks and, consequently, in the process of problem-solving. Due to the appealing nature of such tasks and to the high potential opened by this kind of analyses, SNA has become a popular approach in a myriad of fields, from Biology to Business. For instance, some companies use SNA in order to maximize positive word-of-mouth of their products by targeting the customers with higher network value (those with higher influence and support) (Domingos and Richardson, 2001; Richardson and Domingos, 2002; Leskovec et al., 2007a). Other companies, such as the ones operating in the sector of mobile telecommunications, apply SNA techniques to the phone call networks and use them to identify customer's profiles and to recommend personalized mobile phone tariffs, according to these profiles. These companies also use SNA for churn prediction, i.e. to detect customers who may potentially switch to another mobile operator by detecting changes in the patterns of phone contacts (Dasgupta et al., 2008; Wei and Chiu, 2002). Another interesting application emerges from the domain of Fraud Detection. For instance, SNA can be applied to networks of organizational communications (e.g. Enron company dataset) in order to perform an analysis of the frequency and direction of formal/informal email communication, which can reveal communication patterns among employees and managers. These patterns can help identify people engaged in fraudulent activities, thus promoting the adoption of more efficient forms of acting towards the eradication of crime (Xu and Chen, 2005; Shetty and Adibi, 2004).

Despite the fact the origins of network studies go back a few centuries ago, in recent years we witnessed an impressive advance in network-related fields, mainly due to the growing interest in social networks (Wasserman and Faust, 1994; Aggarwal, 2009; Abraham et al., 2009; Aggarwal, 2011; Furht, 2010; Zafarani et al., 2014; Barabási, 2016a), which became a "hot" topic and a focus of considerable attention. For this reason, a lot of students, practitioners, and researchers are willing to enter the field and explore, even superficially, the potential of SNA techniques for the study of their problems. Bearing this in mind, in this chapter our aim is to provide a general and succinct overview of the essentials of SNA for those interested in knowing more about the area and strongly oriented to use SNA in practical problems and different classes of networks.

The remainder of this Chapter is organized as follows. We begin by pointing out some types of representations that can be used to model social networks. Then, we introduce the best known statistical measures to analyze them with a different perspective of net-

works: the entire social network and ego-network. Afterward, we talk a little about the Link Analysis task and explain how it can be used to identify influential and authoritative nodes. Then, we discuss the task of link prediction referring to the state of the art techniques. Later, we devote a section to the problem of finding communities in networks. It is then followed by a section focussed on anomaly detection in data networks. Finally, this overview ends with the identification of the current trends arising in the field of SNA.

2.1.1 Representation of Graphs

A social network consists of a finite set of vertices and the relations, or ties, defined on them (Wasserman and Faust, 1994). The established relationships can be of personal, or professional, nature and can range from casual acquaintance to close familiar bonds. Besides social relations, links can also represent the flow of information/goods/money, interactions, similarities, among others. The structure of such networks is usually represented by graphs. Therefore, networks are often regarded as equivalent to graphs.

A graph is composed of two fundamental units: vertices and edges. Every edge is defined by a pair of vertices, also called its endpoints. Vertices are able to represent a wide variety of individual entities (e.g. people, organizations, countries, papers, products, plants, and animals) according to the application field. In turn, an edge is a line that connects two vertices and, analogously, it can represent numerous kinds of relationships between individual entities (e.g. communication, cooperation, friendship, kinship, acquaintances, and trade). Edges may be directed or undirected, depending if the nature of the relation is asymmetric or symmetric. Formally, a graph G consists of a nonempty set $V(G)$ of vertices and a set $E(G)$ of edges, being defined as $G = (V(G), E(G))$. The order of a graph G is given by the total number of vertices n or, mathematically, $|V(G)| = n$. Analogously, the size of a graph G is the total number of edges $|E(G)| = m$. The maximum number of edges in a graph is $m_{max} = \frac{n(n-1)}{2}$, for undirected graphs, and $m_{max} = n(n-1)$, for directed ones.

In the literature, two main types of graph-theoretic data structures are referred to represent graphs: the first one is list structures and the second is matrix structures. These structures are appropriate to store graphs in computers in order to further analyze them using automatic tools. List structures, such as incidence lists and adjacency lists, are suitable for storing sparse graphs since they reduce the required storage space. On the other hand, matrix structures such as incidence matrices, adjacency matrices or sociomatrices, Laplacian matrices (contains both adjacency and degree information) and distance matrices (identical to the adjacency matrices with the difference that the entries of the matrix are the lengths of the shortest paths between pairs of vertices) are appropriate to represent

full matrices. Several types of graphs can be used to model different kinds of social networks. For instance, graphs can be classified according to the direction of their links. This leads us to the differentiation between undirected and directed graphs. Undirected graphs (or undirected networks) are graphs whose edges connect unordered pairs of vertices or, in other words, each edge of the graph connects concomitantly two vertices. A stricter type of graph is the so-called directed graph (or directed network). Directed graphs, or in the abbreviation form digraphs, can be straightforwardly defined as graphs whose all edges have an orientation assigned (also called arcs), so the order of the vertices they link matters. Formally, in a directed graph if E_{12} is an arc and v_1 and v_2 are vertices such that $e_{12} = (v_1, v_2)$, then e_{12} is said to join v_1 to v_2, being the first vertex v_1 called initial vertex, or tail, and the second vertex v_2 called the terminal vertex, or simply head. Graphically, directed edges are depicted by arrows, indicating the direction of the linkage. This type of graphs can be either cyclic, i.e. graphs containing closed loops of edges or "ring" structures, or acyclic (e.g. trees). A typical example of an undirected graph is Facebook^{TM} since, in this social network, the established friendship tie is mutual, or reciprocal (e.g. if I accept a friend request from a given person then it is implicitly assumed that I and that person are friends of each other). Likewise, Twitter^{TM} is an example of a directed graph since a person can be followed by others without necessarily following them. In this case, the tie between a pair of individuals is directed, with the tail being the follower and the head being the followed, meaning that a one-way relationship is established.

Regarding the values assigned to edges, we can make a distinction between unweighted and weighted graphs. Unless it is explicitly said, we always assume that graphs are unweighted. Unweighted graphs are binary since edges are either present or absent. On the other hand, weighted graphs are richer graphs since each edge has associated a weight $w \in \mathbb{R}_0^+$ providing the user with more information about, for instance, the strength of the connection of the pair of vertices it joins. According to Granovetter (1973, 1995) in social networks the weight of a tie is generally a function of duration, emotional intensity, the frequency of interaction, intimacy, and exchange of services. Therefore, strong ties usually represent close friends, and weak ties represent acquaintances. In other kinds of networks, the weight of a tie can represent a variety of things, depending on the context; for instance, a tie can represent the number of seats among airports, the number of exchanged products, etc. For undirected and unweighted graphs, adjacency matrices are binary (as a consequence of being unweighted) and symmetric (as a consequence of being undirected, meaning that $a_{ij} = a_{ji}$), with $a_{ij} = 1$ representing the presence of an edge between vertices i and j, and $a_{ij} = 0$ representing the absence of an edge between vertex pair (i, j). For directed and weighted graphs the entries of such matrices take values from interval $[0, max(w)]$ and are non-symmetric. In both cases, we deal with non-negative

matrices.

2.2 Topological Properties of Graphs

Data networks exhibit some graph topologies given below:

Multi-graphs: Graphs where multiple number of edges can exists between any two nodes are said to be multi-graphs.

Recurring Links: Multi-graphs are of two kinds, one is where exists different kinds of relations between two nodes at the same time. Example a relation between two products in an e-commerce network can be similarity or brought together etc. These are also called heterogeneous networks. The other kind is where the same connection occurs multiple times temporally. For example, a phone call between two users. We call them as recurring links over time.

Unique Edge When there exists multiple links between two nodes with the same relation, they are considered one of a kind and unique. The unique edge is supposed to be one edge and can contain a weight representing those multiple edges, which is the case in this work.

Scale-Free: Networks with degree distributions that follow a power law at-least asymptotically are known as scale-free.

Directed Graph: Directed graphs are also known as digraphs, where all the edges are directed. That is, the edges are defined by an ordered pair of vertices. The edge from a->b is not the same as b->a. They are also expressed as incoming and outgoing links from the perspective of a node.

Weighted Graphs: These are the graphs where the edges are associated with weights, which can represent their strength. The weights can be acquired from the network itself example frequency of an edge or it can be gained from an attribute.

Irregular Graphs: Where the vertices in a graph have different number of neighbours.

Connected Graphs: The graphs where all the nodes in the network are connected directly (complete graph) or indirectly (where a path exists between any two nodes) otherwise they are disconnected graphs.

Heterogeneous Graphs: Graphs with nodes and edges of different types. The graphs with two types of nodes are Bipartite.

Dense Graphs: A maximum dense graph is the one which has all the nodes directly connected to each and every other node in the network. If it is a directed graph then the nodes should be connected in both the directions. It typically lies between 0 and 1 inclusive, with 1 indicating maximum density.

Sparse Graph: Number of edges lesser than the maximum possible edges between the nodes in a graph defines its sparsity. Networks from real-world scenarios are usually known to have a sparsity ≈ 1. Where $Sparsity = 1 - Density$.

2.3 Analyzing Data Networks

Besides the above topological properties, the Graph Theory and Network Science incorporates some metrics to measure, evaluate and compare graphs. They are also referred as socio-metrics in this dissertation. As we apply those metrics (direct or to understand another metric) in the rest of this thesis we provide here a brief introduction to them. They are basically divided into two kinds, node level and network level metrics. Below we present some node level metrics followed by the network level measures. The node level metrics manifest a micro view of the network about the nodes in focus. They can be used to quantify the importance, influence, dependencies or relevance of a node to the other nodes in the network.

2.3.1 Node-level Metrics

In this subsection, we present some graph measures and popular metrics used in the analysis of social networks. These measures are useful in the sense that they provide us insights about the role of nodes in the network. Studying the role of individuals and how they interact in the network context aims at understanding the behavior of the social systems that generated those networks, which is normally the final goal of such analysis. The measures we will introduce in the following subsections can be divided according to the level of analysis one wants to perform: at the level of small units, such as nodes, or at

the level of the whole network. The former explores general measures of centrality as a way to understand how the position of a vertex is within the overall structure of the graph and, therefore, helps identify the key players in the network. The latter provides more compact information and allows the assessment of the overall structure of the network, giving insights about important properties of the underlying social phenomena.

Centrality or prestige is a general measure of how the position of an actor is within the overall structure of the social network and can be computed resorting to several metrics. The most widely used are degree, betweenness, closeness and eigenvector centrality. The first three were proposed by Freeman (1978) and were only designed for unweighted networks. Recently, Brin and Page (2012) came up with extensions to weighted networks. The fourth metric - eigenvector centrality - was later proposed by Bonacich (1987) and has its foundations on spectral graph theory. It became especially popular after being used as the basis of the well-known Google's PageRank algorithm, which we will talk about in the next Section. Although more actor-level statistical measures were proposed in the literature, in this subsection we will focus on explaining the mentioned measures of centrality. These measures determine the relative importance of an actor within the network, showing how the relationships are concentrated in a few individuals and, therefore, giving an idea about their social power. Higher centrality measures are associated to powerful actors in the network, since their central position offers them several advantages, such as easier and quicker access to other actors in the network (useful for accessing resources such as information) and the ability to exert control over the flow between the other actors (Freeman, 1978). These central actors are also called "focal points". At the end of the section, we will also introduce the concept of transitivity and explain how it can be computed using a clustering coefficient.

The reader must take into account that some of these actor-level metrics (e.g. degree, betweenness, and closeness) may need to be normalized in order to perform comparisons of networks with different orders and sizes.

Degree Centrality: The degree centrality of a node is the number of nodes directly connected to it or the number of edges incident on it. In the case of recurring links between two nodes the number of unique edges are considered. In the case of digraphs, it can be categorised into two types based on the direction of edges. The number of nodes with edges pointing towards a node is called its indegree centrality. While in the opposite direction from a node is its outdegree centrality. It is usually denoted as $deg(v)$ or $C_D(v)$ for a degree centrality of a vertex v. The indegree and outdegree centralities are denoted with $+$ and $-$ symbols as deg^- and deg^+ respectively.

Degree centrality is regarded as central and important measure in network analysis. In social networks it portrays the importance of nodes with number of friends, posts or followers/followees. The direction of the links also distinguishes the importance. The number of incoming links are regarded as the basis for prestige or support of a node, While the number of outgoing links are sometimes considered as influence. Not always the degree centrality refers to the similar characteristics as above. In some cases, more number of incident links can indicate fraud or extensive activity. More number of incoming links/-calls can also refer to service phone numbers in the context of telecommunications. The type of links as negative or positive also influence the position of node. Therefore the importance relates to the context of analysis and the domain.

Weighted Degree Centrality: It is the sum of all the weights over the edges connecting to a node. In the recurring links scenario the weights on unique edges. Denoted as *wdeg* or simple *w*. Again, we can compute the weighted indegree and outdegree centralities as the sum of weights over incoming and outgoing edges incident on a node respectively. Weights are often assumed as positive, however negative weights can also exist. Weights usually indicate the strength of relationship. They not only modify the way degree centrality is calculated but also impact other measures like betweenness, closeness etc in multi-graph, recurring link or weighted networks.

Betweenness Centrality: The betweenness centrality (C_B) of a node is the number of shortest paths that pass through it. Where a path is a sequence of connected edges between any two nodes in the network. It can be expressed as in Equation 2.1.

$$C_B(v) = \sum_{i,j \in V(G) \setminus i} \frac{ij(v)}{st}, \tag{2.1}$$

Where ij denotes the number of shortest paths between vertices i and j (usually $\sigma_{ij} = 1$) and $\sigma_{ij}(v)$ expresses the number of shortest paths passing through node v. High betweenness nodes act as a gateway to pass or block information between different communities in the network. Hinz et al. (2011) found high betweenness and degree centrality nodes yielding better results in viral marketing when chosen as seed points.

Besides being a property of node, betweenness can also indicate the position of an edge in the network. The edge betweenness can be calculated in the similar way considering an edge e instead of a node n. The edges with high betweenness act like bridges for passing information between communities. Usually bridges are weak ties with the absence of triangles (which are strongly knit groups). The strength of these weak ties is

studied in Granovetter (1973). These are weak ties but strong edges in terms of betweenness.

Closeness Centrality: The reciprocal of the average length of shortest paths between a node and all the other nodes in the graph is said to be its closeness centrality (C_C). It signifies how close a focal node is to all the other nodes in the network. The nodes with high closeness centrality are able to reach all the nodes in the network easily relative to the other nodes in the network. In the networks with a number of components (for the definition of components refer section 2.3.4.2), the closeness can be computed on the nodes within the components. Since the closeness of nodes across components will be ∞. The Equation is given below:

$$C_C(v) = \frac{n-1}{\sum_{u \in V(G) \setminus v} d(u,v)}. \tag{2.2}$$

Where n is the number of nodes in the graphs. Bond III et al. (2004) found closeness centrality to be related with reputational effectiveness in organization.

It can happen the node with high degree, betweenness and closeness centrality can all be the one based on the structure of a network. Nevertheless a node in a network can also be quantified based on its values for two or more centrality indices together, by weighting a metric more than another or uniformly.

Eigenvector Centrality: Eigenvector centrality is a relative score which is based on the idea that links to the nodes with high score increases the score of the focal node. This score is given by the first eigenvector of the adjacency matrix. The basic idea behind eigenvector centrality is that the power and status of an actor are recursively defined by the power and status of his/her alters. Alters is a term frequently used in the egocentric approach of social networks analysis, and it refers to the actors that are directly connected to a specific actor, called ego. In other words, we can say that the centrality of a given node i is proportional to the sum of the centralities of i's neighbors. This is the assumption behind the eigenvector centrality formula, which is as follows:

$$x_i \frac{1}{\lambda} \sum_{j=1}^{n} a_{ij} x_j, \tag{2.3}$$

Where x_i/x_j denotes the centrality of node i/j, a_{ij} represents an entry of the adjacency matrix A ($(a_{ij}) = 1$ if nodes i and j are connected by an edge and $(a_{ij}) = 0$ otherwise) and λ denotes the largest eigenvalue of A.

Eigenvector centrality is a more elaborated version of the degree, once it assumes that not all connections have the same importance by taking into account not only the quantity, but especially the quality of these connections.

Clustering Coefficient Clustering coefficient of a node is the extent to which its neighbours are connected. It is the proportion of links between a node's neighbours to the total number of links that can exist between them. Average clustering coefficient is a global measure for the whole network given by averaging the clustering coefficients of all the nodes in a network.

Watts and Strogatz (1998) proposed a local version of the clustering coefficient, denoted c_i and $i = (1,...,n)$. In this context, transitivity is a local property of a node's neighbourhood that indicates the level of cohesion between the neighbours of a node. This coefficient is, therefore, given by the fraction of pairs of nodes, which are neighbours of a given node that are connected to each other by edges (see Equation 2.4).

$$C_i = \frac{2|e_{jk}|}{k_i(k_i-1)} : v_j, v_k \in N_i, e_{jk} \in E \tag{2.4}$$

Where N_i is the neighbourhood of node v_i, e_{jk} represents the edge that connects node v_j to node v_k, k_i is the degree of node v_i, and $|e_{jk|}$ indicates the proportion of links between the nodes within the neighbourhood of node v_i.

Eccentricity Eccentricity of node in a connected graph is its maximum distance from any other node in it. For a disconnected graph the eccentricity of all the nodes are said to have infinite eccentricity. The maximum eccentricity of any node in the graph is the diameter of the graph.

2.3.2 Network-level Metrics

Before explaining each one of the network-level statistical measures, there are three fundamental concepts that should be first introduced: path, geodesic distance between two nodes and eccentricity of a vertex.

A path is a sequence of nodes in which consecutive pairs of non-repeating nodes are linked by an edge; the first vertex of a path is called the start vertex and the last vertex of the path is called the end vertex. Of particular interest is the concept of geodesic distance, or shortest path, between nodes i and j, denoted as $d(i, j)$. The geodesic distance can be defined as the length of the shortest path, or the minimal path, between nodes i and j.

In turn, the eccentricity is the greatest geodesic distance between a given vertex and any other in the graph, as defined in Equation 2.5. These three concepts are on the basis of most of the network-level metrics we are going to introduce, namely, the diameter/radius, the average geodesic distance, the average degree, the reciprocity, the density and the global clustering coefficient.

$$\varepsilon = \max_{i \in V(G) \setminus v} d(v,i). \tag{2.5}$$

2.3.2.1 Diameter and Radius

The diameter D is given by the maximum eccentricity of the set of vertices in the network and, analogously, the radius R can be defined as the minimum eccentricity of the set of vertices, as defined in Equations (2.6). Sparser networks have generally greater diameter than full matrices, due to the existence of fewer paths between pairs of nodes. Leskovec et al. (2005) discovered that, for certain types of real-world networks, the effective diameter shrinks over time, contradicting the conventional wisdom of increasing diameters. In the context of SNA, this metric gives an idea about the proximity of pairs of actors in the network, indicating how far two nodes are, in the worst of cases.

$$D = \max\{\varepsilon : v \in V\}, \qquad R = \min\{\varepsilon : v \in V\}. \tag{2.6}$$

2.3.2.2 Average Geodesic Distance

The average geodesic distance for all combinations of vertex pairs in a network is usually denoted by l and is given by Equation (2.7) below.

$$l = \frac{1}{\frac{1}{2}n(n-1)} \sum_{i \geqslant j} d(i,j), \tag{2.7}$$

where $d(i,j)$ is the geodesic distance between nodes i and j, and $\frac{1}{2}n(n-1)$ is the number of possible edges in a network comprising n nodes. This metric gives an idea of how far apart nodes will be, on average. For instance, in the SNA context the average geodesic distance can be used to measure the efficiency of the information flow within the network.

When there is the case of a network having more than one connected component, the previous formula does not hold, since the geodesic distance is conventionally defined as infinite when there is no path connecting two vertices. In such situations, it is more appropriate to use the harmonic average geodesic distance, defined in Equation 2.8, once

it turns infinite distances into zero nullifying their effect on the sum.

$$l^{-1} = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geqslant j} d(i,j). \tag{2.8}$$

2.3.2.3 Reciprocity

Reciprocity r is a specific quantity for directed networks that measures the tendency of pairs of nodes to form mutual connections between each other. There are several ways to compute this metric. The most popular and intuitive way is to compute the ratio of the number of mutual connections in the network to the number of all connections, as shown in Equation 2.9.

$$r = \frac{\#mut}{\#mut + \#asym}, 0 < r < 1, \tag{2.9}$$

where $\#mut$ denotes the number of mutual dyads and $\#asym$ the number of asymmetric dyads. Adopting this definition, the value of reciprocity represents the probability that two nodes in a directed network point to each other. By definition, in an undirected network, reciprocity is always maximum $r = 1$, since all pairs of nodes are symmetric.

2.3.2.4 Density

Density ρ is an important network-level measure, which is able to explain the general level of connectedness in a network. It is given by the proportion of edges in the network relative to the maximum possible number of edges, as defined in Equation 2.10.

$$\rho = \frac{m(G)}{m_{max(G)}}, 0 < \rho < 1, \tag{2.10}$$

where $m(G)$ is the number of edges in the network and $m_{max(G)}$ denotes the number of possible edges, which is $\frac{n(n-1)}{2}$ for undirected networks and $n(n-1)$ for directed ones. Density is a quantity that goes from a minimum of 0, when a network has no edges at all, to a maximum of 1, when the network is perfectly connected (also called complete graph or clique). Therefore, high values of it are associated to dense networks, and low values are associated to sparse networks.

2.3.2.5 Global Clustering Coefficient

There are several ways to compute the global version of the clustering coefficient. We adopt the one proposed by Watts and Strogatz (1998), that obtains the global clustering coefficient c, for the whole network, through the computation of the average of all lo-

cal values $c_i(i = 1,...n)$, as shown in Equation 2.11. Small-world networks (Watts and Strogatz, 1998), such as the ones we find in real social contexts, are characterized by high global clustering coefficients, meaning that the property of transitivity among nodes emerges more often and in a stronger way, increasing the probability of clique formation.

$$c = \frac{1}{n}\sum_{i} c_i. \tag{2.11}$$

2.3.3 Statistical Analysis of Networks

2.3.3.1 Distributions of Node Level Metrics

Node metrics identify the properties of nodes. On a network level it is essential to understand the probability distributions of the values of node metrics for each and every node in the network. This function has been widely used in the literature to understand the structure of a network and realising the models for network formation and evolution (Leskovec et al., 2007a; Leskovec and Faloutsos, 2006). For example closeness, betweenness and degree distributions. These distributions in most of the real-world graphs follow close to power law functions (Faloutsos et al., 1999). Other known distribution functions of real-world applications data include power laws with exponential cutoffs Newman (2001a), log normal McGlohon et al. (2011) and double pareto log normal distributions Seshadri et al. (2008), etc.

2.3.3.2 Mean of Node Metrics

In some situations it is necessary to summarize the distributions of node indices in a network. However, in the long and fat tailed distributions as discussed above the arithmetic mean is necessarily not the measure of central tendency. In such cases it doesn't provide the sufficient statistics but can serve as a property describing the network. Therefore, in some sense it provides the information about a network.

The most commonly used metric is the average degree centrality of the network. As the name suggests it is the mean of the degree centralities of all the nodes in the network. It is represented in Equation 2.12. According to Costa et al. (2011) it gives an idea about the global connectivity of a network.

$$\bar{k} = \frac{1}{n}\sum_{i=1}^{n} k_i. \tag{2.12}$$

Average Weighted Degree centrality is also a mean as the above metric, but of the weights over all the edges in a network. It also follows similar distributions as specified above.

2.3.3.3 Densification Power Law:

Densification power law Leskovec et al. (2005) is known as a functional property of time-evolving real graphs. It states that the edges in the evolving networks grow super linearly over number of nodes, with a densification exponent α as given in equation 2.13. Where α typically lies between 1 and 2 in the real-world networks studied by the authors.

$$e(t) \propto n(t)^a \tag{2.13}$$

Where e(t) and n(t) denote the number of edges and nodes of the graph at time t, and a is an exponent that

2.3.4 Ego Networks

An ego network is a local network of a particular node. An ego represents a focal node from the network and all the other nodes in the ego network connected to it are called alters. An ego centric network maps the relationships of an ego with alters and also between themselves. An ego network can be of level/radius $l = 1$ (i.e comprising of nodes only adjacent to the ego), $l = 2$ (adding nodes adjacent to the nodes at $l = 1$), ...$l = D$ (diameter of the graph) gives the whole network being considered. The levels generally studied in the ego network analysis are 1 and/or 2. The alters with direct connections to the ego are called primary alters, while the alters adjacent to primary alters are called secondary alters and so on. Wellman (1996) describes an ego network as a personal network. The studies made by Everett and Borgatti (2005) indicate that the local ego betweenness is highly correlated with the betweenness of the actor in the complete network.

2.3.4.1 Ego network Analysis

Network analysis, from the viewpoint of egos, has attracted much attention over the last decade. Some of the predominant reasons include scalability, because of the exponentially growing data it has been difficult to analyze the whole network en masse. analyzing ego networks not only gives insights into the whole network but can be exploited extensively with memory constraints. The growth pattern of ego networks highly influences the growth of social networks (Tabassum and Gama, 2016a). Epasto et al. (2015) argue that it is possible to address important graph mining tasks by analyzing the ego-nets of a social

network and performing independent computations on them. Akcora and Ferrari (2014) shows how social trust can be measured from user's ego network connections. Secondly, in the prevalent trend of user profile building and personalization in applications (Liu et al., 2018), preferences (Pereira et al., 2018), recommendations (Sun and Zhu, 2013) and services (Wang et al., 2008), ego network analysis and mining is quite pertinent. It is applicable in many other realms including IOT to tackle tremendous data generated from personal interactions. Also serves in studying the structural behaviors of influential, powerful or controlling nodes etc. Though there are numerous applications and advantages of ego network analysis and mining, the research work in this area is still in its infancy, mainly because of the restrictive structure of ego networks. Nevertheless, below we discuss some ego-based network level measures that can be used for ego network analysis besides the conventional statistical measures and the SNA measures discussed above. Though they are network level measures, they are designed to particularly address the properties/characteristics of an ego (node level) in its personal network but not the alters'. Some of the metrics briefed below (Effective Size, Efficiency and Constraint) were introduced by Burt (2009) to analyze personal networks. To get a more detailed explanation of those metrics the readers can also go through (Hanneman and Riddle, 2005). The practitioners of R (Ihaka and Gentleman, 1996) can use egonets package for implementing these measures. Before we discuss the measures below, the readers need to understand the concepts of redundancy and structural holes. When more than one path exists between two nodes in a network it is said to be **redundant**. A **structural hole** is a separation/link between non redundant contacts. It can also be a bridge between two nodes connected to different clusters.

2.3.4.2 Number of Components

Components are subgraphs in which all the pairs of vertices are connected to each other by at least one path and have no connections with the rest of the graph. In the directed graphs, when the nodes are reachable from every other node while ignoring directions is called a weakly connected component. If every node is mutually reachable from every other node, then the components are strongly connected.

In an ego network, the components are considered by ignoring the connections to the ego. The measure of the number of components in egos neighborhood shows the importance of ego as a bridge between components. The more the number of components (large) in ego's neighborhood, the ego is treated to be more important in regards to reaching many groups with a single point of contact (example: spreading information or virus).

2.3.4.3 Effective Size

The measure of effective size portrays the control of ego over alters or the benefit received for every unit invested over alters. It is the average non-redundancy score of all the primary alters in an ego network. It is given by the number of alters in the ego network minus the average degree of primary alters in the ego network while not considering the edges adjacent to the ego. If there are no edges between alters themselves then the ego is the only bridge between them with the highest betweenness. For example, when two primary alters are strongly connected to each other the information benefit to the ego from both of them is probably the same. In this context of information benefit they are considered redundant. The same case applies when two primary alters are not connected to each other but are connected to a mutual secondary network. Therefore information benefits are maximized in a non-redundant network. A limitation of this measure is that if there is another bridge or path between two nodes except ego and is not considered because it is not included in the ego network, then the controlling power of ego will be overestimated.

In the sparse networks like social networks (where density ≈ 0) the effective size of network increases as the number of alters increase, whereas in the dense networks it remains constant. Therefore when the number of alters increase in the sparser networks we can assume that the effective size is increasing. The maximum limit of effective size is equal to the number of alters in the network.

$$ES_e = \sum_j [1 - \sum_q p_{eq} m_{jq}], q \neq e, j. \tag{2.14}$$

Where p_{eq} is proportion of i's energy invested in relationship with q, and m_{jq} is calculated as j's interaction with q divided by j's strongest relationship with anyone. In the simplest form it can be given as,

$$ES_e = n - \frac{\sum_{a=1}^{n} d_a - 1}{n}. \tag{2.15}$$

Where n is the number of alters and d_a is the degree of an alter a. Borgatti (1997) reformulated the above equations in a more simplest form which is given as equation 2.16. Where t is the total number of ties to the ego network while excluding the ties to ego.

$$ES_e = n - \frac{2t}{n}. \tag{2.16}$$

2.3.4.4 Efficiency

Efficiency is a normalized form of the effective size of an ego-network. It is the effective size of an ego-network divided by the number of alters in it. Efficiency always lies between 0 and 1 (inclusive).

For example, if there are two or more ego networks of different sizes and you want to compare the benefits/authority (application-specific) etc., of the egos in them over their alters, then the efficiency is an appropriate measure as it averages per alter. The ego-network with high clustering coefficient would have less efficiency.

2.3.4.5 Constraint

Constraint is similar to redundancy as in the measure of effective size but quite distinguishable as it also considers structural holes around every alter in the network. If the network is more redundant with less structural holes then the ego is more constrained. When two or more alters in an ego network are connected to each other then the ego is constrained in its actions. Constraint of an ego e is the sum of constraints of all the alters over ego in the ego-network. Constraint (C_{ej}) of each alter j over the ego e is dependent on the proportion of its relationship to the ego and all the other alters in the network P_{ej} and the sum of proportional relationship with the other alters in the network P_{qj} and also their proportional relation with the ego P_{eq}. Which is given in the equation below.

$$C_{ej} = (P_{ej} + \sum_{q} P_{eq}P_{qj})^2, q \neq e, j. \tag{2.17}$$

One of the applicability of the above ego-centric measures was given by Burt (2009) to optimize structural holes in a network to yield maximum benefits from them. He analyzed the ego networks of managers and their growth pattern. He found that managers with increasing effective size reached top positions. He also included that though the effective size has positive effects on information benefits and control it is dominated by the negative effects of constraint. According to his observations, the players/egos with relationships free of structural holes at their own end and rich in structural holes at other end are structurally autonomous and well positioned for the information and control benefits he regarded. Leveraging his observations can help investigate networks to gather information and insights. Tabassum and Gama (2016a) employed these measures to compare evolving ego-networks and their samples, to evaluate their efficiency.

2.3.4.6 Krackhardt Efficiency

Krackhardt Efficiency is one of the four measures defined by Krackhardt to measure the extent to which a graph is an out-tree. It is a measure of non-redundancy in multiple components of a graph or multiple weak components of a digraph (For the definition of components refer Section 2.3.4.2). If we consider an ego network as a hierarchical structure and ego being the root node then the non-redundancy of edges can be calculated using this measure. The graph is said to be highly efficient if there are $(N_i - 1)$ number of links between N_i number of nodes in every component G_i of a graph G. Efficiency is inversely proportional to the density of each component in the graph. If the density increases, the efficiency decreases. It is given by the equation below. Where E(G) is the total number of edges in graph G. In figure 2.1 the krackhardt efficiency is calculated considering an undirected graph. Additionally, other measures defined by Krackhardt can also be employed over ego networks to identify specific properties delineated by them.

$$1 - \frac{E(G) - \sum_{i=1}^{n}(N_i - 1)}{\sum_{i=1}^{n}(N_i(N_i - 1) - (N_i - 1)} \tag{2.18}$$

Demonstration: To analyze how the position of an ego is reflected by the above measures, we show some simple examples of ego networks in figure 2.1, where **E** is an Ego with few alters. Their respective measures are demonstrated in Table 2.2 which expresses the relation between these measures.

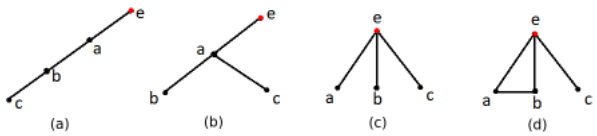


Figure 2.1: Structurally different ego-networks for demonstration.

Table 2.2: Effective Size, Efficiency, Krackhardt Efficiency and Constraint measures of ego-networks from figure 2.

Measures/Networks	Figure 2.1(a)	Figure 2.1(b)	Figure 2.1(c)	Figure 2.1(d)
Effective Size	2.0	1.0	3.0	2.3333
Efficiency	0.6666	0.3333	1.0	0.7777
Krackhardt Efficiency	0.6667	0.6667	0.6667	0.4444
Constraint	1.25	1.2222	0.3333	0.6111

2.4 Network Mining

In the above section we presented the analytical methods to understand and quantify the behavior of nodes and networks. However the node-centric measures can also serve as features for the learning tasks given below. These tasks mentioned below are specific to networks considering the connections between nodes.

2.4.1 Link Prediction

The link prediction problem has been traditionally approached in a static environment: given a snapshot of the network, the goal is to infer which links are missing (Liben-Nowell and Kleinberg, 2003). Assuming the network will evolve to other state in the future, the link prediction problem may also be interpreted as the problem of predicting which links are more likely to appear in the future, given the current state of the network.

Temporal link prediction refers to the problem of link prediction in time-evolving networks, in which multiple snapshots of the network are available. In global terms, this problem is defined as follows: given the states of the network for the previous T time instants, how to predict future (new or re-occurring) links? Thus, given the sequence of states of the network at instants 1 to T, the goal is to predict which are the links which are more likely to occur at instant $T+1$ (Fernandes et al., 2017).

The problem of temporal link prediction has been tackled by considering both time-agnostic (in which the temporal information is not exploited) and time-aware methods (in which temporal information is incorporated).

In the context of time-agnostic methods, the most common approach is to collapse all the network time stamps and to apply the traditional methods, designed for static environments, to the collapsing result. These traditional approaches consist in computing a similarity score between each pair of network nodes so that higher scores reflect higher similarity. These scores are usually defined based on the node neighborhood of the nodes (Newman, 2001b; Adamic and Adar, 2003; Salton and McGill, 1986) or on the paths between the nodes (Katz, 1953; Lü et al., 2009; Papadimitriou et al., 2012).

O'Madadhain et al. (2005a) and Wang et al. (2007) considered classification oriented methods. The idea of these methods is to use a set of network features to train a classifier, which is further applied to predict future links. Both works used logistic regression, however, in the second work, the authors considered co-occurrence probabilities in addition to the topological and semantic features.

Regarding time-aware methods, the most common approach is to consider time-series-based methods. The idea of such methods is to use the state of the network in the given

instants to construct time-series, which are further forecasted. The differences among the existing methods reside on at least one of the following specifications: (i) the type of feature being modeled by the time-series, (ii) the forecasting models and (iii) the forecasts post-processing. In this context, one of the first methods was proposed by Huang and Lin (2009), which modeled occurrence frequency using ARIMA. The major limitation of this work is its inability to predict new links, that is, the method only predicts re-occurring links. This limitation was addressed in precedent works by considering other types of features such as similarity scores (da Silva Soares and Prudêncio, 2012; Hajibagheri et al., 2016; Güneş et al., 2016). These methods also extended the existing work by considering other forecasting models. Moreover, in (da Silva Soares and Prudêncio, 2012; Hajibagheri et al., 2016), the authors subjected the forecasting result to an SVM.

In (Dunlavy et al., 2011; Spiegel et al., 2011), the authors considered the modeling of the time-evolving network as a tensor and resorted to tensor decomposition techniques, combined with forecasting models to estimate the future time slice of the tensor, corresponding to the future state of the network.

Bringmann et al. (2010) and Juszczyszyn et al. (2011) proposed methods which were based on the mining of evolution patterns.

2.4.2 Community Detection

Community detection is very well known problem in data networks. To understand the community detection problem the reader should know what is a community in terms of a social network. Communities are groups, modules or clusters of nodes which are densely connected between themselves and sparsely connected to the rest of the network. The connections can be directed, undirected, weighted etc. Similarly, the context of a relation can also be different. The communities can characterize a group of nodes as friends when the link type is "friend of". Similarly, in metabolic networks the proteins can be grouped based on the similarity of functions, where the link type is "similar function". Therefore, the type of connection defines the context of the community formation. A network of same nodes can form different communities when the relation type is different. Nevertheless, the links can be straightforward like friendship on facebook or it can be derived, example similarity as stated above. The communities can be overlapping (where a node belongs to more than one community) or distinct. Community detection is similar to the problem of clustering data points. It has a wide scope of applicability in real-world networks. Besides that, it has been typically used for partitioning data into communities for feeding parallel processors which would minimize the cross communications.

A number of community detection algorithms have been proposed so far, of which most of them are not scalable to very large networks, which is the current need. Moreover how to evaluate community structures is still an open problem. If a network had all the community structures well-defined with a group of nodes forming cliques and only one node making a connection with the rest of the network (which are termed as "whiskers" in Leskovec et al. (2009)), outlining clusters would have been an uncomplicated task. Which is not the case with real-world networks that are very sparse and huge at the same time, hence increasing the complexity. Leskovec et al. (2009) found the nodes from larger communities blend into the core of network and consequently decreasing the quality of community. Nevertheless, there have been quite a number of methods trying to address this optimization problem, of which some of them we discuss here.

2.4.2.1 Hierarchical Algorithms

Hierarchical algorithms fall into two categories, divisive and agglomerative Scott (1988).

Divisive Methods: These methods start from the network level assuming it as a community, then disregarding edges to separate the network into communities and maximize some function. One of this kind is the seminal work of Newman and Girvan (2004). Where the edges with high betweenness are chosen to be removed from the network recursively. As high betweenness edges are known to be weak bridging ties. However, betweenness itself is an expensive metric and updating it after every edge removal makes this algorithm computationally expensive.

Agglomerative Methods: These behave in an opposite way to the preceding one. They begin at the node level and keep on merging nodes that are similar or related in some sense. This is done until a halt point. The very first examples include the method given by Breiger et al. (1975), which works by iteratively forming product moment correlation matrices. The setting was also applied to mutli-graph data with more than one type of relation. However, it also joins the expensive class of algorithms.

The above methods are categorised based on their approach to the problem. However, the literature can be further classified by zooming into more specific techniques followed in detecting communities while following the above approaches.

2.4.2.2 Optimization of Quality Metrics

One of the fundamental techniques of partitioning the nodes into communities is by maximizing a specific quality metric. The same metrics are used to validate and compare the

quality of communities. Therefore, it inherits some biases. There are a number of quality metrics studied in the previous works. However, there is no consensus on the one that works well in all the applications. Here we discuss some of them.

Modularity Maximization One of the extensively used optimisation metric in the literature is modularity. Modularity function is defined as the number of edges within communities minus the number of expected edges in the same at random Newman (2006a). This was devised into equation 2.19.

$$Q = \frac{1}{2m}\sum[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j), \tag{2.19}$$

where m is the number of edges, k_i and k_j represent, respectively, the degree of nodes i and j, A_{ij} is the entry of the adjacency matrix that gives the number of edges between nodes i and j, $\frac{k_i k_j}{2m}$ represents the expected number of edges falling between those nodes, c_i and c_j denote the groups to which nodes i and j belong, and $\delta(c_i, c_j)$ represents the Kronecker delta. Q typically lies between -1 and 1. Accordingly, the larger Q the better community structure.

Optimizing modularity is computationally hard. To reach an optimal value for Q, the algorithm needs to exhaustively search for the possible divisions, which would increase the time exponentially with the increase in number of vertices Newman (2004a). Therefore, in the pursuit of an approximate method for optimizing this function a number of different methods have been investigated in this area such as spectral clustering (Shen and Cheng, 2010), extremal optimization (Duch and Arenas, 2005), genetic algorithms (Shang et al., 2013), simulated annealing (Mu et al., 2015), particle swarm optimization (Li et al., 2019) etc. Greedy algorithms being one of them.

Newman and Girvan (2004) was the first to introduce a greedy algorithm based on modularity maximization for finding communities. It is a agglomerative approach of hierarchical clustering. It starts by considering each node as a community and iteratively joining pairs of nodes (forming communities) that result in modularity increase or a small decrease. Some other greedy optimisation algorithms with enhanced performance include (Clauset et al., 2004a; Wakita and Tsurumi, 2007; Schuetz and Caflisch, 2008; Blondel et al., 2008a). From which Louvain (Blondel et al., 2008a) is the one popular method which has been implemented and advanced by many other research works.

Other Metrics Besides modularity there are quite a lot of metrics that have been explored in the previous works, to optimize and also to evaluate. Some of them are cohe-

siveness, normalized cut and separability (Shi and Malik, 2000), etc. A detailed review of the metrics can be found in Chakraborty et al. (2017).

Furthermore, while detecting communities researchers also focus on the intricate details of communities such as identifying central positions in communities (Newman, 2006b). Tracking evolution of communities over time (Dakiche et al., 2019) etc. For more detailed study on community detection one can refer to (Parthasarathy et al., 2011), a latest survey can be found in (Khan and Niazi, 2017), and on dynamic networks Rossetti and Cazabet (2018).

2.4.3 Anomaly Detection

Anomaly detection deals with the identification of unusual patterns or abnormal behaviors in the data. The items or observations that exhibit those behaviors are referred to as anomalies, outliers, deviations or exceptions Hodge and Austin (2004). Anomaly detection is closely related to the problem of detecting faults or frauds and therefore critical in many application scenarios. There are quite a number of surveys exploring the problem of anomaly detection. For a detailed review the interested readers can refer to the given articles for methods applied on data without relations (Hodge and Austin, 2004; Chandola et al., 2009; Niu et al., 2011; Habeeb et al., 2019) and on networks (Savage et al., 2014; Gupta et al., 2013; Kaur and Singh, 2016; Anand et al., 2017).

Anomalies in graphs can be nodes, edges or sub-graphs. Below we present the conventional methods for anomaly detection in data. However the anomaly detection models not considering relations between data items can be applied to the networks as well, by considering nodes or edges as data points and the properties derived from the interconnections as their associated features, as demonstrated in Chapter 8. The properties could be the measures discussed in the preceding section. In case of streaming or evolving networks these measures can be incrementally calculated as described in Chapter 3.

2.4.3.1 Supervised Methods

Necessarily, supervised methods such as classification use labelled data which is quite difficult to obtain in this case. Though obtained, the anomalies are very less compared to the normal items resulting in class imbalance problem, which is investigated in Vilalta and Ma (2002); Chawla et al. (2004). Some predictive models exercised for anomaly detection in the literature include, KNN (Gu et al., 2019), one class SVM (Li et al., 2003), Principal component analysis (Tax, 2002), data streams (Wu et al., 2019), statistical methods (Hodge and Austin, 2004) etc.

2.4.3.2 Semi supervised Methods

These methods are employed when some instances with labels are available and some without. We refer to some of the models dealing with this problem here, such as using ensemble of feature models (Noto et al., 2012), deep belief nets (Wulsin et al., 2010), etc.

2.4.3.3 Unsupervised Methods

This is a most common approach in anomaly detection as most of the times labels are not available. The results are evaluated based on domain knowledge or expert guidence. Some of them include K-means (Münz et al., 2007), K-medoids Deep neural network for time series data (Zhang et al., 2019), frequent itemsets (Leung and Leckie, 2005), Mahalanobis distance (Hodge and Austin, 2004), time series (Rebbapragada et al., 2009).

2.4.3.4 Graph Specific Methods

Akoglu and Dalvi (2010); Hassanzadeh et al. (2012) realized stars and clique like structures indicating anomalous behavior. Which was detected by fitting a power curve over number of nodes vs edges or other properties in a user's ego network. However, it is unfeasible for large graphs to extract ego networks of all the users. Similarly, finding dense sub-graphs has been exploited in the literature for anomalies, few methods include using local heuristics (Beutel et al., 2013), spectral subspaces (Jiang et al., 2014). However, there are not many methods addressing this issue in graphs with various contexts and perspectives, moreover dynamically. Scalable methods for large and temporal networks is still an open problem.

2.5 Chapter Summary

In this chapter, we presented a concise overview of social network analysis methods, objectives, and applicability. A number of SNA measures and related tasks in view of different types of networks are demonstrated. With the discovery of networks in most of the applications' generated data and the quality of information extracted, network analysis is gaining much popularity these days. The complexity is increasing as the available amount of data is increasing. Advancements in processing, manipulating and mining high-velocity massive scale networks is one of the current vital concerns. With the predominance of web 2.0, IOT and Industry 4.0 etc., it is only going to get more demanding and challenging.

As the networks generating in real-time are not static but dynamic and evolving, the recent works have been profoundly interested in exploring the growth patterns, mining problems and resolving the challenges associated with it. Some of the latest works are concentrated in designing algorithms for faster, incremental and memory efficient computations of SNA measures (discussed in this Chapter) for very large and high velocity graphs, which is still a challenge for many of the metrics that needs traversing entire or most of the graph on every update. Link prediction, community detection and clustering etc., are actively studied fields in recent decades but still lack efficiency. The current and future trends also include application-specific usability, scalability, and enhancements in these areas. The emerging lines of applicability are in the areas of social reputation, smart cities, multi-agent systems, intelligent objects, bio-informatics, earth sciences, cognitive sciences, mobility patterns, recommendations etc., besides the existing realms of fraud detection, social media, gene expressions, protein interactions, marketing, churn prediction etc. The recent surging demands by these applications on the complex real-world have also required advancements in the area of diverse network topologies like multi-level, heterogeneous and evolving networks etc. Therefore, this Chapter paves the way for a basic understanding of more complex problems associated with network analysis.

Chapter 3

Streaming Network Analytics

3.1 Introduction

Networks with changes in the number/behavior/features of nodes and links as a function of time are known as evolving networks. For example, the social network of people living in this world can be considered evolving, as new nodes get added while some of the nodes expire and connections/relations between the nodes keep on forming and breaking across time. This phenomenon causes changes in the structure of the network as a whole, across time. There exist many examples of time-evolving networks, as shown in Chapter 1, most of them being generated from real-world applications. Statistical analysis of these networks in itself is a challenge because of their transient structure and distributions, added to their size and complexities. Last two decades have encountered extensive research in the area of network analysis to study the evolution of structure and properties of these networks for various purposes like generating synthetic or artificial networks, developing models for graph mining (link prediction/recommendation, community/anomaly detection, event detection/prediction, classification/segmentation, pattern mining etc.), decision making, solution optimization/influence maximization and network analytics etc. In the subsection (3.3) below, we present a brief survey of works that indulged in the evolution analysis of networked data. Note that the evolving graphs are temporal but the temporal graphs need not always be evolving they can be dynamic though i.e nodes need not be added or deleted in the networks (sometimes the temporal property is only associated with edges).

Graph analytics deals with extracting patterns or information by analyzing and modelling graphs. Streaming graph analytics for evolving networks have augmented set of challenges over the static graph analytical models. Some of which we have detailed in Chapter 1. Here we discuss some possible ways to deal with those challenges and the

ones that have been applied in this thesis. Therefore, this chapter helps in setting up a stage for the later chapters in this thesis.

3.2 Definitions

In this section, we present the definitions followed in the rest of the chapters. That includes the representation of an evolving network stream. While some chapters need an improved definition according to the type of graph, which is explained when needed. We also provide here the definition of a dynamic sample, which is an important problem addressed in this thesis.

Definition 3.2.1 (Evolving Network Stream)**.** We model an evolving network G as a stream of links/edges $\{e_1, e_2, e_3, e_4...\} = E$. Every edge $e_i = (u, v, t)$ is composed of a pair of vertices from V and a time-stamp t, which indicates the time of occurrence of e_i or simply e which is unique. We assume that the edges are streaming in the order of time-stamps. In a directed graph an edge between (u, v) is different from (v, u). The size of edge set E is given by a changing cardinality at every t, which is cumulative of the new edges.

Definition 3.2.2 (Dynamic Sampling)**.** A dynamic sample S evolves with the Graph G such that at any time t, $S(t)$ is representative of $G(t)$ and $S(t) \subseteq G(t)$. In the case of networks, the samples are meant to be representative not only statistically but also structurally.

3.3 Evolving Network Analysis

Apart from using the SNA measures given in Chapter 2 for a static network analysis, these measures are also used to study the dynamic and evolving properties of networks. We delineate some works below, which used these properties to characterize evolving networks based on the regularities from their results of the analysis.

Researchers in this area (Albert and Barabási, 2000; Huberman and Adamic, 1999a; Barabâsi et al., 2002; Krapivsky et al., 2001; Aiello et al., 2000) found that most of the real-world data networks grow by following a power-law degree (in-degree, out-degree or undirected) distribution (at least asymptotically)(for the definition of degree distribution refer to Oliveira and Gama (2012)). Barabási and Albert (1999) reasoned it was because of the preferential attachment of the nodes i.e new nodes attach preferentially to the already well-connected nodes. Dorogovtsev and Mendes (2001) proved that different kinds

of preferential linking produce different types of scale-free networks (whose degree distribution follows a power-law). Reed and Jorgensen (2004) proved that if a stochastic process that grows exponentially, and is observed once 'randomly', the distribution of the observed state will follow power-laws in one or both tails.

Huberman and Adamic (1999b) while exploring the growth dynamics of world wide web, explained that for sites which are typically organized in hierarchical, tree-like, fashion, the number of pages added at any given time to a site will be proportional to those already existing there and the growth rate $g(t)$ of number of pages per site is uncorrelated from one time interval to the other about a positive mean value g_0. As a consequence, each particular growth rate gives rise to a power law distribution. They also indicated that the evolutionary dynamics of the web are dominated by occasional bursts in which a large number of pages suddenly appear at a given site. These bursts are responsible for the long tail of the probability distribution and make average behavior to depart from typical realizations. They concluded that those networks evolve in an asymptotically self-similar structure without having a natural scale.

While most of the previous works concentrated only on the sparsity of the networks, Faloutsos et al. (1999) derived some more interesting relationships in the evolution of networks. They analyzed the Internet topologies over three instances in a year, where the size of network increased by 45%. As a consequence they defined four power-law relationships (one is an approximation) on a growing network, which are stated as: **Power-Law 1 (rank exponent):** The outdegree, d_v, of a node v, is proportional to the rank of the node, r_v, to the power of a constant, R. **Power-Law 2 (outdegree exponent):** The frequency of an outdegree, f_d (the number of nodes with outdegree d), is proportional to the outdegree d to the power of a constant, O. **Power-Law 3 (Eigen exponent):** The eigenvalues, λ_i, of a graph are proportional to the order, i, to the power of a constant, E. Where i is the order of λ_i in the decreasing sequence of eigenvalues. **Approximation 1 (hop-plot exponent):** The total number of pairs of nodes, $P(h)$, within h hops, is proportional to the number of hops to the power of a constant, H.

Newman (2001b) tested the previous theories of clustering and preferential attachment over growing networks. He proved empirically the probability of a link between two nodes is strongly positively correlated with the number of mutual acquaintances/neighbors exponentially and the number of previous links linearly.

Further ahead Barabâsi et al. (2002) analyzed the topological properties of very sparse co-authorship networks (like approx. 70K nodes and 70K edges aggregated) over smaller time steps (compared to the above-referenced literature) of one year for seven years. Based on their empirical measurements the authors discovered that the average degree of these networks increases in time, and the node separation decreases. In addition, they

also found the clustering coefficient of these networks decays with time while relative size of the largest cluster increases.

Leskovec et al. (2007a) extensively studied the evolving properties of networked data from seven different domains with varied time spans and nodes and edges arriving at different speeds. Based on their results they showed that the average degree of these networks increases as a function of time (as given by Barabâsi et al. (2002)), hence densifying these networks over time and their densification follows a power law. Where the edges increasing over the number of nodes with a scaling exponent α which generally lies between 1 and 2. Added to that, the effective diameters of those networks keep decreasing over time. In (Leskovec et al., 2008), the authors tried to find the preference of nodes for the formation of new edges in the network evolution process. They demonstrated that the preferential attachment concept given by Barabási and Albert (1999) based on degree and age of nodes is inherently non-local and biased. On the contrary, the locality of nodes plays an important role in edge formation, and most of the new edges form by typically closing triangles. Therefore locality with preferential attachment can supplement similar network generating models.

The works referred above have mainly focused on the analysis of evolving networks to yield tractable mathematical models or simple generative models of network growth. Consequently, these models can be used to explore their latent properties. The success of these models depends on their ability to incorporate the growth statistics of those networks. Additionally, some studies used the growth statistics of networks to evaluate their sampling techniques (Leskovec and Faloutsos, 2006; Tabassum and Gama, 2016b,a). However, the analysis of networks is also a primary step in developing models in an evolving network mining task.

3.4 Processing Evolving Graph Streams

A data stream is considered an ordered sequence of instances that can be read only once or a small number of times using limited computing and storage capabilities Gama and Gaber (2007). The idea of a data stream is to read data items sequentially, update sufficient statistics and discard it or store on secondary storage. While in the case of graphs there are two kinds of data items, one is objects and the other is the connections between them. The objects are considered nodes/vertices and the connections are links/edges. The stream can be processed as an incoming flow of objects but then we need to build the relationships between them using the information associated with the nodes. We can also stream edges directly as an instance with their associated properties. We followed the second option

of reading edges in the order of time. In Chapter 8 we use those instances to update and extract features for nodes. The edges can then be discarded and only a small number of nodes in question can be held in memory depending on the application.

In the case of dynamic samples (Chapter 5), we process edge streams and store only a small fraction of edges with values that maximize the objective function. Further keep on updating or discounting their value and consequently replacing them with other edges in-case they achieve higher values. For all the processing and programming algorithms in all the chapters through out this thesis we used **JAVA** and applied our own implementations to compute social network analysis metrics in a streaming fashion in Chapters 5-8. Additionally, in other Chapters we used the implementations of socio-metrics in **Gephi**, **Cytoscape** and **R**. For computing statistical measures to validate results we used **R** packages. The network visualizations were carried out in **Gephi**.

Storage of a graph in motion in the main memory is another aspect in stream processing. Graph in motion is the small part of the streaming graph that is retained in the memory for computations. It should be stored in an effective way to accommodate less possible space and ease of computations. A number of data structures such as matrices (two-dimensional arrays), lists, trees, etc., have been effectively used for storing graphs. Usually real-world networks are very sparse therefore storing them in matrices (adjacency, incidence, laplacian, distance) would occupy memory also for a substantial number of zero entries. Hence the possibility is to store only the non zero entries or edges in lists or sparse matrix implementations. While sparse matrix representations are efficient in matrix based operations like multiplications, factorization etc, They are not very efficient for updates, modifications, sorts etc that are essential for streaming and require additional data structures. Lists maintain an index to identify and order objects. Lists are slower with O(n) to access objects without knowing index therefore we employed hash maps which stores an entry as a key value pair and faster time complexity O(1) for search, insert, delete and update operations.

3.5 Evolving Socio-metrics

In Chapter 2, we presented the social network analysis metrics that have been referred to in this thesis. Here we briefly discuss their streaming versions and some methods for computing them over evolving networks.

3.5.1 Measures Recognizing Simple Updates

Socio-metrics such as degree centrality, weighted degree centrality, reciprocity, density, average degree centrality need simple mathematics for performing incremental updates.

Degree Centrality ($deg(v)$): The degree centrality of any vertex v at an instance of time t can simply be updated by incrementing its value by one sequentially as soon as a streaming edge e arrives with v being one of the vertex.

$$deg_t(v) = deg_{t-1}(v) + 1 \tag{3.1}$$

The same is applicable to indegree (deg^-) and outdegree (deg^+) centralities and weighted networks as well. The space complexity is $O(n)$, where n is the number of vertices in V till t which is the result set. However, instead of storing all the nodes and their degree, the number of nodes can be reduced by storing only application relevant nodes or using online sampling techniques such as in the following section. Note in the case of directed networks and recurring links the space complexity increases to O(n+m) where $m = |E|$, as we need to store also the unique edges to keep track if they have occurred previously in the opposite or the same direction respectively in the stream. The time complexity is O(1) using hashmaps.

Weighted Degree Centrality ($wdeg(v)$): In the case of weighted networks $wdeg(v)$ until t is

$$wdeg_t(v) = wdeg_{t-1}(v) + w_t(v,u) \tag{3.2}$$

Where $w_t(v,u)$ is the weight of an edge connecting the vertex v and any vertex u. In the case of recurrent links we need a simple increment as in degree centrality which is given below in Equation 3.3. In this case we don't need to store the edges.

$$wdeg_t(v) = wdeg_{t-1}(v) + 1 \tag{3.3}$$

Reciprocity (r): Reciprocity for a node or the graph at any given point in time in a stream can be given using the above metrics

$$r_t = \frac{(deg_t^+ + deg_t^-) - deg_t}{deg_t} \tag{3.4}$$

For a node the deg is $deg(v)$ and for a graph G the deg is total degree $deg(G) = \sum_{v=1}^{n} deg(v)$.

Density (D): To compute the density of a graph we only need to maintain two counts incrementally, the number of nodes n and the number of edges m in a stream until t as shown in Equation 3.5 for undirected graphs. These counts can also be obtained from the above streaming measures.

$$D_t = \frac{2m_t}{n_t(n_t - 1)} \tag{3.5}$$

Average Degree Centrality (ADC)**:** It is another simple metric that can be computed recursively at every time point using $deg_t(G)/n_t$.

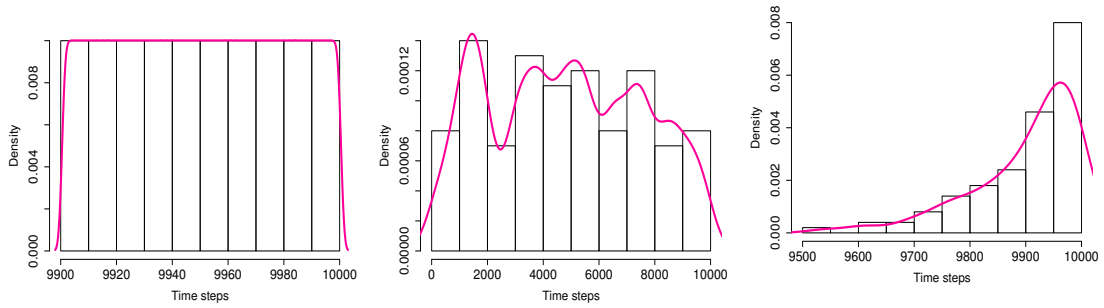
The above formulas maintain the exact measures and are accumulating. The same measures can be implemented using the approaches in the next section to be incremental and decremental at the same time. Path based measures need neighbourhood information of nodes or graph traversals which are relatively complex. However, there are some incremental and streaming implementations for betweenness centrality (Green et al., 2012; Kas et al., 2013a; Kourtellis et al., 2015), closeness centrality (Kas et al., 2013b; Sariyuce et al., 2013), community detection (Sun et al., 2014; Cordeiro et al., 2016) among others.

3.6 Sampling Massive Data Streams

Online sampling has found many applications in the streaming data scenarios where the computational speed is slower than the speed of stream flowing in. Additionally, in the cases where entire data cannot fit in memory some observations arriving in the stream are chosen to be held in the memory and replace or not as the stream grows. Stream sampling is found to be beneficial in many applications like joins, distinct counts, running simulations, extrapolating true data distributions, etc. However, it needs to be done in an application specific way by avoiding loss of information Gama and Gaber (2007). In this section, we present some common sampling strategies and a new variant for sampling data streams which have been fitted out for networks in the following chapters.

3.6.1 Sliding Windows

Sometimes applications need recent information and its value diminishes by time. In that case sliding windows continuously maintain a window size of recent information. It is a common approach in data streams where an item at index i enters the window while another item at index $i - w$ exits it. Where w is the window size which can be fixed or adaptive. The window size can be based on number of observations or time. The later are known as time windows. Windows can be overlapping or discrete.



(a) Time window (b) Reservoir sampling (c) Biased random stream sampling

Figure 3.1: Temporal distributions of samples at the end of stream

3.6.2 Reservoir Sampling

This algorithm Vitter (1985) maintains a reservoir with a predefined sample size k. Firstly the reservoir is filled with the initial items from the stream. Every item after that is computed for the probability k/i of being inserted. Where i is the length of the stream exhausted till then or the index of the item waiting to enter the reservoir. If the probability of the contending item i in the stream is greater than the probability of an item in the reservoir which is $1/k$, then uniformly at random an item j is picked from the reservoir. The picked item j is replaced with the item i in the stream. In case the probability is less the streaming edge is discarded. As i increases its probability to enter the reservoir decreases. Hence, leading to very old items from the stream. To overcome this, some sampling strategies try biasing items in the reservoir by imposing and updating weights of all the items recursively (Efraimidis, 2015), which leads to increased complexity. In the section below, we provide another technique, that does not require explicit weights for biasing.

3.6.3 Biased Random Stream Sampling

Unlike the above algorithm where the probability of items entering the reservoir diminishes as the stream progresses, this algorithm ensures every item i goes into the reservoir with probability 1. An item j from the reservoir is chosen for replacement at random. Therefore, on every item insertion, the probability of removal for the items in the reservoir is $1/k$, where k is the size of reservoir. Hence, the item insertion is deterministic but deletion is probabilistic. The probability of j staying in the reservoir when i arrives is given by $(1-1/k)^{(i-j)}$. As i increases, the probability of j staying in reservoir decreases. Thus the item staying for a long time in the reservoir has an exponentially greater proba-

bility of getting out than an item inserted recently. Consequently, the items in the reservoir are superlinearly biased to the latest time. This is a notable property of this algorithm as it does not have to store the ordering or indexing information as in sliding windows. It is a simple algorithm with $O(1)$ computational complexity.

Histogram plots demonstrating the temporal distributions of the resultant samples at the end of a stream are shown in Figure 3.1. The stream is simulated with 10000 data items distributed uniformly overtime. The samples contain 100 items at any point in time. The figures follows the distributions as suggested in the above algorithms.

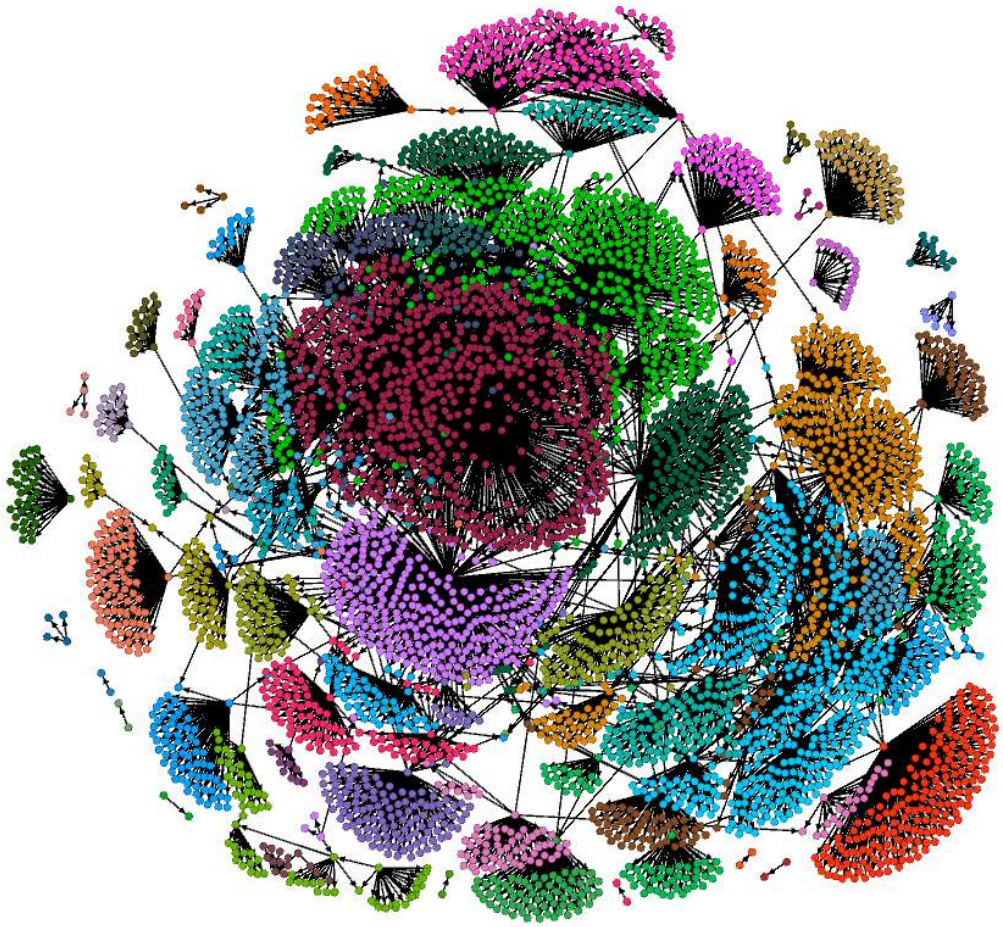


Figure 3.2: A network sample visualization

3.7 Visualization

Visualization of networks represents the connections between nodes effectively. It helps understand the underlying structure and the macro and micro level view of network. However, visualizing very large networks is a challenge, as the nodes from large networks

cover every pixel of the screen without being able to define a picture. The sampling strategies exploited in this thesis can be exercised to address this problem. An example visualization is shown in Figure 3.2. Visualization techniques can also enhance the comprehensibility and explainability of analytics and results. Therefore, it can be exercised impressively to communicate structural patterns with clients and detail their dashboards with real-time evolving network visualizations with results. Further in Chapter 8, we use visualization techniques to interpret the patterns of fraudulent users in a telecom network. Furthermore, visualizing the graphs of evolving networks is much more complex owing to the addition and deletion of nodes and edges over time. We used Gephi (Bastian et al., 2009) and Graphstream library (Dutot et al.) for evolving network visualizations related with this thesis. Graphstream provides also interactive visualizations. For a brief overview on evolving network's visualization layouts and strategies the readers can go through Aggarwal and Subbian (2014) and Cordeiro et al. (2018).

Chapter 4

Sampling Massive Streaming Graphs

4.1 Chapter Overview

The problem of analyzing massive graph streams in real-time is growing along with the size of streams. Sampling techniques have been used to analyze data streams in real-time. However, there are very few techniques for sampling graph streams. Therefore, in this chapter, we present sequential techniques for sampling evolving network streams from our work in Tabassum and Gama (2016b). We implemented the state of the art space saving algorithm for generating topK edges' network samples, and also propose a simple biased version of reservoir sampling, which shows better comparative results than reservoir sampling. When sampling network streams it is difficult to answer questions like, which structures are well preserved by the sampling techniques over the evolution of streams? Which sampling techniques yield proper estimates for directed and weighted graphs? Which techniques have the least time complexity etc.? In this work, we have answered the above questions by comparing and analyzing the evolutionary samples of such graph streams. To evaluate the sampling techniques, we compared the structural metrics from the respective samples.

The experiments were carried out rigorously over a massive stream of 3 hundred million calls made by 11 million anonymous subscribers over 31 days. We also evaluated node and edge-based methods of sampling. Our overall results and observations show that edge-based samples perform well than their counterpart. The three algorithms compared in this chapter are namely, space saving algorithm, reservoir sampling, and a biased version of reservoir sampling. We empirically compared the distribution of degrees and biases of evolutionary samples.

4.2 Background

As described in Gama (2010a), a data stream is an ordered sequence of instances that can be read only once or a small number of times using limited computing and storage capabilities. These sources of data are characterized by being open-ended, flowing at high-speed, and generated by non-stationary distributions in dynamic environments. In terms of such graph streams, sampling is the process of selecting a subset of streaming graph to represent the characteristics of the entire graph stream at a given point of time. Because of the time and space limitations, it is difficult to analyze and mine massive social streams in real-time. Wherefore, samples are generated so as to gain approximate solutions for real-time queries. A very small number of algorithms have been introduced in this research area of streams, which are referenced below.

Considering static graph sampling strategies in Leskovec and Faloutsos (2006), the authors compared some sampling algorithms like random node and edge sampling, random walks and forest fire with aggregated data sets of about five hundred thousand edges. Their goal was to find a general sampling method that would match the full set of the properties of the original graph so that sample can be used for simulations and experiments. Our goal in this work is to find appropriate sample techniques for specific properties of graphs. Some other works of static sampling include Stumpf et al. (2005), where the authors investigated to prove that the subnets of scale-free networks are not scale-free, by random sampling of a static network. Goodman (1961) considered choosing some ego-networks from the entire plenary graph. In that case, the samples are biased to few nodes in the network. Sampling algorithms for pure topology types were discussed by Airoldi and Carley (2005). Krishnamurthy et al. (2005) studied three types of methods for sampling Internet graphs with a reduction of 70 % of the original graphs.

While the above algorithms disregarded the significance of streaming data, some efficient and classical techniques have been proposed so far in the literature for this purpose. However, they didn't consider the connections between data items. The seminal work of Reservoir sampling for data streams was given by Vitter (1985). We apply this algorithm on graph streams and analyze its properties in the following sections. Further, Ksample (Kepe et al., 2015) implemented a variable size sample version of the above reservoir sampling in the same context of data streams. In Chapters 5 and 7, we update the reservoir sizes in the evolution of the stream according to the requirement, making them variable size. Biased reservoir sampling by Aggarwal (2006) is a biased variant of again the reservoir sampling, where the authors introduce exponential weights on data items based on time, such that the new items have a greater probability of entering the reservoir, resulting in the latest information. Cormode et al. (2009) proposes a practical

time decay model with an exponential function for sampling and aging items in a data stream. Consequently, the above two methods use explicit weights and track the order of items to implement time bias. Surprisingly, our algorithm is superlinearly biased to time without explicitly weighting edges as shown in Chapter 3.

However, there are a few graph stream sampling algorithms such as in Ahmed et al. (2014) and Zhang et al. (2017) where both the nodes and edges are stored individually based on a probability. The reservoir size is based on the number of nodes while the number of edges keep changing. Zhang et al. (2017) deleted the nodes with minimum degree to maintain high degree nodes. This would require ordering nodes and shows no accountability of temporal distribution which is of high importance in real-time streams. Papagelis et al. (2013) introduced an online sampling technique to rank items in the neighbourhood of a given node. However the samples were not evaluated structurally.

Nevertheless, the previous works on static and stream sampling did not answer questions like, which sequential techniques preserve the structure of evolutionary streams? Which sampling techniques best preserve the community structures for weighted graphs? Which are most time-efficient? And which sampling methods are appropriate for generating approximate solutions for specific real-time queries? Which sampling techniques preserve the relevant structural properties of graph specific for an application? Nevertheless, no previous works focused on the evolution analysis of samples and the graphs with weighted and bi-directional edges. In this work, we answer the above questions regarding samples of real-time streams like call graphs of Telecommunication Networks (TN). We do this by plotting the measures used for comparing the structures and properties preserved by different samples generated in real-time. We also deal with the weighted and Bi-directional properties of a graph. Furthermore, we analyze the evolution of metrics over 31 days of Call graph streams. Enhancing the diversity of algorithms for sampling, we introduce an algorithm by modifying the existing reservoir Sampling Algorithm given by Vitter (1985).

4.3 Methodology

In order to find the best suitable sampling algorithms for a scale-free network like telecommunication call graphs (which follow the definition of evolving networks in Chapter 3, we analyzed the evolution of stream by generating snapshots of 31 samples at the end of each day, each cumulative from the beginning. For example, first sample for a stream of 1 day, second sample for the total stream of day 1 and day 2 etc.

Identifying community structures has its real-time applications like customer profiling, segmentation, targeted marketing, fraud detection etc., for telecom service providers. To answer queries like above, we need to answer the questions like, which samples preserve community structure? In this work, we have exploited the evolution of community structures for 31 samples by evaluating centrality measures such as Average Degree Centrality (ADC) and Average Weighted Degree Centrality (AWDC) of graphs; Indegree Centrality (deg^-) and Outdegree Centrality (deg^+) of nodes. Number of Communities in the sample graphs. Degree Distribution of the samples. We have also analyzed the network connectedness using metrics: Number of Connected Components (NC) and Average Component Size (ACS).

Then, we compare the time for computation by each of the techniques. Using the approx. time taken for one single pass over the stream of minimum size 7,845,201 and a maximum size of 15,440,707 calls streams. However, we also present the results to process a single edge, for edge-based methods.

For answering queries like, which samples are appropriate for finding top remunerative customers? Who are the top frequent callers? We use the generated samples to run real-time queries for specific applications, like finding top influential players using Eigen Vector Centrality (EVC).

4.4 Sampling Algorithms and Methods

In this section, we present the algorithms that we have used for generating samples. These algorithms can be implemented by using two types of methods. One is node-based sampling and the other is edge-based sampling method. By using these two methods, we also present results and observations to find an appropriate technique for generating real-time samples.

4.4.1 Space Saving Algorithm

The Space Saving Algorithm (Metwally et al., 2005) is the most approximate and efficient algorithm for finding the top frequent elements from a data stream. The algorithm maintains the partial interest of information as it monitors only a subset of elements from the stream. It maintains counters for every element in the sample and increments its count when the element re-occurs in the stream. If a new element is encountered in the stream, it is replaced with an element with the least counter value and its count is incremented. We employ this space efficient algorithm for generating samples of top K nodes and top K edges.

4.4.2 Reservoir Sampling

This is a well-known algorithm of Reservoir Sampling (RS), denoted as Algorithm R in Vitter (1985). In algorithm R, the author maintained a reservoir of elements with a predefined sample of size k. In the streaming scenario, initially, the reservoir is filled with the initial elements from the stream. Every element i after that is computed for the probability of being inserted. A random number r is generated between 1 and i. If $r \leq k$, the element at position k already in the sample is displaced with the new element at position i else the new one is discarded. As discussed in Section 3.6, this algorithm leads to samples with very old elements from the stream.

4.4.3 Biased Random Sampling

We have known the random sampling techniques where the odds of an element getting into the reservoir is based on a probability (they may enter the reservoir or not). In this section, we refer to the algorithm introduced in the Section 3.6, which is a random sampling algorithm where every element (in our case edges) in the stream definitely enters the reservoir. The size of the reservoir is fixed, therefore the new element replaces an old one already in the reservoir uniformly at random. We call it Biased Random Sampling (BRS).

Algorithm 1 Biased Random Sampling Algorithm

input : Unbounded *stream*
output: Realtime *sample* of size k

Filling the reservoir with first k items/objects;
for $i = 1$ **to** k **do**
  $sample[i] \leftarrow stream[i]$;
end
Inserting new items into the stream;
while *stream!= EOF* **do**
  $i = i + 1$;
  $pos \leftarrow Random(1,k)$;
  $sample[pos] \leftarrow stream[i]$;
end

Algorithm 1 represents BRS. As a general initial step, the reservoir is filled with the first items (edges) from the stream. Then, we do not compute the probability of later items, as every item definitely enters the reservoir. For replacing an item already in the sample, a random number is generated between 0 and the size of the reservoir. The element at the position of random number is replaced with the item from the stream. Here the probability of every item entering the reservoir is equal to 1, but the probability of the

item to stay in the reservoir diminishes as the stream progresses. Hence, this technique is biased towards new items from the stream as also seen in Chapter 3. As most of the real-time analytics require the latest information, this sampling method can effectively enhance their performance.

4.4.4 Node-Based Methods

Node-based methods in general, sample a set of nodes from the original graph. The resultant samples contain a set of vertices from the graph stream and having no connections between them. Therefore we need to acquire their corresponding edges as well. The samples posses only nodes and no structure. To evaluate this method, we have implemented the space saving algorithm by sampling top frequent nodes and call it as (SSN), detailed in Section 4.6.

4.4.5 Edge-Based Methods

As the name suggests, these samples are generated by selecting a subset of edges from the original graph. The resultant graph is a subgraph of the original graph with nodes and edges. We have conducted rigorous experiments implementing algorithms in section 4.4 using edge selection for sampling, i.e. RS, BRS and space saving algorithm by sampling edges (SSE). In Section 4.6, we discussed the experiments in detail.

4.5 Case Study

4.5.1 Telecommunication Networks

Sampling on graphs have been studied in different domains, with petty work in telecommunication networks. In this work, we quantify the applicability of sampling algorithms on this domain. Call Detail Records (CDR's) from TN are one of the largest and fastest data streams with stupendous mass of information hidden. We made use of such anonymous CDR stream of 386,492,749 calls made by 11,916,442 subscribers over 31 days. Spread across 24 hrs per day and gathered from geographically distributed sources.

4.5.2 Semantics of Call Graphs

We modeled telecommunication call graphs as nodes corresponding to callers and callees. The edges between them represent calls. These edges can be weighted using frequency of calls, duration of call, etc. The edges are bidirectional, corresponding to incoming and

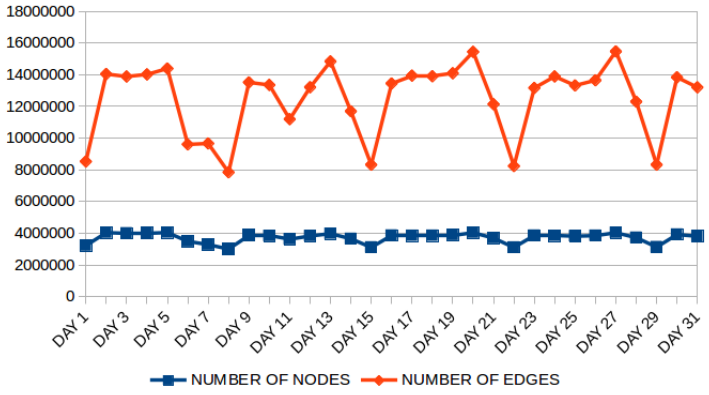


Figure 4.1: Evolution of nodes and edges in the call network stream

outgoing calls. The snapshots of number of nodes and edges per day stream are shown in Figure 4.1. The figure shows decreased call activity on Sundays compared to other days of the week.

4.6 Experimental Evaluation

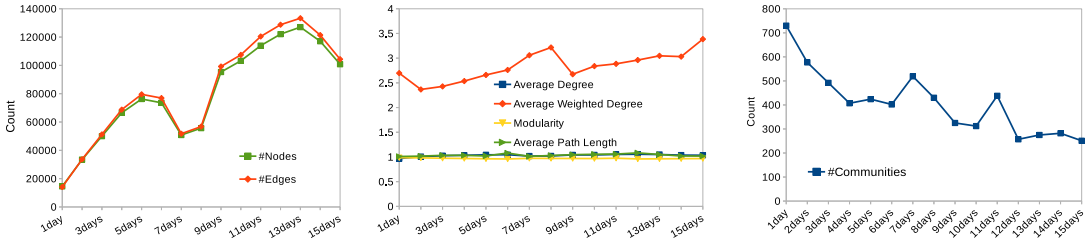
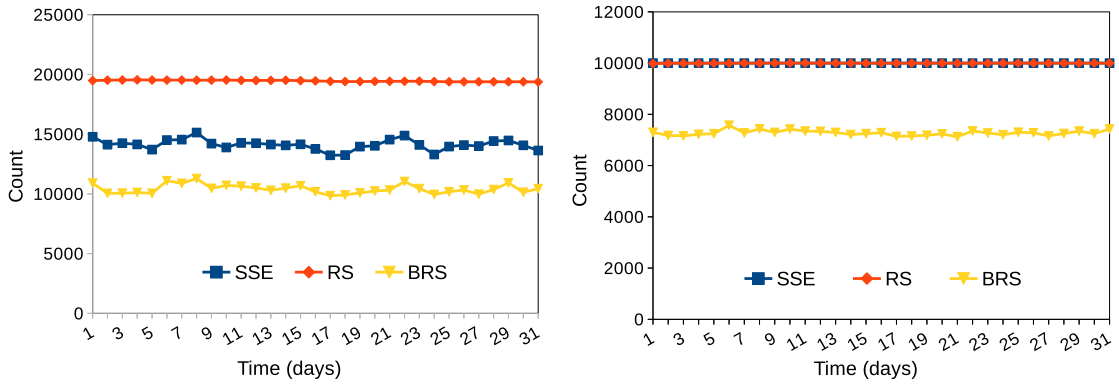


Figure 4.2: Evolution Analysis using SSN

To evaluate the node-based methods, we have used SSN to sample top frequent callers. We generated 15 sample snapshots of 10k nodes for 15 days of stream each from the beginning day 1. The samples generated using this method have only nodes and no structure; therefore, we also acquire the corresponding edges of nodes in the real-time. As the number of edges incident upon the sampled nodes increases, so are the adjacent nodes. As a result, we have a sub-graph with an increased number of nodes derived from the associated edges. The days when the chosen nodes are very sparsely connected, specifically the days of holidays and weekends, we get low number of nodes and edges. However, the time for computation of such methods also increases substantially with the added time for acquiring edges and their corresponding nodes. The evolution of the number of nodes, edges and properties of the graph are represented in Figure 4.2. We did not proceed with

the other algorithms for node-based method, because of space and time complexity mentioned above.



(a) Node evolution (b) Edge evolution

Figure 4.3: Number of nodes and edges

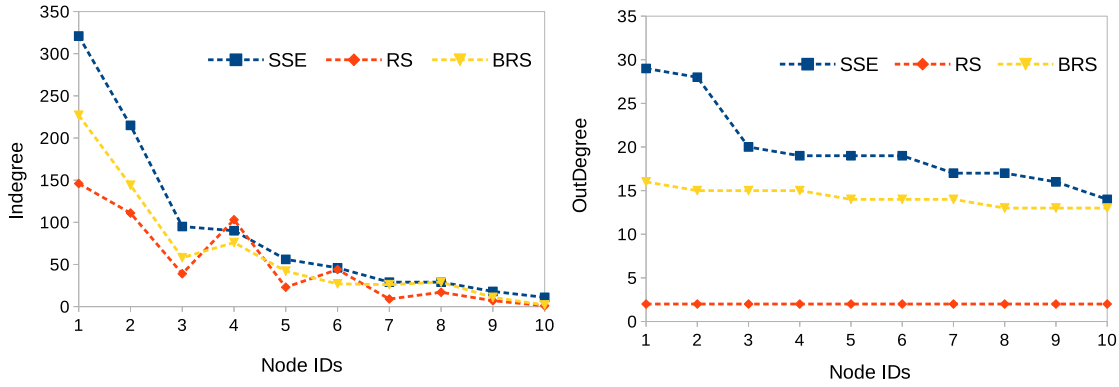
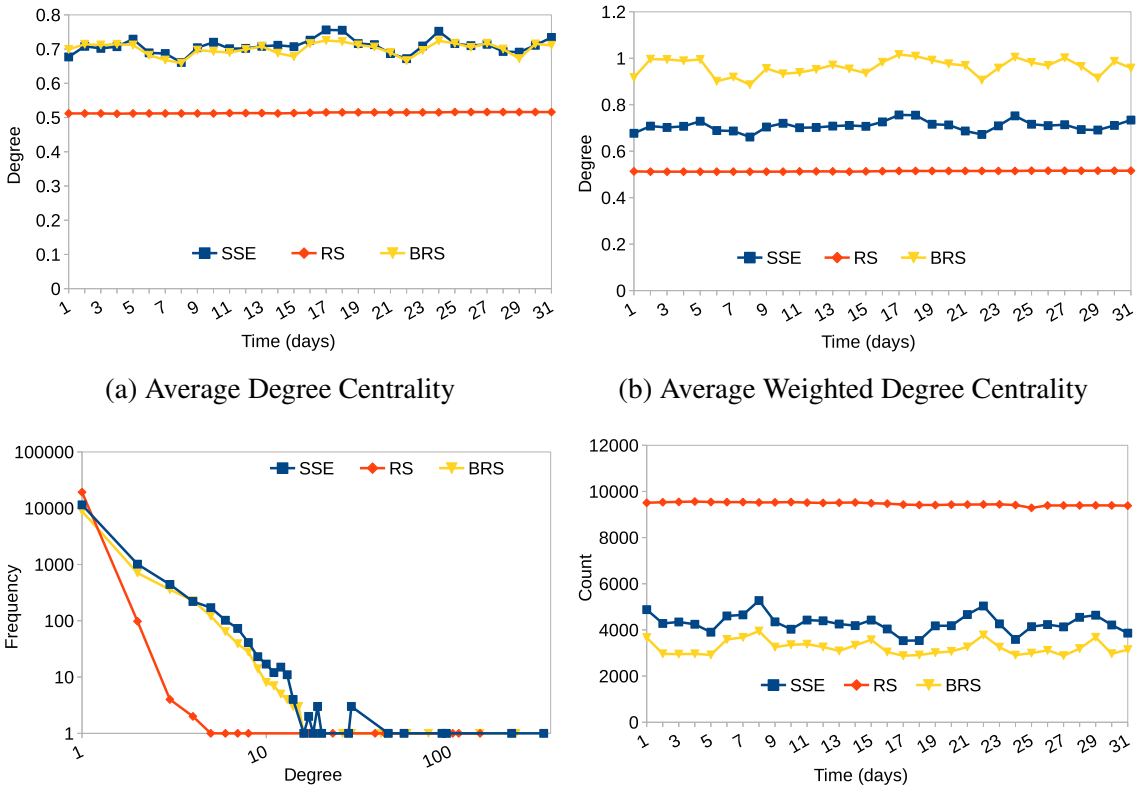


Figure 4.4: Indegree and Outdegree centralities

For evaluating edge-based methods we have generated 31 sample snapshots of size 10^4 edges from 31 days of call graph streams each from the beginning. This is done for three sampling algorithms using edge selection, i.e SSE, RS and BRS. The number of nodes and edges in the resultant subgraphs are shown in Figure 4.3. This figure shows the number of distinct edges in each sample. BRS samples contain less number of distinct edges and more number of repetitive edges/calls exhibiting calling behavior of callers. This indicates BRS is good at preserving edge weights based on frequency. Because in this work, we map the multi-graph of calls onto a weighted network, thus the weighted edges indicate frequency of calls between two nodes/callers. SSE samples show no weighted edges, as it selects the top frequent edges but do not store their frequency. One can see the subgraphs of RS have a negligible amount of weighted edges mapped from frequency of calls.

Figure 4.4 plots the degrees of top 6 indegree and outdegree nodes of three samples; we observe that SSE samples contain nodes with high deg^-'s and deg^+'s while RS samples have nodes with least deg^- and deg^+. When we compare both the deg^- and deg^+ we find that all the three samples are biased towards high indegree nodes and low outdegree nodes. This suggests the structure of original graph with a high number of incoming calls and less number of outgoing calls for each node on an average.



(a) Average Degree Centrality

(b) Average Weighted Degree Centrality

(c) Degree Distribution

(d) Number of Communities

Figure 4.5: Structural Evaluation

4.6.1 Community Structure

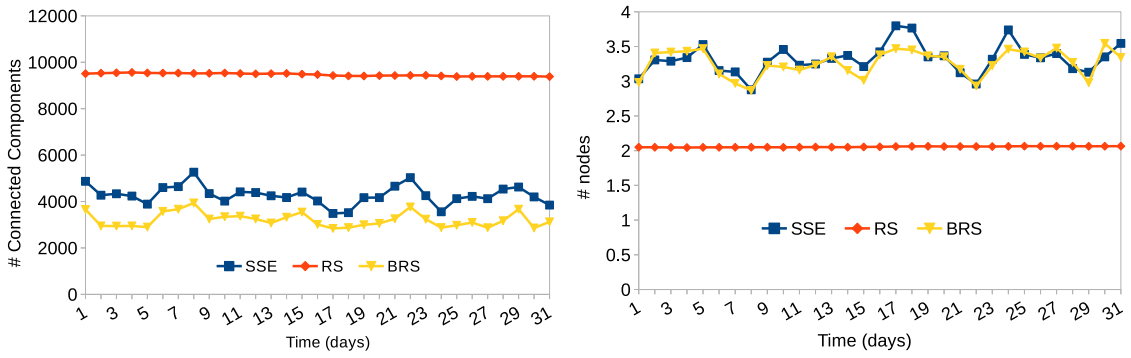
In our experiments, we have used degree centralities to evaluate the community structures in subgraphs. Figure 4.5(a) depicts the ADC of the three methods. RS shows the least average degree centrality than SSE and BRS which are almost the same. This suggests that RS is biased towards very low degree nodes, while SSE and BRS show similar degree centralities throughout the evolutionary stream. Figure 4.5(b) plots the AWDC of three samples. Comparing both the above referenced plots, we can observe that the ADC's of SSE and RS are similar to their respective AWDC's. While the AWDC of BRS is more

compared to its ADC. When ADC and AWDC are similar, the edges in the samples have no weights. When AWDC is greater than ADC the samples contain weighted edges. In both the Figures 4.5(a) and 4.5(b), SSE and BRS show low graph centralities on Sundays that relates to the original graph with low activity on Sundays, while RS curve shows no deviations.

Figure 4.5(c) plots the logarithmically binned degree distribution of nodes in the samples. We observe that 99.4% of nodes from the RS sample have degree 1. which indicates that a large number of nodes are sparsely connected, displaying the least community structure. Figure 4.5(d), depicts the number of communities in each sample, from which we notice that RS contains maximum number of communities. From both the above results, we infer that RS samples have a maximum number of communities with a minimum degree than SSE and BRS. The above results confer that RS shows least community structure with more number of nodes sparsely connected.

4.6.2 Component Structure

Figure 4.6(a) shows NC's in the three samples. We identify RS samples contain maximum NC's for all the days. BRS sample contain least NC's. Figure 4.6(b) plots the ACS of each sample, where SSE and BRS has similar ACS samples and greater compared to RS. From both the figures it is evident that RS samples exhibit least component structure with more NC's having least ACS compared to SSE and RS.



(a) Number of Components (b) Average Component Size

Figure 4.6: Component Structure

4.6.3 Time Complexity

To measure the time complexity of three algorithms, we have used two metrics, one is min time and the other is max time. Minimum time for one pass over the stream of 7845201

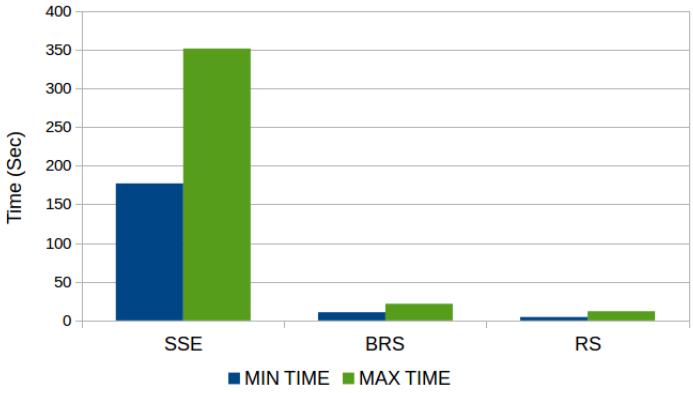


Figure 4.7: Time Complexity

edges and maximum time for one pass over the stream of 15468336 edges. Figure 4.7 plots the time for computation by three algorithms using the same hardware, from which it is apparent that the time for computation by SSE is maximum. RS takes the least time for computation while BRS with slightly more. However, the time for processing each edge as it arrives in real-time is much lower to approx. 15 ms to 20 ms for SSE, 0.5 to 1 ms for BRS and 0.3 ms to 0.4 ms for RS.

4.6.4 Running Real-Time Queries

Which samples are best suitable for running real-time queries and getting approximate answers to queries like, who are the top influential players in the network? To answer questions like these, we have used EVC to compute top influential nodes from the three samples generated by SSE, RS and BRS. All the three samples generate similar and accurate results for top 6 nodes, evident from Figure 4.8. As analyzed by Sarmento et al. (2015) call graphs exhibit a power law distribution with few nodes displaying high activity and a majority of nodes displaying least activity. The above results imply that the referenced sampling techniques capture highest activity nodes. However, the results of the three samples begin to differ as the number of top influential nodes increases.

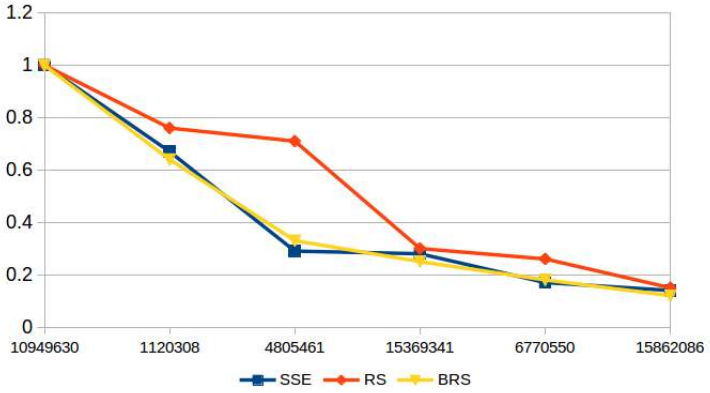


Figure 4.8: Eigen Vector Centrality

4.7 Empirical Observations

After analyzing the samples generated using four techniques discussed in section 4.4, we observe that: For generating structured subgraphs from node-based methods, we need to acquire the edges associated with nodes, which increases the time for computation. The evolution analysis of temporal samples generated using SSN shows a gradual increase in the size of samples.

For edge-based methods, we observe that RS is biased to nodes with low degree centralities and BRS and SSE nodes exhibit higher degree centralities compared to it. BRS best suits for measuring weighted centralities based on the frequency of edges. Hence, it is also suitable for running real-time queries for finding frequent items over the sample. BRS and SSE sample communities with high average degree centralities. That shows a better community structure when compared to RS. Therefore, BRS and SSE would be more suitable for applications analyzing cluster or community structure. SSE and BRS generates samples with better component structure compared to RS. RS and BRS has good performance with runtime compared to SSE. For using samples to run queries like top frequent items, SSE would be appropriate as it samples top frequent edges, while not considering other factors. All the three sampling algorithms give similar results for applications like finding most minimum number of top influential nodes using EVC.

4.8 Chapter Summary

In this chapter, we analyzed the properties of samples to compare the sampling techniques. We have used a real-time massive stream to evaluate the evolving samples, so that real-time queries and experiments can run over those samples. Different sampling algorithms have been evaluated in this work, based on structures, properties, time complexity and

applications. We have also evaluated weighted graph samples. We proposed an algorithm with the modification over reservoir sampling algorithm for networked streams. We have found that on an average of all the measures our proposed algorithm has exhibited better performance. Our results drive many observations related to the biases and appropriateness of sampling techniques. We have also analyzed the evolution of samples that can be used for evolution prediction in future works.

Figures 4.9, 4.10 and 4.11 visualize and interpret the properties of samples as proved in the above sections statistically. In Figure 4.9, we can see no dark edges representing weights but it still gives a good structure for communities. Figure 4.10, displays an RS sample with a very very poor community structure and looks highly biased to low degree nodes which supports the above results. Figure 4.11, shows BRS maintaining well-connected nodes and also with weights.

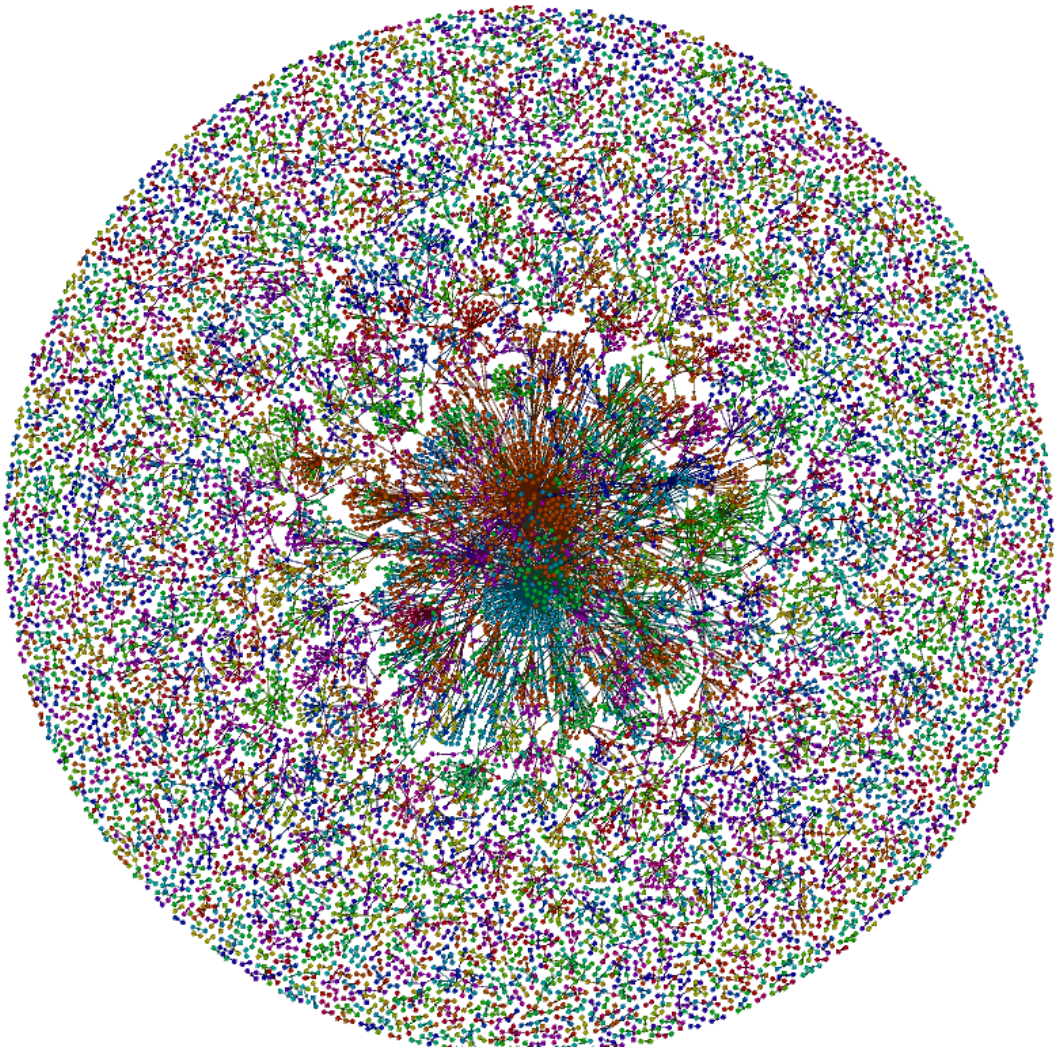


Figure 4.9: Pictorial representation of 10^4 top K edges sample at the end of 31 days stream using Space Saving (colors represent communities).

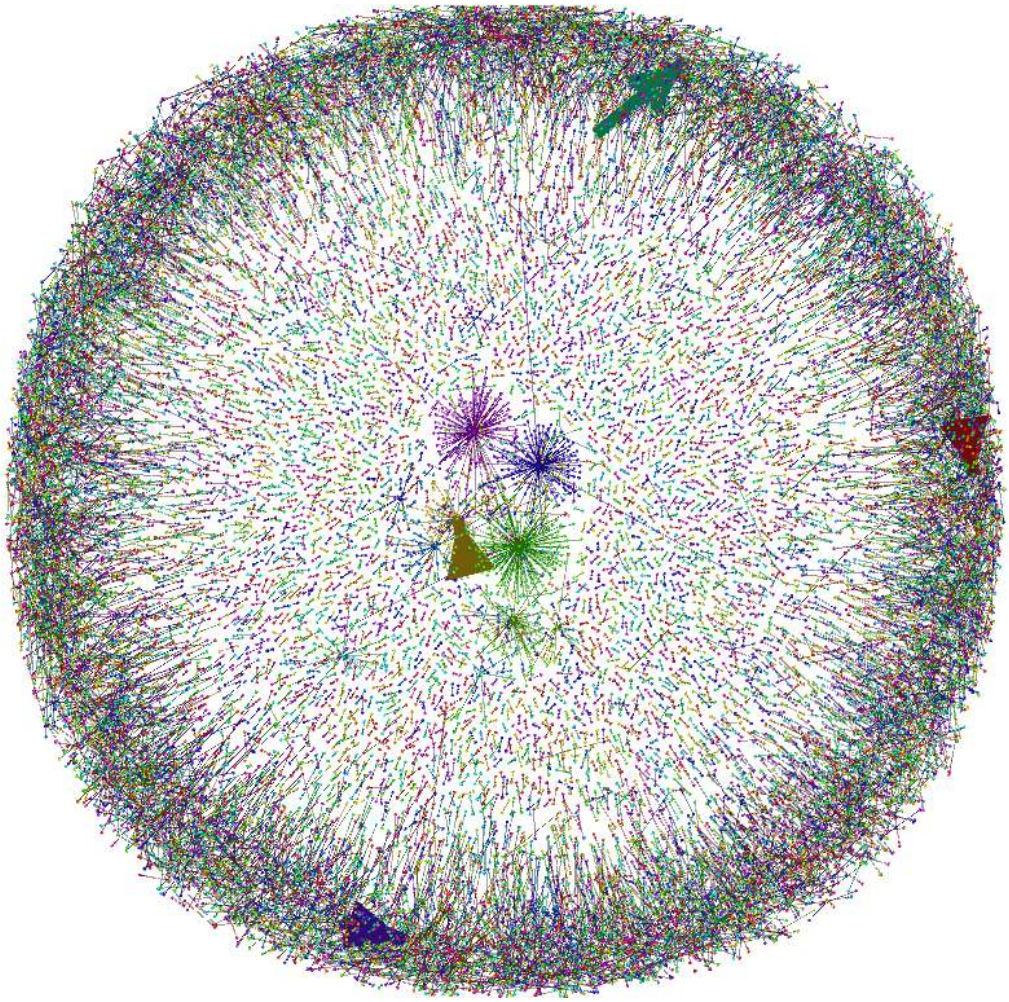


Figure 4.10: Streaming sample snapshot using Reservoir Sampling.

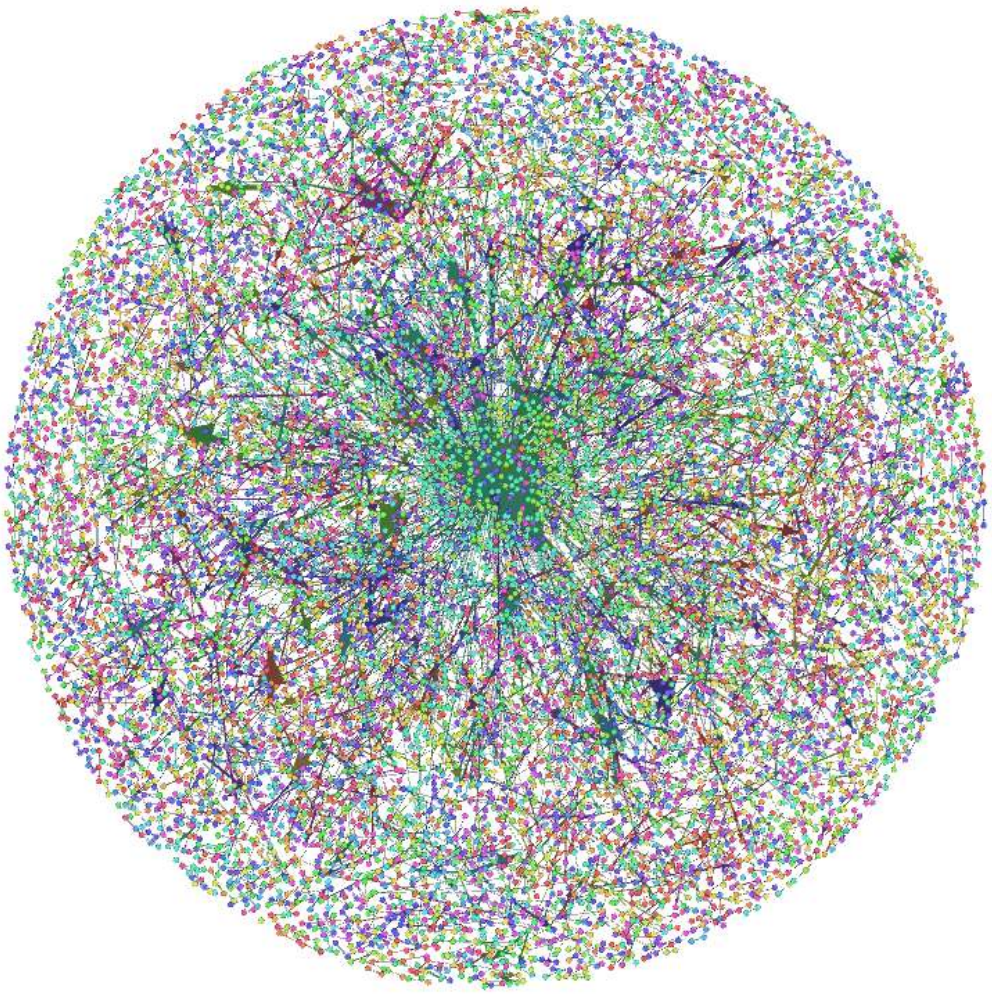


Figure 4.11: Streaming sample snapshot using Biased Random Sampling.

Chapter 5

Dynamic Sampling for Multi-graphs

Considering the avalanche of evolving data and the memory constraints, streaming networks' sampling has gained much attention in the recent decade. However, samples choosing data uniformly from the beginning to the end of a temporal stream are not very relevant for temporally evolving networks where recent activities are more important than the old events. Moreover, the relationships also change overtime. Recent interactions are evident to show the current status of relationships, nevertheless, some old stronger relations are also substantially significant. Considering the above issues, we propose a fast memoryless dynamic sampling mechanism for weighted or multi-graph high-speed streams. For this purpose, we use a forgetting function with two parameters that help introduce biases on the network, based on time and relationship strengths. Our experiments on real-world data sets show that our samples not only preserve the basic properties like degree distributions but also maintain the temporal distribution correlations. We also observe that our method generates samples with increased efficiency. It also outperforms current sampling algorithms in the area.

5.1 Chapter Overview

Handling and processing, high-velocity networked data generating from real-world applications is a current exigency. Dynamic sampling is an exemplary way to deal with the issues relating to massive evolving data, like answering approximate queries, running simulations, understanding and modeling true network structure, inadequate data, detecting events/changes in the network, etc. Apart from other applications, one of its major appeals lies in estimating the true network properties that cannot be handled in entirety (Ahmed et al., 2017) . Though sampling population is a statistically established area, it is not much explored in the current scenario of real-time dynamic networked data. A

comprehensive survey on sampling network streams can be found in Ahmed et al. (2014). Previous works in sampling streaming networks concentrated on preserving structural properties of a network at any time t. However, the temporal order of edges or their weights were not considered in building the samples, as it would be unfavorable to one pass and memory constraints. (Chen et al., 2013) studied the predictive value of links based on their age. They concluded that the young links are more informative than the older ones in predicting the formation of new links. Therefore, on recognizing the above issues, we tried to address the problem of dynamic sampling in multi-graphs or recurring link network streams, with the contributions stated below:

1. We propose one pass, memoryless, and bounded size stream sampling algorithm that maintains strong and stable structural relationships over time in a multi/weighted graph stream.

2. Apart from the traditional measures to evaluate stream sampling algorithms, we propose some interesting properties to look for in an on-line sample.

3. Our empirical results show that the samples from the proposed algorithm maintain temporal activity patterns, preserve the basic properties of network and perform better than the state of the art methods.

Rest of the chapter is organized as follows: In section 5.2, we presented a brief overview of the related state of the art. Problem definitions are given in section 5.3. Our method is outlined in subsection 5.3.1. Results in comparison with the true network are presented in subsection 5.4. Additionally, a comparative assessment with other methods is carried out in subsection 5.4.3. Finally, the summary and future works are summarized in section 5.5.

5.2 Related Work

Most of the stream sampling algorithms are variants of the classic *Reservoir Sampling* algorithm (Vitter, 1985), which is a probabilistic uniform random sampling method over a data stream. This algorithm runs into a limitation of old and stale items in the sample/reservoir, as the probability of new items entering into the reservoir decreases while the stream progresses. Aggarwal (2006) tried to introduce bias in the above algorithm by exponentially decreasing the probability of items in the reservoir to the probability of new items in the stream temporally. This bias was introduced to get an anytime sample with the results focusing more on the latest data while not completely ignoring the old data in

the stream, as is ignored in sliding windows (which is another alternative to sample the latest data). However, the approach was limited to data streams.

Ahmed et al. (2014) presented a simple edge stream sampling that uses the same approach as reservoir sampling (Vitter, 1985). In this case, a new edge enters into the reservoir if its hash value is within top-m minimum hash values, where m is the size of reservoir. However, the method did not ensue as an efficient representative sample in their work. Additionally, Ahmed et al. (2014) also proposed a *Partially-induced edge sampling algorithm* called (PIES). This algorithm works by storing nodes and edges probabilistically in their reservoirs while deleting the one already present at random as in the reservoir sampling. It maintained a fixed size reservoir of nodes while the reservoir size for edges varied based on nodes. CPIES, an update over PIES is given by Zhang et al. (2017). They modified the decremental module of PIES, by deleting the nodes from the reservoir with the minimal degree to produce a better cluster preserving structure. PIES had a selection bias to high degree nodes, which enhances in CPIES as it tends to delete low degree nodes. The above sampling techniques store nodes and then attain the corresponding edges, thereby increasing computational complexity. Further there is no information on how old the edges are or the importance of weights. Papagelis et al. (2013) proposed sampling algorithms that given a user in a social network quickly obtains a near-uniform random sample of nodes in its neighborhood using random walks. Another simple variant of reservoir sampling was given by Tabassum and Gama (2016b), where unlike the above algorithms, the incoming edges are not chosen based on a probability function instead all the streaming edges enter the reservoir and randomly replace the edges already in it.

We use a similar concept as Aggarwal (2006) from data streams, but on a temporal stream of edges in a dynamic network with a different approach. Instead of decreasing the probability of edges in the reservoir exponentially, we use the exponential function over the weights of edges in the network. Thus the weights are not imposed by the algorithm but are natural weights acquired from the network. The method works by sending every new edge into the reservoir/sample but the probability it stays in depends on the time step t when it occurred or reoccurred and its weight at t. A similar approach is used over ego network streams by Tabassum and Gama (2016c), which is detailed in Chapter 6.

5.3 Problem Definition

Here we represent an evolving network as a stream of recurring edges $\{e_1, e_2, e_1, e_3, ...\} \in E$ generating from a graph stream G. Every edge $e_i = (u, v,)$ is composed of a pair of

vertices's from V. For brevity we denote an e_i as e. A time-stamp t indicates the time of occurrence of e. We assume that the edges are streaming in the order of time-stamps.

Definition 5.3.1 (Multi-graph)**.** In our definition of a temporal multi-graph with recurring links, an edge e can recur randomly in G at various time-stamps t. Another variable τ is a discrete time-step/time-interval with granularity defined by the user. Initially when $\tau = 0$ we have $G = \emptyset$. A (true) network G at $\tau = n$ is given as all the network from τ_0 to τ_n. At every τ, the weight w of an edge e is computed as

$$w(e, \tau) = \#\text{ of occurrences of any e in } \tau \tag{5.1}$$

The weights of an edge e in any time step τ i.e, $w_e(\tau)$ are also incremented in a streaming way sequentially. For weighted graphs w is the weight of an e in the time-interval τ.

Definition 5.3.2 (Edge vector stream)**.** Every edge e in G can occur in multiple τ's. Therefore every edge is a temporal vector **a** of the state of its presence (with a weight $w_e(\tau)$) or absence in every τ. $M(\tau)$ is the number of unique edges in $G(\tau)$ at any τ.

Definition 5.3.3 (Edge Stability)**.** We define edge stability $s_e(\tau)$ as the property of an edge e in a multi-graph stream G at every τ is given as

$$s_e(\tau) = \frac{\#\text{ of elapsed } \tau\text{'s where e is present in } \mathbf{a}}{\text{total number of } \tau\text{'s elapsed}} \tag{5.2}$$

Stability lies in the range of [0,1]. If an edge e occurs in every time-step τ in **a**, then the edge is considered most stable with the stability 1. For example, in a co-authorship network considering τ as one year. If two authors collaborate with a scientific paper for 2 years in an observational period of 4 years then their edge stability is equal to 0.5 at the end of 4 years. Therefore most stable edges in a graph stream characterize a strong relationship.

Definition 5.3.4 (Problem statement)**.** Given an evolving temporal stream $G(\tau)$, our sampling algorithm aims to produce a sample $\hat{G}(\tau)$ that preserves the importance of edges based on time τ, their strengths w and stability s.

5.3.1 Sampling with a bias to latest and stable edges (SBias)

For generating and biasing the sample $\hat{G}$ at any τ, towards the recent behavior of network, we use a forgetting function from the class of memoryless exponential functions. This

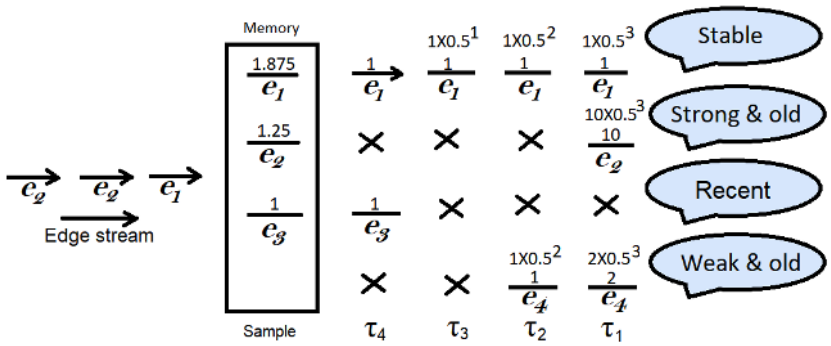


Figure 5.1: Pictorial representation of SBias sample at τ_4 with $\alpha = 0.4$ and $\theta = 0.5$.

function is applied on every e from M edge vector streams, after every τ to get $\hat{w}(e,\tau)$, which is the weight of an edge in $\hat{G}(\tau)$. The function is defined as follows:

$$\hat{w}(e,\tau) = w(e,\tau) + (1-\alpha)\hat{w}(e,\tau-1) \tag{5.3}$$

The parameter α defines the bias rate and typically lies between 0 and 1 inclusive. In general, this parameter α is chosen in an application specific way. We use α according to the sample size we need. When $\alpha = 0$ the function returns the true network G with original weights.

$$\hat{w}_e(\tau) = w_e(\tau) + (1-\alpha)\hat{w}_e(\tau-1) \tag{5.4}$$

For an e in E with currying, the update function at every τ as a one pass incremental stream can be given as in equation 5.4. When α is 1, we forget all of the previous edge occurrences of an e and only maintain information from the latest τ. If e does not recur in **a** then $\hat{w}$ is monotonically decreasing and is removed from the sample if it is less than a threshold θ. If e appears again in G at a time-step $\tau+i$ it gets added to $\hat{G}(\tau)$ again. All the edges in the sample at a given time τ can be represented as:

$$\hat{w}_\tau(e) = w_\tau(e) + (1-\alpha)\hat{w}_{\tau-1}(e) \tag{5.5}$$

Where θ lies between 0 and the highest $\hat{w}(e)$ in the network at a time τ, because we did not normalize the weights which would cost extra computation and knowing all $w(e)$ in advance (not favorable for one pass). After each τ, the edges in the sample with $\hat{w} < \theta$ are deleted from it, retaining stronger and stable edges from M but we still have the weak edges from the latest τ because of the recent bias. Therefore the sample size decreases to $\hat{M}$ at $\tau+1$. A pictorial representation of our method is shown in figure 5.3.1. Weights of edges are updated with their frequency in the stream. A cross indicates the edge was

absent in that τ.

A higher value of α retains most recent edges in $\hat{G}(\tau)$ exponentially in the order of τ, while a lower value tends to include older edges also. Likewise, a higher value of θ would retain more strong and stable edges in $\hat{G}(\tau)$ and a lower value retains less strong and stable ones. Therefore, we can tune the parameter according to the requirement.

The space complexity of SBias is $O(\hat{M})$, where $\hat{M}$ is the edge-set for output at any τ, which is a minimum space requirement as an algorithm cannot require less space than output. In the worst case, when we need no forgetting and the sample is equal to the original network, the space complexity is $O(E)$ where $\hat{M} = |E|$. In other words, the space complexity increases with the sample size. We use hash map for storing all the edges and updating at every at τ, therefore the time complexity is $O(1)$ and in the worst case is $O(E)$. The tightest bound is obtained when $\alpha = 1$, the sample is only the network from the latest time step τ. Therefore lower bound of the sample $\hat{G}(\tau)$ is the individual network at τ not including the network from other $\tau's$, which is minimal compared to whole network until τ.

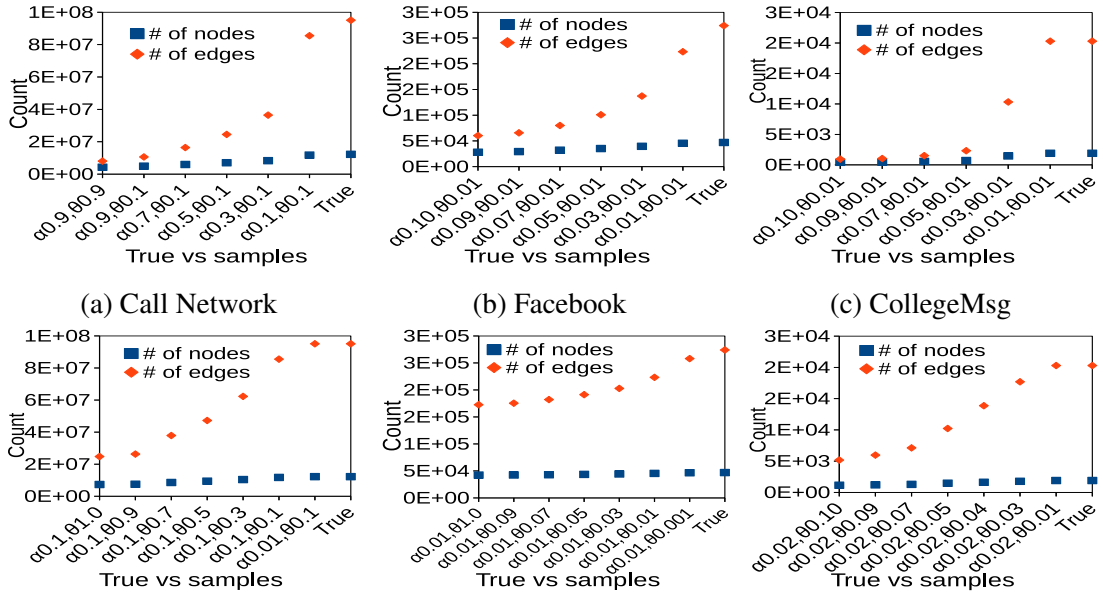
Table 5.1: Networks' properties

Data sets	$\|E\|$	$\|V\|$	unique $\|E\|$	$\|\tau\|$	components	Avg. deg	Avg. wt deg	Avg. Stability
Call Network	389994643	12213391	95090672	30	28260	15.6	63.9	0.072
Face book	876993	46952	274086	1506	1981	37.3	71.2	0.002
College Msg	59835	1899	20296	193	4	10.7	31.4	0.009

5.4 Experimental Evaluation

We followed two approaches for evaluating our method. Firstly we evaluated the representative properties of the samples against the original network. Then we compared our samples with the samples from Reservoir Sampling Vitter (1985) and Biased Random Sampling Tabassum and Gama (2016b).

Here we give some handy notations to help understand experiments. The true network is the network G from definition 5.3.2. The size of the network $M(\tau)$ and size of the



(a) Call Network (b) Facebook (c) CollegeMsg

(d) Call Network (e) Facebook (f) CollegeMsg

Figure 5.2: Parameters α and θ influencing the size of network.

sample $\hat{M}(\tau)$ is given by the number of unique e in them. The sample% is the percentage of edges from G at the end of the observed stream.

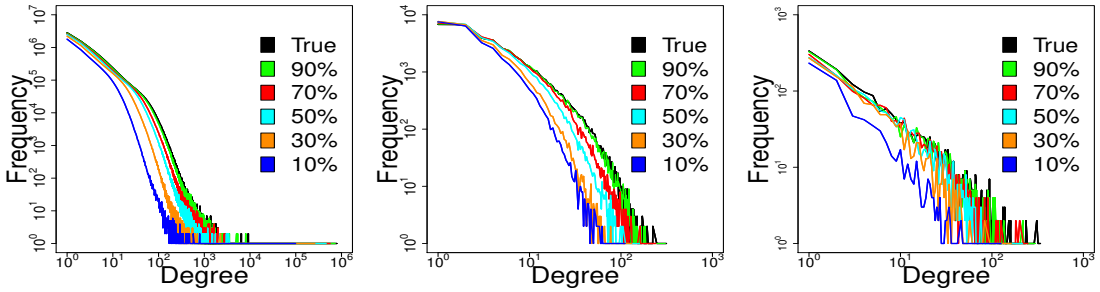
5.4.1 Data sets

Call Network: We use a massive anonymized CDR (Call Detail Records) data provided by a service provider. The average speed of data is 10 to 280 calls per second around mid-night and mid-day. Calls in the data are associated with timestamps when the call was initiated.

Facebook Wall Posts: This data set is a subset of users posts on other users' wall on Facebook which is obtained from KONECT[1] networks. The detailed description of data is also available at Viswanath et al. (2009a). The data spans from 2004 to 2009. The data is temporally very sparse (few months have just one edge).

CollegeMsg: This data set is comprised of private messages sent on an online social network at the University of California, Irvine Panzarasa et al. (2009), obtained from SNAP data sets[2].

[1]Data available at http://konect.uni-koblenz.de/networks/facebook-wosn-wall
[2]Data available at https://snap.stanford.edu/data/CollegeMsg.html



(a) Call Network (b) Facebook (c) CollegeMsg

Figure 5.3: Degree distributions, true network vs samples.

5.4.2 Comparative assessment with the true network

5.4.2.1 Parameter Sensitivity

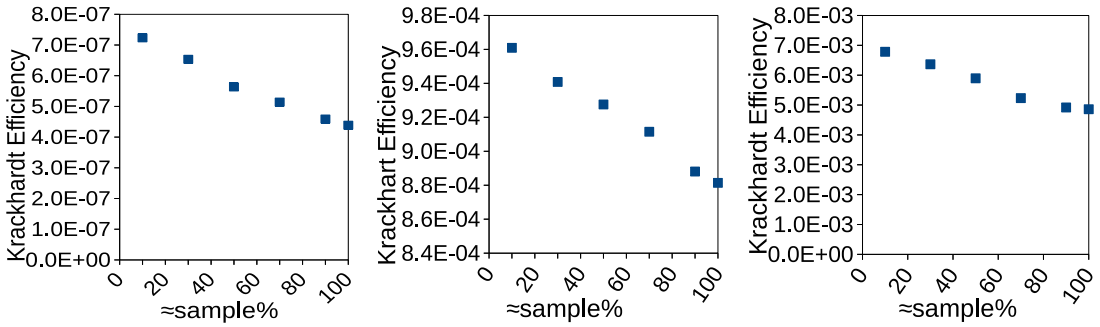
To evaluate the sensitivity of parameters empirically, we did two experiments, firstly varying α by fixing θ and secondly varying θ by fixing α with equal intervals. We see both the outcomes in figure 5.2 are similar but the algorithm is more sensitive to the change in α than θ. In all the experiments, we see that as we increase the parameter values the number of edges is getting closer to the number of nodes which shows a structure closer to the spanning tree; we will quantify this using Krackhardt efficiency in the following subsection. Note the number of nodes is also decreasing. We make use of the parameter sensitivity property to generate samples of different sizes.

5.4.2.2 Degree Distribution

For every degree d, we count the number of nodes with degree d. In all the data sets (figure 5.3) we see that the samples follow a similar distribution as in the true network. As the stream length and stability varies in three data sets we had to use different parametric values to get equal percentage samples.

5.4.2.3 Krackhardt Efficiency

Krackhardt Efficiency Krackhardt (2014) was defined by Krackhardt to measure the extent to which a graph is an out-tree or a spanning-tree (if directions in the network are not considered). It is a measure of non-redundancy in multiple components of a graph or multiple weak components of a digraph. The graph is said to be highly efficient if there



(a) Call Network (b) Facebook (c) CollegeMsg

Figure 5.4: Krackhardt efficiency, true network vs samples.

are $(N_i - 1)$ number of links between N_i number of nodes in every component C_i of a graph C. The measure of efficiency is calculated with the equation below.

$$1 - \frac{E(C) - \sum_{i=1}^{n}(N_i - 1)}{\sum_{i=1}^{n}(N_i(N_i - 1) - (N_i - 1)} \tag{5.6}$$

When there are multiple components in a graph, the above equation calculates the efficiency of components in a spanning forest. Figure 5.4 shows that the samples have components with an increased efficiency, that is getting close to 1 as the sample size decreases. As our method removes the weaker edges from the network we retain a sample with less redundancy and high weighted edges, approaching a minimum spanning forest. This is a nice property as many path-finding algorithms internally build spanning trees, that increases their complexity. The true network is denoted as 100% in figure 5.4.[3]

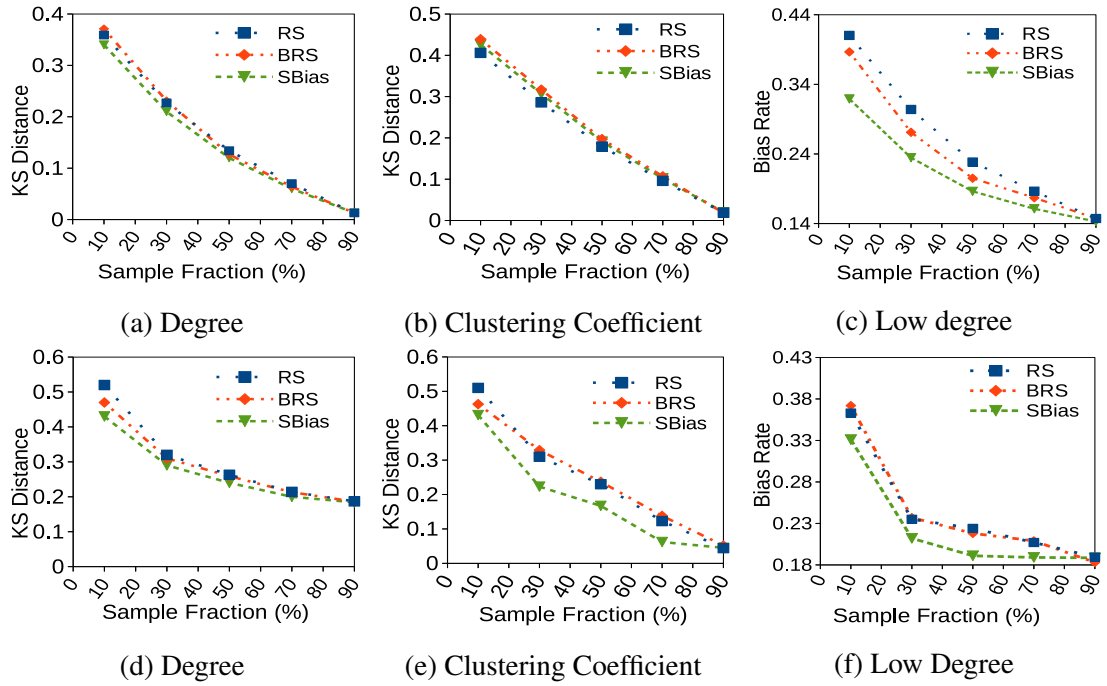
5.4.3 Comparative assessment with other methods

In the following subsection, we demonstrate two state of the art sampling algorithms that were applied on temporal multi-graph network streams in Tabassum and Gama (2016b). To our knowledge, these are the most related and recent approaches to our problem.

5.4.3.1 Reservoir Sampling for edge stream (RS)

Reservoir sampling was the algorithm given by Vitter (1985) for sampling data streams. Here we employ it on a stream of edges from evolving networks to get dynamic sample at any point in time. The algorithm is detailed in section of chapter 4. As it is a multi-graph stream the sample can contain multiple edges between two nodes. Therefore, the frequency of these edges is mapped as their weight considering it as unique edge.

[3]Note: The values of y-axis in the figure 5.4 should be increased by a constant c (y-axis=y-axis+c). For call networks c=0.999999, facebook c=0.999, and CollegeMsg c=0.99



(a) Degree (b) Clustering Coefficient (c) Low degree

(d) Degree (e) Clustering Coefficient (f) Low Degree

Figure 5.5: KS-Distance of distributions and lowest degree bias rate in Facebook (a-c) and CollegeMsg (d-f).

5.4.3.2 Biased Random Sampling for edge stream (BRS)

This algorithm is proposed as a simple variant of RS in Tabassum and Gama (2016b). We presented the algorithm in chapter 4. Unique edges with weights are maintained in the same way as in the above samples.

Table 5.2: Correlation of temporal distributions with aggregated network

True vs sample%		90	70	50	30	10
Call Network	SBias	0.996	0.957	0.917	0.891	0.145
Facebook	RS	0.995	0.987	0.960	0.918	0.846
	BRS	0.996	0.988	0.967	0.925	0.850
	SBias	**0.997**	**0.993**	**0.984**	**0.975**	**0.967**
CollegeMsg	RS	0.987	0.953	0.915	**0.866**	**0.745**
	BRS	0.986	0.953	0.895	0.792	0.442
	SBias	**0.997**	**0.974**	**0.921**	0.783	0.156

Though the above two algorithms are known to be fast and simple, we were only able to run them on two data sets from three. The algorithms were not able to handle the size of call network's samples.

Table 5.3: Correlation of temporal distributions with edges per τ

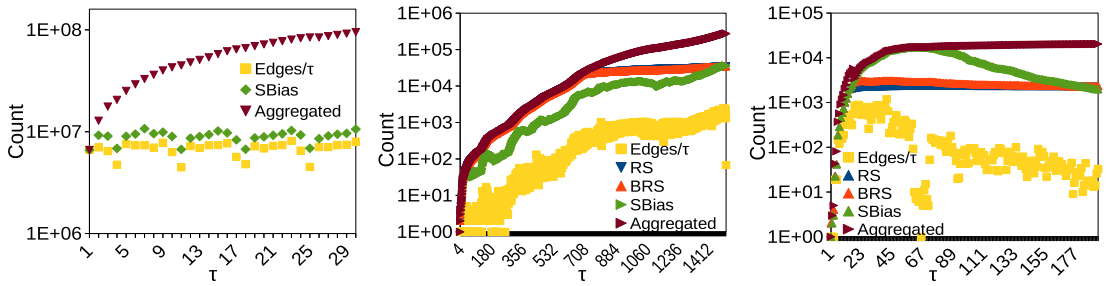
Edges seprated/τ vs sample%		1	3	5	10	30	50	70	90
Facebook	RS	0.65	0.75	0.79	0.86	0.89	0.91	0.90	0.91
	BRS	0.64	0.75	0.80	0.87	0.90	0.90	0.90	0.91
	SBias	**0.99**	**0.96**	**0.96**	**0.96**	**0.95**	**0.94**	**0.92**	**0.91**
CollegeMsg	RS	0.08	0.10	0.06	0.008	-0.17	-0.27	-0.39	**-0.51**
	BRS	0.46	0.52	**0.49**	**0.33**	**-0.08**	**-0.25**	**-0.38**	-0.52
	SBias	**0.82**	**0.62**	0.36	0.004	-0.36	-0.48	-0.52	-0.55

5.4.3.3 Distance Measure

We used two-sided Kalmogorov-Smirnov D-statistics to measure the distance of sample distributions of different methods with the true distribution. The distance in degree distributions is given in figure 5.5. We observe that SBias gives better distributions of cluster structures.

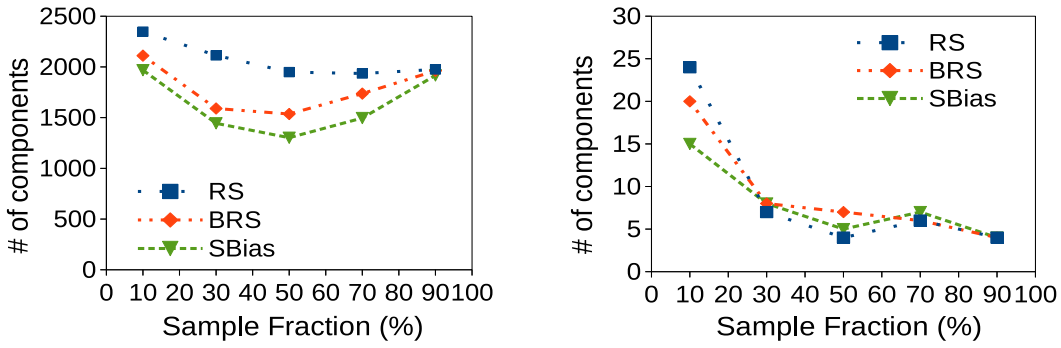
5.4.3.4 Low Degree Bias Rate

Usually, samples from large networks are prone to low degree biases. Therefore, we examined the bias to the nodes with degree one, in the samples compared to the true network, which is shown figure 5.5(c) and 5.5(f). Bias rate shows the percent of one-degree nodes in the network to other degree nodes. Y-axis in the figures 5.5(c) and 5.5(f) begins with the bias rate of the true network to show the comparative increase of bias in the samples. As the sample size decreases RS and BRS get more biased to low degree nodes.



(a) Call Network (b) Facebook (c) CollegeMsg

Figure 5.6: Illustration of temporal distributions of dynamic samples of 10% in comparison to the # of unique edges separated per τ of the true network. Y-axis is plotted in log scale to clearly show the patterns in low scale together with high scale data.



(a) Facebook (b) CollegeMsg

Figure 5.7: Number of components in samples by three algorithms at the end of stream.

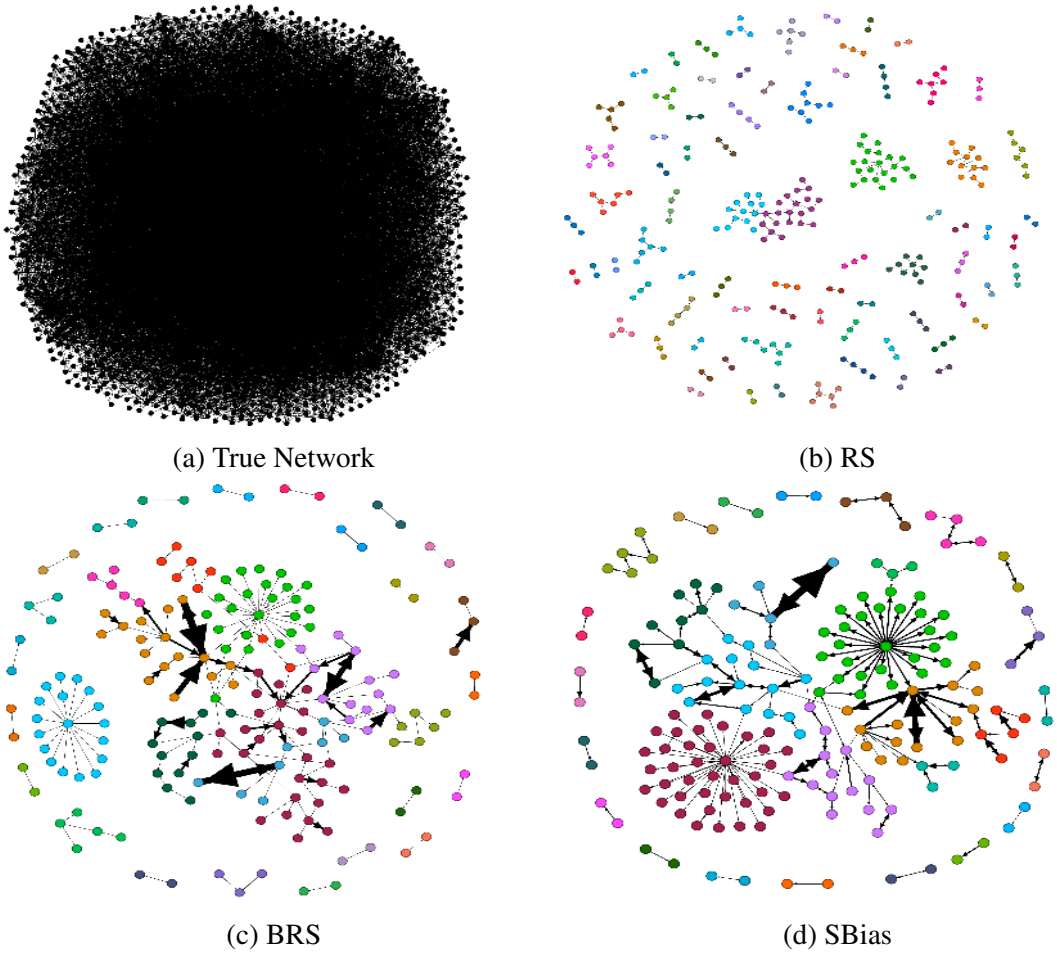
5.4.3.5 Temporal Correlation

The temporal distribution of network size (# of unique edges per τ) exhibited by SBias for larger samples in the observational period is very similar to the true network distribution. The Pearson Correlation coefficients are shown in table 5.2. The temporal distributions of the smaller samples are quite close to the distribution of individual edges per τ of the original network. Note, True network is the aggregated network, while edges/τ is individual network separated per τ. A 90% sample is more correlated to true network temporal distribution, while a 1% sample is more correlated to individual edges per τ temporal distribution. This property of our algorithm can help in identifying and predicting temporal trends, changes and behaviors.

The Pearson Correlation between temporal distributions from the beginning to the end of the stream for different sample sizes is given in table 5.3. An illustration of temporal patterns preserved by 10% sample is given in figure 5.6. RS and BRS do not follow the pattern of true network variations for both the data sets, as soon as the reservoir gets filled up they go flat (few deviations are seen in BRS) by maintaining the fixed sample size. When the true distribution behaves as a straight line they get a better score. They perform better on CollegeMsg network than Facebook because the size of the network is not growing exponentially over time as most other real-world networks do. Where as SBias follows the true network pattern edges/τ. Though it generates variable size samples the distance between the true network and SBias temporal distribution is constant for small samples and increases with the sample size.

5.4.3.6 K-Components

Components are the disconnected clusters in a graph. Graphs from the real-world have many components (at least due to the limited availability of temporal data). Therefore, we



(a) True Network (b) RS

(c) BRS (d) SBias

Figure 5.8: Snapshot at the end of observed stream (CollegeMsg) of true network and using sampling algorithms (sample fraction 1%)

compare the number of components from the samples of same size. Lesser the number of components is better. Bias to low degree nodes increases the number of components. We see that SBias gives better component structure compared to RS and BRS (figure 5.7).

5.4.3.7 Visualization

To illustrate the properties discussed above, we pictorially represented (figure 5.8) the sample snapshots at the end of observed stream from collegeMsg data. Some core properties and nodes are apparent in the samples which are otherwise difficult to visualize in the true network. This demonstrates a major application of sampling. However the efficiency of results depends on the representativeness of sample. From figure 5.8 we observe that SBias gives a better structure with less number of components and isolated edges, nearing true network structure which is shown in empirical results as having less bias to low degree nodes. We can also see many strong edges in the figure 5.8(d) than RS

and BRS, which is the property of SBias. Some descriptive properties of the network and samples are listed in the table 5.4.

Table 5.4: Measures of networks from figure 5.8. SBias exhibiting sample properties close to the true network structure

Properties	V	E	Avg. Deg	Density	Components
True Network	1899	20296	10.688	0.008	16
RS	296	225	0.76	0.005	76
BRS	189	225	1.196	0.006	25
SBias	182	225	1.236	0.007	21

5.5 Chapter Summary

We proposed a one pass, memoryless sampling algorithm for multi/weighted network streams to focus on the relationships and time variable. We empirically demonstrate that SBias preserves the degree distribution without biases and decreases redundancy in samples. It also maintains the patterns in temporal streams. In comparison to reservoir sampling and biased random sampling, we show that SBias preserves well the component structure. It is also able to handle the large network samples we generated from call network, which the other two algorithms were not able to process. We also proposed some measures for evaluating samples, like Krackhardt efficiency, temporal distributions, and stability of edges.

For the future works, we would like to analyze the performance theoretically and mathematically and understand the limitations of SBias. We intend to compare our method to other state of the art methods for sampling streams by upgrading them to handle multi or weighted graphs, which will help us explore new properties. An interesting direction is also to estimate the properties of true network from the samples of SBias. Another appealing direction we would like to pursue is using SBias in an application specific way, for example, link prediction, as it gives bias to recent links and their strengths and, change detection, as it preserves temporal distributions.

Chapter 6

Ego Networks Evolution Analysis

With the realization of networks in many of the real-world domains, research work in network science has gained much attention nowadays. The interactions between objects in the data are exploited to gain insights into the real-world connections. One of the notions is to analyze how these networks grow and evolve. Most of the works rely upon the socio-centric networks. The socio-centric network comprises of several ego networks. How these ego networks evolve greatly influences the structure of the network. In this work, we have analyzed the evolution of ego networks from a massive call network stream by using an extensive list of graph metrics. By doing this, we studied the evolution of structural properties of graphs and related them with real-world user behaviors. We also proved the densification power law over the temporal call ego networks. Many of the evolving networks obey the densification power law and the number of edges increases as a function of time. Therefore, we discuss a sequential sampling method with a forgetting factor to sample the evolving ego network stream. This method captures the most active and recent nodes from the network while preserving the tie strengths between them and maintaining the density of the graph and decreasing redundancy.

6.1 Chapter Overview

Enormous streams of graphs are generated from a number of real-time applications, some times at a speed of more than millions of nodes and billions of edges per day. These social streams provide an abstraction of interactions between real-world social entities or individuals. Studying the structural properties of these streams enables powerful insights and extrapolations of the real-world. Space and time complexity is one of the challenging issues related to analyzing these streams. Networks representing real-world social structures are usually temporal and evolving. The rapidly changing and evolving structure of

these graphs, calls for an exigency of latest and up to date results. Processing the real-time network stream as it arrives, is one of the best solutions for the above problem. Therefore, we employ the stream processing approach to process enormous data. Some of the social network analysis methods that can be applied over streams of graphs are given in Sarmento et al. (2015). Furthermore, we use a streaming ego network approach over a telecommunications' call graph stream of temporal edge/calls' as in Tabassum and Gama (2016b).

An ego network is based on the relationships of a single node called "ego" with the other nodes in a social network. An ego can represent an individual, entity, object or organization. All the other nodes related to the ego in the network are called alters. An ego network maps the relationships of ego with alters and also between themselves. In the recent work of Epasto et al. (2015) by Google.com, the authors argue that it is possible to address important graph mining tasks by analyzing the ego-nets of a social network and performing independent computations on them. The studies made by Everett and Borgatti (2005) indicate that the local ego betweenness is highly correlated with the betweenness of the actor in the complete network. Wellman (1996) describes an ego network as a personal network. The author explains that the importance of local ties becomes apparent by redefining the composition of personal community networks in terms of the number of contacts (interactions) that egos have with the active members of the networks instead of the traditional procedure of counting the number of ties (relationships). In this work, we analyze the evolution of ego networks by using a bunch of social network analysis metrics.

We also discuss the growth pattern of our ego networks. Leskovec et al. (2005), discussed how large graphs evolve over time. They stated a densification power law, which is followed by these networks. In our work, we test the densification power law over the temporal ego networks of call graph stream and observe that it obeys the densification power law and follows the similar properties of large graphs. We also consider the properties of real-world graphs such as diameter, path length etc.

We observe the call ego networks satisfy the densification power law and the number of edges grows superlinearly to the number of nodes; with this, the evolving graphs can get humongous in no time. There are no sampling strategies discussed so far on real-time streaming graphs preserving the tie strengths between nodes in the network until in Tabassum and Gama (2018), which is discussed in Chapter 5. Nevertheless, there are also no sampling techniques designed for ego networks to preserve tie strengths. Now the obvious question is, how do we capture the ego network of an evolving multi-graph stream over time with the least possible edges while preserving the structure, properties and efficiency of an ego network? For which, we proposed a streaming ego network

sampling method using a forgetting factor in Tabassum and Gama (2016d). The proposed method is suitable for dynamically evolving multi-graphs. We use this method over a real-world temporal stream of edges/calls to generate a sample stream in real-time. Our results show that the proposed method preserves tie strengths in the networks. We also show that our method decreases redundancy in the network while preserving the importance of ego. We measure the importance and efficiency of network using some socio-metrics. We evaluate our method by comparing the samples generated by varying parametric values, with the original ego network. The proposed method can also be implemented over a socio-centric network.

The following chapter is organized as follows: In Section 2, we discuss some related works. In Section 3, we described our call network data and the metrics we used in our experiments in Section 4. We proved the densification power law for our evolving ego networks in Section 5. In Section 6, we analyzed the properties and structure of evolving call ego network. Further, in section 7, we proposed a sampling method for ego network multi graph streams with a forgetting factor. Section 8 and 9, we evaluated the above method by comparing the samples with the original network.

6.2 Related Work

The concept of ego networks was discussed by L.C.Freeman in Freeman (1982), where he described an ego network as a social network, built around a particular social unit called ego. Wellman (1996) discusses the importance of local ties in personal networks. Burt (2009) studied the effects, gaps and relationships between the neighborhood of a node, referring them as structural holes. He also introduced metrics to evaluate an efficient-effective network that strives to optimize structural holes in order to maximize information benefits.

Most of the research work in this field is carried out by analyzing the structure and growth pattern of evolving socio-centric networks and evolutionary nature of socio-centric graphs (Albert et al., 1999; Broder et al., 2000; Milgram, 1967a; Watts and Strogatz, 1998; Newman, 2003a; Leskovec et al., 2005). Nevertheless, there are few works which studied the structure of ego networks (Burt, 2009; Hanneman and Riddle, 2005; Freeman, 1982; Tabassum and Gama, 2016d,e). To the best of our knowledge, this is the first work about analyzing the evolution of ego networks. We would analyze the evolution of the ego network for 31 days using an extensive list of graph level and node level metrics.

Ma et al. (2010) proposed an ego-centric network sampling approach for viral marketing applications. The authors employed a variation of forest fire algorithm for sampling

ego network. They compared the degree and clustering coefficient distributions of sampled ego networks with the original ego network. In this work, we discuss an edge-based sampling method with forgetting factor over an evolving ego network stream of temporal edges.

6.3 Description of Data

Telecommunications' call graphs are one of the massive streams of calls generated in real-time. We made use of such anonymized temporal call stream of 31 days available from a service provider. The network data stream is generated a speed of 10 to 280 calls per second around mid-night and mid-day. On an average, we have 12.4 million calls made by 4 million subscribers per day. A Streaming approach is highly feasible for this kind of rapidly evolving data.

From the above massive stream of calls for 31 days, we built the ego networks by selecting egos with five different properties. The first ego network $egonet_1$ is built by selecting an ego with a degree equal to average degree of network. The $egonet_2$ with an ego of highest indegree centrality of graph and is also the node with highest eigen vector centrality of graph. The $egonet_3$ and $egonet_4$ with highest betweenness centrality and lowest outdegree centrality respectively for enhancing the diversity of ego networks. We built these ego networks by accumulating all the adjacent edges of the ego and their adjacent edges i.e a network of radius 2. We generated the ego network streams from a call network stream of 400 million calls made by 12 million subscribers on an aggregated scale. In order to avoid a duplicated number of edges as it is a multi-graph, we maintained unique edges between any pair of nodes in the network and mapped them onto a weighted graph.

6.4 Metrics for Evaluating Ego Networks

In this section, we discuss an extensive list of graph metrics we would use in the later sections to analyze the densification of call ego network to analyze the evolution of structural, topological and behavioral properties of call ego network and to evaluate the proposed sampling method of forgetting factor. We exploit these properties at the graph level and node level.

6.4.1 Graph level metrics

We studied the properties of ego network graphs using average degree, average weighted degree, density, diameter and average path length.

Additionally, for evaluating evolving samples using our proposed method, we compared the degree distributions of the samples at the end of 31 days with the original network using the kolmogorov-Smirnov test. We use the D-statistics from the test and also p-values to evaluate our null hypothesis (H_0) that our sampled ego networks follow the same distribution as the original ego network. The degree distribution of the network is obtained by counting the frequency of each degree d in the network. The frequency of each degree d is given by the number of nodes with degree d in the network snapshots at the end of 31 days.

We compared the effective size and efficiency of samples with that of ego network using ego metrics introduced by Burt (2009). The effective size of the ego network (ES) is the number of alters that an ego has, minus the average number of ties that each alter has to other alters. In the simplest form, for an undirected ego network of radius 1, the effective size can be given with the eq 6.1. Efficiency (EF) of an ego network is the proportion of ego's ties to its neighborhood that are "non-redundant." Efficiency is the normalized form of effective network size(eq 6.2). Therefore, it is a good measure for comparing ego networks of different sizes.

$$ES = n - \frac{\sum_{a=1}^{n}(d_a - 1)}{n} \tag{6.1}$$

$$EF = \frac{ES}{n} \tag{6.2}$$

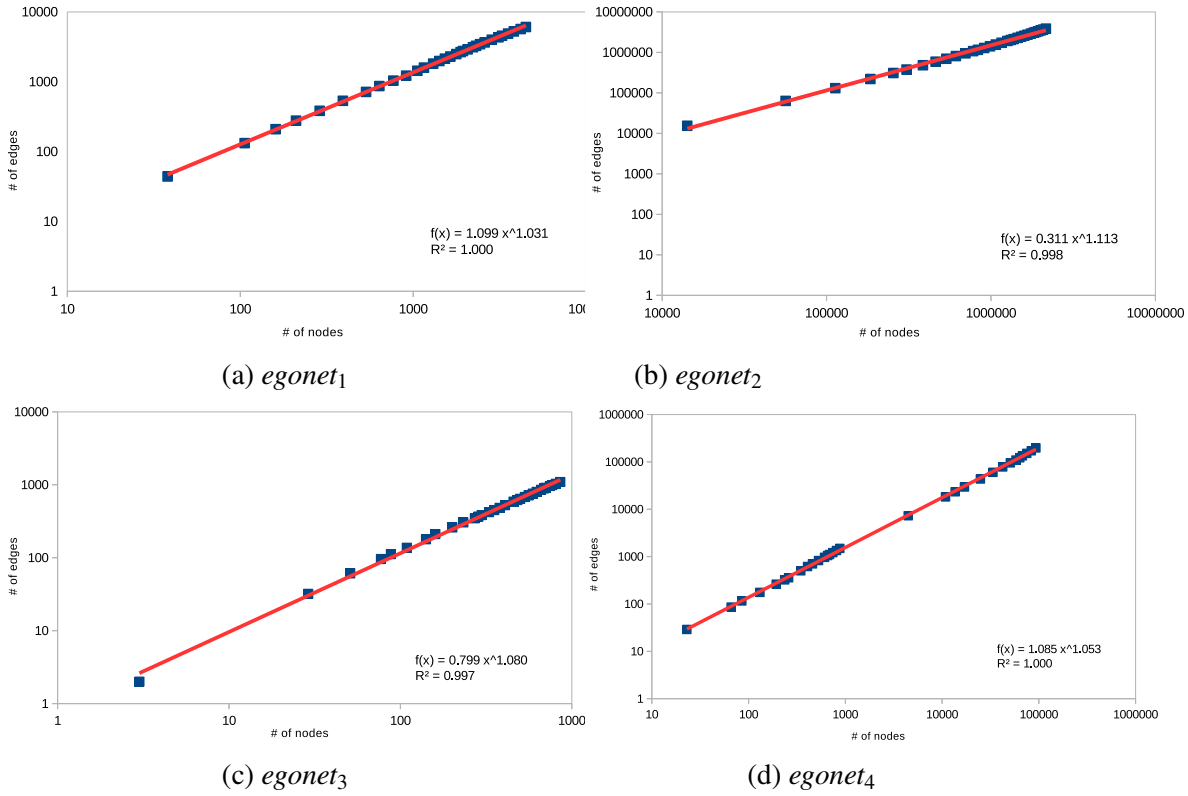
Where n is the number of alters in the ego network and d_a is the degree of an alter a.

6.4.2 Node level metrics

The node level centrality metrics discussed in the later sections are degree, weighted degree, closeness (CC), and eigen vector centralities (EVC). We also explored the eccentricity and clustering coefficient of the ego.

6.5 Densification Law on Evolving Call Ego Networks

Leskovec et al. (2005) studied the temporal evolution of number of nodes vs number of edges. Besides, the authors employed the measures of average outdegree to ascertain the



(a) $egonet_1$ (b) $egonet_2$

(c) $egonet_3$ (d) $egonet_4$

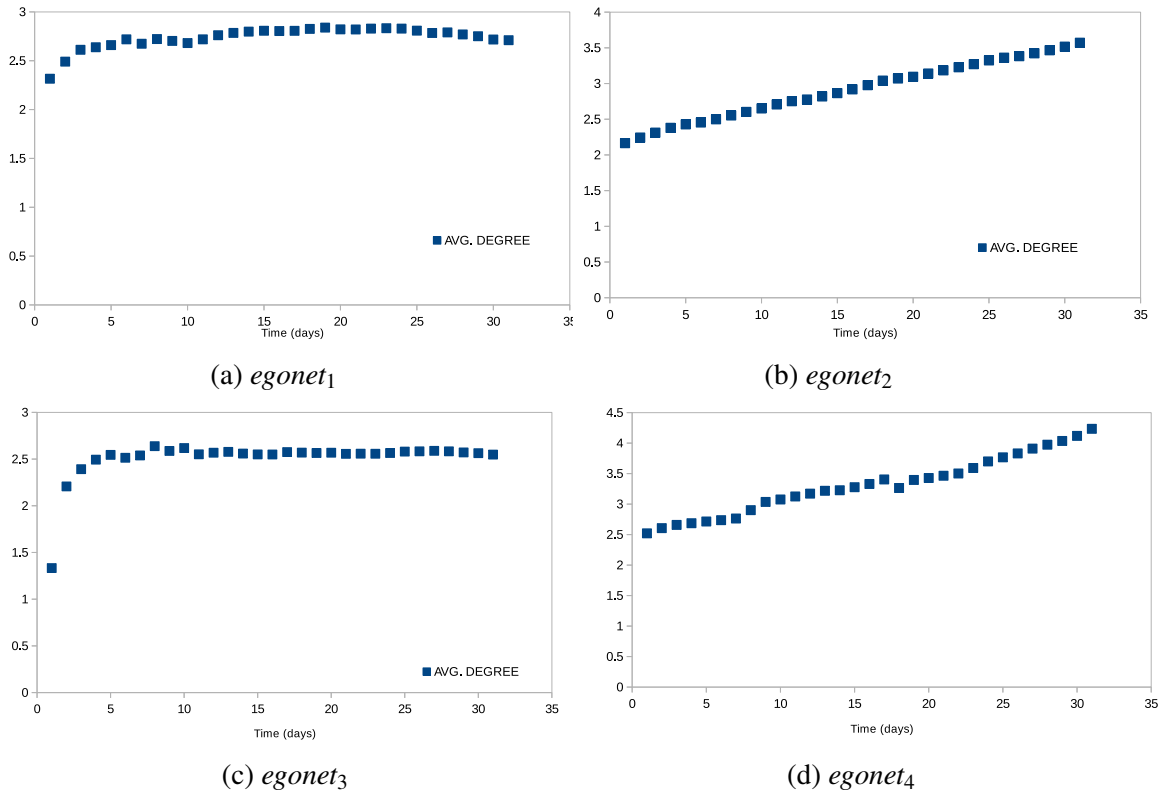
Figure 6.1: DPL plot for temporal call ego networks

densification law proposed by them. They validated that, most of these graphs densify over time, with the number of edges growing superlinearly to the number of nodes and their average degree increases. They investigated the above properties in an evolving citation graph, autonomous systems graph and affiliation graph. The authors stated that as the graphs evolve over time, they follow the relation given by the equation 6.3

$$m(t) \propto n(t)^a \tag{6.3}$$

Where m(t) and n(t) denote the number of edges and nodes of the graph at time t, and a is an exponent that generally lies strictly between 1 and 2. The authors refer to such a relation as a densification power law or growth power law where the number of edges grows superlinearly to the number of nodes. The authors also show that the average degree of these graphs gradually increases. With this justification, the authors prove that the graphs densify over time.

In this section, we investigate the densification power law (DPL) over the temporal stream of call ego network by depicting a densification power law plot (DPL plot) for the number of nodes n(t) and the number of edges e(t) at each timestamp t. In our experiments, we used a timestamp of one day.



(a) $egonet_1$ (b) $egonet_2$

(c) $egonet_3$ (d) $egonet_4$

Figure 6.2: Average degree evolution in temporal call ego networks for 31 days

We used the four temporal evolving ego networks from the call/edge stream, as described in Section 6.3. We grabbed the snapshots of ego networks at the end of each day and calculated the number of nodes and edges. Figure 6.1 shows the DPL plots for the call ego networks. As the slope of the line in a log-log plot gives the exponent in a power law relation, in the figure discussed above, we derived the lines obeying power relation with the best fits of 0.99 and 1.0 with their respective points. Therefore, the slope of these lines gives the densification exponents as a = 1.03, 1.1, 1.08, and 1.05 (in Figure 6.1(a), 6.1(b), 6.1(c) and 6.1(d) respectively) which shows a superlinear growth of edges over nodes. Hence, we deduce that the ego networks of a call network also follow the densification power law as many other socio-centric networks, with the number of edges growing super linearly to the number of nodes with their respective exponents a.

We consider the average degree of ego networks per timestamp, which is plotted in Figure 6.2. We see that the average degree of graphs for figure 6.2(b) and 6.2(d) (ego network of highest indegree centrality node and highest eigen vector centrality of graph) is gradually increasing. Average degree of graphs in Figures 6.2(a) and 6.2(c) are is slightly increasing with the evolution.

From the above experiments with the densification power law and the average degree,

we see that the graphs are densifying. Hence we require sampling techniques in real-time to analyze such enormous evolving data.

6.6 Evolution Analysis of a Temporal Ego Network

In this section, we have analyzed the evolution of the ego network ($egonet_1$) over a period of one month using the metrics mentioned in Section 6.4. As described in the earlier section, we have constituted the adjacent nodes of an ego and their adjacent nodes in the ego network as the stream progresses for a month. Then we took snapshots of the ego network per equal intervals of timestamp i.e one day in our case. To investigate the evolving structure and properties of a call ego network, we undertook a piecemeal structural analysis of network by employing the following metrics per day i.e. average degree, average weighted degree, density, diameter and average path length and derive some empirical observations. We also made use of a bunch of centrality metrics to study the importance of ego in the network and compare the position of ego during evolution; they are degree, weighted degree, closeness (CC), and eigen vector centralities (EVC). We also explored the eccentricity and clustering coefficient of the ego.

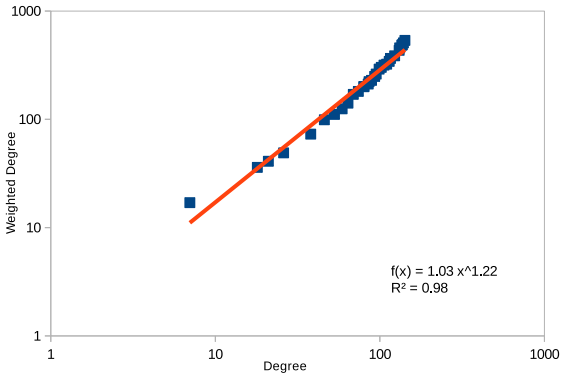


Figure 6.3: Degree vs weighted degree of ego network

Figure 6.3 plots the degree vs. weighted degree of the ego in the ego network over a log-log plot. The equation of the line that best fits our temporal data points is given in the figure. The slope of the line is given as 1.22, which shows a power relation between degree and weighted degree of a node. Therefore, we can say that the weighted degree of the node is growing superlinearly over its degree. The above analysis demonstrates a social behavior that the people are more interested in maintaining their old relationships or friends than making new friends. However, they also show little interest in making new pals as well.

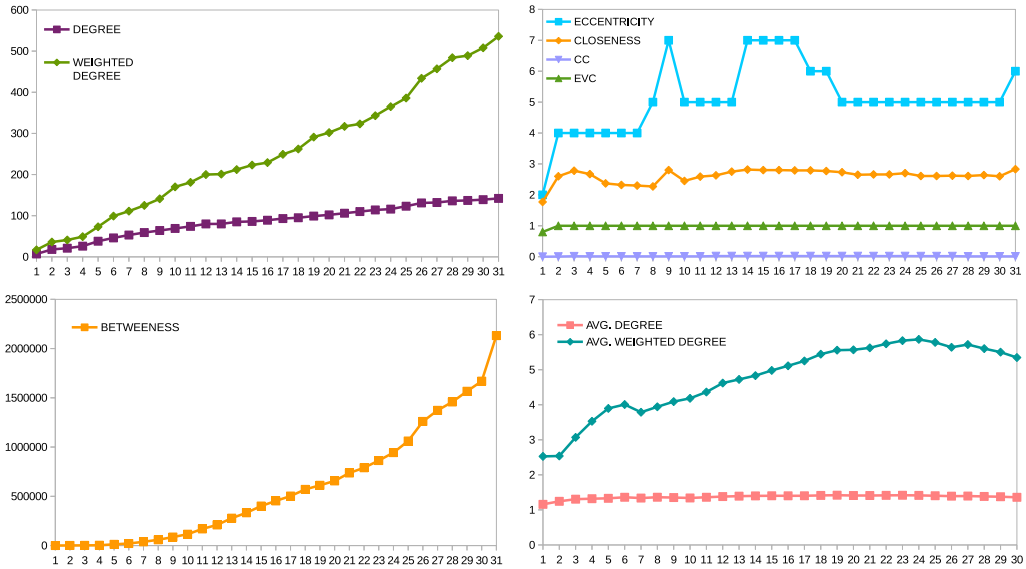


Figure 6.4: Metrics over a temporal call ego network

Figure 6.4 depicts the graph metrics and node metrics over the evolving ego network. When considering node metrics, we see that the CC and eccentricity of the ego increase with the evolution, but the EVC remains constant, as the ego remains the important person in the network with highest betweenness centrality. The betweenness centrality of the ego also increases with the function of network size.

Figure 6.5 displays the call ego network of a particular ego for day 1 and final accumulated network on day 31. The ego is represented with the red dot in the center. The figure illustrates the evolution of network for one month from timestamp 1 to timestamp 31. We maintained the tie strengths between the nodes by mapping the multi-graph to a weighted graph.

6.7 Sampling Ego Network with Forgetting Factor (SEFF)

In this method, we sample edges from the stream of a temporal network. We start by building the ego network of a specific ego and begin to scrape together all the adjacent ties to the ego and their adjacent ties. We do this by using a set for storing adjacent nodes. For every recurring edge, we increment the edge weight of the corresponding edge by maintaining a hash table. We impose a forgetting factor over edges, following successive grace periods. In our experiments, we use a grace period of 1 day. This means we apply the forgetting factor over the ego network as soon as the stream enters a new day, i.e., we forget the old edges each of a kind (i.e. edges between a pair of nodes), by some fixed percentage defined by the forgetting factor. The forgetting factor is given by

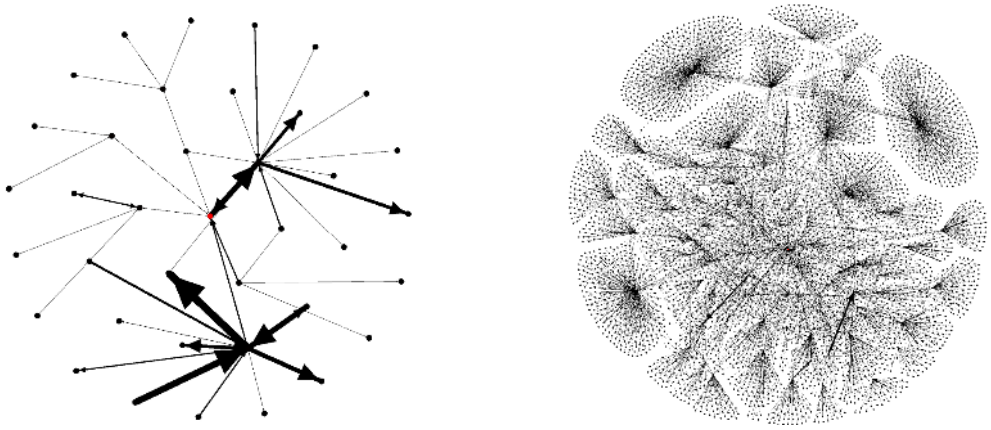


Figure 6.5: Evolution of a call ego network

two parameters, an attenuation factor α and a threshold θ. Where $0 < \alpha < 1$ and also $0 < \theta < 1$. After every grace period or update time t the tie strength between two nodes is given by the equation 6.4.

$$\hat{w_t} = w_t + (1-\alpha)\hat{w}_{t-1} \tag{6.4}$$

Where w_t is the tie strength between any two nodes in the ego network at time t. After every successive grace period, we decrease the edge weight by α and consequently remove the alter/alters adjacent to the corresponding edge, as the edge weight decreases than the threshold value θ. When α=1 we have a maximum forgetting i.e we forget the whole network except the network of the current day. When $\alpha = 0$, we get the original network. If the removed edge corresponds to an alter adjacent to the ego, we remove the adjacent edge and the alter, and all the second level alters adjacent to the alter itself, if the above condition is satisfied. If we forget a second level edge, not having a direct connection to ego, then we only forget the corresponding node. Following this strategy, we can have the most active alters in the ego network at the end of each day.

6.8 Evaluation Methodology

In order to evaluate our method SEFF discussed in Section 6.7, we applied it over a real-world streaming call Graph G of 31 days by randomly choosing an ego e and generating a sample stream of depth $d = 2$ at any point of flow. This was done by generating six real-time sample streams, where each sample stream S_i is generated by different combinations of $\alpha \in \{0.9, 0.8, 0.7, 0.5\}$ and $\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ discussed in Section 6.7. For investigating the above sample streams, we captured their snapshots of sample streams at the end of 31 days each. Each snapshot $S_i^{31} \subset G$. Beforehand, we took a snapshot G_e^{31}

of original ego network stream G_e of e (where $d(G_e) = 2$) at the end of 31 days from the socio-centric call graph G. Each sample graph $S_i^{31} \subset G_e^{31}$. We then compared the conclusive sample snapshots S_i^{31} where $1 \leq i \leq 6$, with the original ego network snapshot G_e^{31} by employing metrics discussed in Section 6.4. We use Kalmogorov-Smirnoff test to compare the degree distributions of the original network with that of samples. Conclusively, we derive some conclusions about the properties preserved by the sample networks.

6.9 Experimental Evaluation

The call networks are the special application scenario for employing our method as these networks are multi-graphs with more than one edge between two users, representing the strength of their relationship unlike a social network based on friendship and, follower and followee relations, where there is a single binomial relation between two nodes. However, the proposed method can be applied to networks with binomial relationships as it forgets edges and eventually forgets nodes. SEFF method is also appropriate for sampling weighted networks.

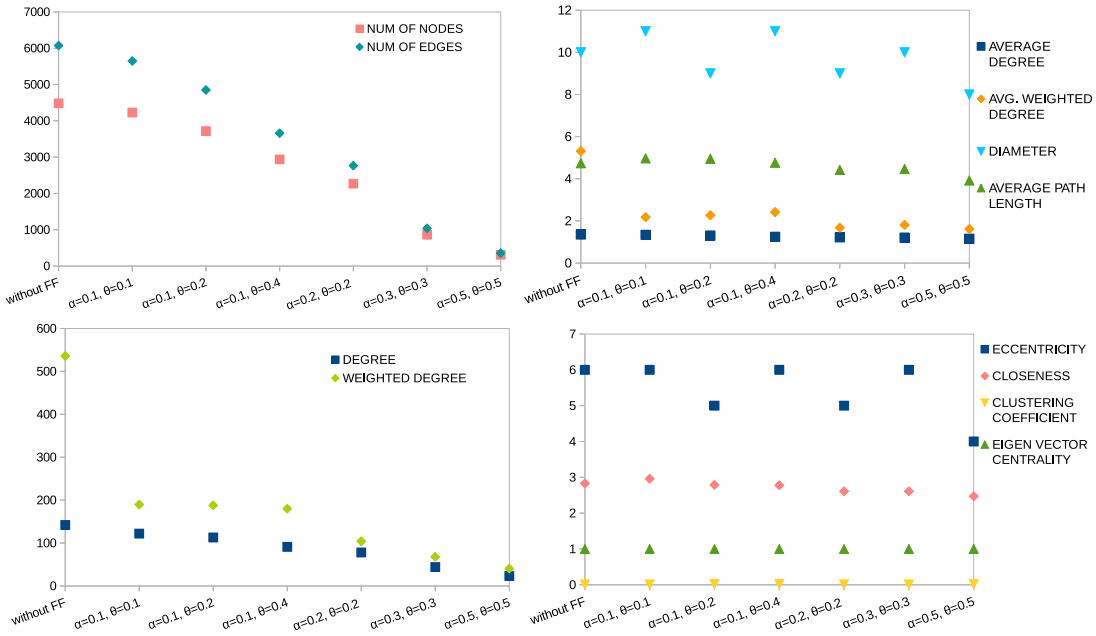


Figure 6.6: Metrics over ego networks with and without forgetting factor

We selected an arbitrary user "ego" from the real-world call/edge stream described in Section 6.3 and start building the ego network of ego with a two-step neighborhood, i.e., by acquiring the neighbors of ego and the neighbor of neighbors of ego. We take a snapshot of the ego network at the end of 31 days stream. Using the same ego, we start

constructing the sample ego networks (using SEFF) gradually as the stream flows for 31 days. For which, we have used six different combinations of α and θ corresponding to six different samples depicted in Figure 6.6. The figure also plots the values of computed metrics discussed in Section 6.4 over the conclusive sampled ego networks and the original ego network.

Figure 6.6(a) shows the number of nodes and the number of edges in the above described ego networks. We observe that the number of nodes gradually decreases with the increasing forgetting factor. For an attenuation value of 0.5 and a threshold value of 0.5, we forget 50% of the edges per day, between two adjacent nodes. This shows we always have the most active nodes with the increased forgetting factor. We also observed that the number of edges decrease in greater proportion than the number of nodes, Almost reaching equal for the highest forgetting factor in the illustration. This exhibits that the proposed SEFF method decreases redundant edges.

We also compare the degree distributions of the original ego network with the samples generated by using SEFF method at the end of 31 days. We applied Kolmogorov–Smirnov test to compare the degree distributions of the samples with the original network. The D-statistics and P-values of tests are given in table 6.1. The p-values are computed using the exact method. The significance level used for the comparisons is 5%. The results show that all the sampled distributions follow the distribution of the original graph. We also observe that the value of θ has a greater impact on the similarity of distributions than α in the SEFF method. We can see the pictorial representation of the degree distributions of the original graph and sample graphs in Figure 6.7

Table 6.1: Comparison of degree distributions using KS-Test

Samples	α=0.1, θ=0.1	α=0.1, θ=0.2	α=0.1, θ=0.4	α=0.2, θ=0.2	α=0.3, θ=0.3	α=0.5, θ=0.5
D-stat	0.146	0.138	0.173	0.146	0.191	0.096
P-value	0.114	0.124	0.065	0.182	0.105	0.724

Figure 6.6(b) depicts metrics over the ego networks. The diameter of the graphs varies with the inclusion and removal of the connecting nodes from the ego network. It depends on the network of ego selected. Average degree and the average path length decreases with the increasing forgetting; this shows that the networks shrink with increased forgetting. The SEFF method has a noticeable effect on the weighted degree of graphs.

The degree and weighted degree of the ego are plotted in Figure 6.6(c). Both the values decreased with the increased forgetting, while the drop in weighted degree is higher; this suggests that when we increased forgetting, we decreased the tie strengths but relatively maintained the ties. In Figure 6.6(d) we see that the eccentricity has a similar effect of

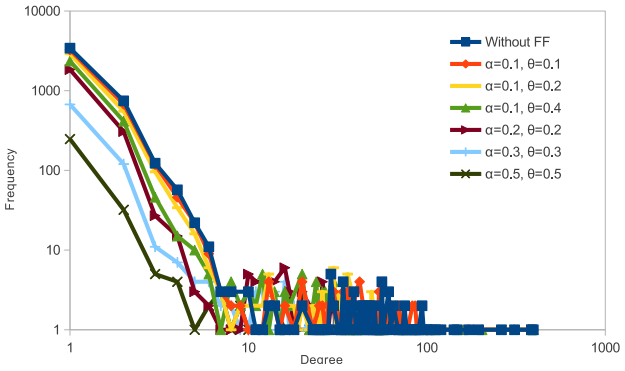


Figure 6.7: Degree distributions of ego networks at the end of 31 days with and without forgetting factor

diameter in the ego network graphs. This corresponds to the conceptual relation between diameter and eccentricity. Closeness of the ego with alters also decreased gradually with the increased forgetting factor. The clustering coefficient of ego is too low to compare. The eigen vector centrality portrays the important node in the network. SEFF preserves the importance of ego alongside forgetting.

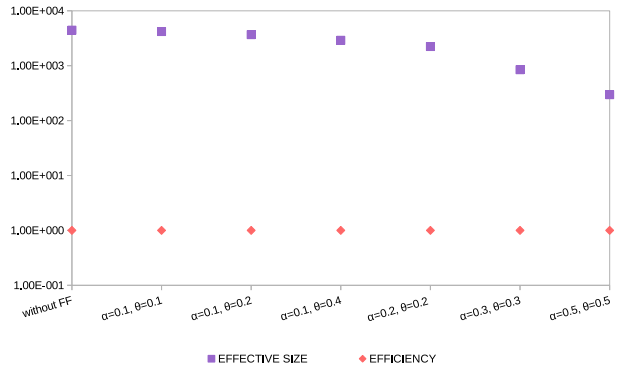


Figure 6.8: Efficiency and effective size of ego networks

Figure 6.8 illustrates the effective size and efficiency of the ego networks. There is a negligible difference in the effective size of samples. Efficiency of the network indicates the impact of ego in the network. In the given figure, we can observe that the efficiency of the network is maintained throughout the samples using SEFF. The measure of effective size of the network is not normalized with the size of the network. Therefore it decreases with the average number of ties that each alter has to other alters.

6.10 Chapter Summary

In this work, we analyzed the evolution of ego network for a period of one month. We exploited the structural properties of network and related them with the natural behavior of users. We also proved the densification law over the ego networks of call graphs for a period of one month and found that the graphs are densifying along time.

We observed the properties of evolving ego network and proposed a sampling method with forgetting factor for streaming multi-graph networks, which preserves the density of graph and retains the tie strengths between nodes. We evaluated our method by exploiting the ground truth of the original graph vs. samples generated by varying parameter values. Based on the empirical experiments, we prove that our method maintains the importance and efficiency of the network and decreases the redundancy while preserving most active and recent nodes from the network.

Part III

Application

Chapter 7

On Fast and Scalable Recurring Link's Prediction

The link prediction task has found numerous applications in real-world scenarios. However, in most of the cases like interactions, purchases, mobility, etc., links can re-occur again and again across time. As a result, the data being generated is excessively large to handle, associated with the complexity and sparsity of networks. Therefore, we propose a very fast, memoryless and dynamic sampling-based method for predicting recurring links for a successive future point in time. This method works by biasing the links exponentially based on their time of occurrence, frequency and stability. To evaluate the efficiency of our method we carried out rigorous experiments with massive real-world graph streams. Our empirical results show that the proposed method outperforms the state of the art method for recurring links prediction. Additionally, we also empirically analyzed the evolution of links with the perspective of multi-graph topology and their recurrence probability over time.

Keywords: *link prediction, evolving network, recurring links, graph streams, temporal networks*

7.1 Chapter Overview

Link prediction is a vital research problem in the area of network science. There are numerous papers addressing this problem from various perspectives and techniques. A summary of it can be found in Al Hasan and Zaki (2011a). Most of these methods differ from each other with respect to model complexity, prediction performance, scalability, and generalisation ability (Al Hasan and Zaki, 2011a). However, new challenges are com-

ing up with the emerging realms of technology, such as the Internet of Things (IoT), Web 3.0 and Smart cities. One of the critical challenge associated is to handle the avalanche of networked data being generated at high speeds. Most of this data flow is a consequence of recurring or repeated links/interactions/communications over time. For example, learning from repeated actions of users in IoT, like, to play the favourite songs, make automated breakfast or a reminder to call etc, without the need for a user to manually set the time.

Typically, the link prediction problem is understood in two settings: predict missing links from the same time interval in a static setting and predict new links for a future time step in a temporal/streaming scenario. However, the problem of predicting recurring links has been sparsely studied. Though the models for predicting new links can be expected to work on repeated links as well but they have been hardly evaluated on this basis (Tylenda et al., 2009). Moreover, the models for predicting new links tend to exploit indirect relationships or neighbourhoods of a node, which is extraneous in the case of recurring links and also impacts scalability.

Stream processing is an exemplary way of processing real-time data. It maintains temporal order which is of utmost importance. Nevertheless, it has a set of challenges associated with it, example one-pass constraints, incremental and decremental characteristics, bounded size algorithms for unbounded data, etc. Therefore, while trying to address these challenges, in this work, we propose to predict recurring links over time with the following contributions:

1. We propose fast and scalable memoryless sample-based streaming algorithm for predicting recurring links in evolving weighted/multi-graphs.

2. We empirically demonstrate how the recurring links are related to the past links in a temporal network and their topological significance in it.

3. We carry out extensive experiments with the real-world networks as large as 400 million edges. Our empirical results clearly demonstrate the high predictive efficiency achieved by the proposed model in comparison with the state of the art and space efficient techniques.

In an attempt to address the issue of scalability and performance we implement the method presented by Tabassum and Gama (2018) to provide online samples. Here we use this methodology to predict recurring links. As the links preserved from this technique are strong, active and latest, we found from our experiments that they provide a significant accuracy gain. The details of methodology are explained in section 7.3.

Rest of the chapter is organised as follows: In section 7.2 we presented a brief overview of the literature in the area of link prediction. Research works related to the problem of

recurring link's prediction are also summarized in section 7.2. The proposed model, algorithms and problem definitions are outlined in section 7.3. Section 7.4 briefs the data networks used and the analysis of recurring links regarding to their formation and evolution. Empirical evaluation and a comparative assessment with other baselines is also carried out in section 7.4. Finally, the conclusions and future works are summarized in section 7.5.

7.2 Literature review

Link prediction problem was initially more focused on static networks (Liben-Nowell and Kleinberg, 2007). Exploiting a number of node neighbourhood-based measures like similarity, proximity and topological features have been a common practice in the literature for predicting new or missing links. Some of these mostly studied features include common neighbours, Jaccard's similarity, Adamic/Adar, preferential attachment, Katz score, hitting time, rooted PageRank, SimRank, shortest distance, clustering index, etc. These features have been used to score and rank the nodes (Liben-Nowell and Kleinberg, 2007) or infused into supervised learning and probabilistic models (Al Hasan et al., 2006a; Wang et al., 2007; Benchettara et al., 2010; Wang et al., 2011) to boost their performance. However, the main issue with these models is the number of possible links is quadratic in the number of vertices in a social network, while the number of actual links added to the graph is only a tiny fraction of this number (Al Hasan and Zaki, 2011a). Therefore, the methods trying to address this problem are discussed below. As our work mainly deals with the combination of temporal networks, streams and recurring links, we present below some interesting works in these areas.

7.2.1 Time aware link prediction methods

Realising the highly dynamic nature of real-world networks, last decade has been more focused on evolving networks and time aware link prediction methods. A survey of related methods can be found in Marjan et al. (2018). However, the basic idea of using proximity measures or structural heuristics (features discussed above) remained the same in most of the works. Papadimitriou et al. (2012) found that local neighbourhood measures based on 2 hop paths are not as efficient as global network measures. On the other hand global measures are computationally expensive. In order to provide faster and scalable results they exploited node neighbourhoods of path length l. This gave a better performance but defining l is crucial. However, the method attained the best precision at $l = 3$ for the social networking data sets used. Whilst Sarkar et al. (2012) proposed

a non-parametric link prediction method considering the local networks of nodes with two hops. Raymond and Kashima (2010) obtained low-rank approximation of similarity matrices for faster link propagation when compared to conjugate gradient-based methods. Song et al. (2009) used count-min sketches to approximate path-based proximity measures like Katz, rooted PageRank, and escape probability for large graphs. Their values were incremented for every snapshot. Meanwhile, Rossetti et al. (2011) included features from multi-dimensionality of networks. They combined the multi-dimensional versions of node neighbourhood features with the temporal information to achieve better performance. Most of the methods listed above tried to provide a faster version in terms of their global counterparts using approximated solutions. Nevertheless, all the methods above have mainly focused on new or missing links and exploiting many computationally extensive features in the case of very large networks.

7.2.2 Link prediction in streams

With the memory limitations of the conventional systems, it is infeasible to hold massive real-time data in memory and perform multiple passes on it. Streaming data models have gained much interest nowadays due to the current requirements of real-time processing and the nature of unbounded evolving networks. Stream processing solutions have proved to be efficient in presenting plausible models on temporal and evolving data. Some examples include, classification (Pan et al., 2014), concept drifts (Gama et al., 2014), triangle counting (Lim et al., 2018), matrix factorisation (Matuszyk et al., 2018), change detection (Pereira et al., 2019), estimating global network properties and sampling (Ahmed et al., 2017; Tabassum and Gama, 2016b; Zhang et al., 2017), etc.

Though streaming methods were explored in many areas, it is surprisingly less in the area of evolving networks' link prediction. One of the notable works was given by Zhao et al. (2016). They followed the suite of adapting the typical heuristics of common neighbours, Jaccard's similarity, Adamic/Adar but in a graph stream setting. To make the approach scalable they introduced a vertex-biased sampling technique which uniformly sampled the vertices of all degrees without prioritising high degree vertices. The reported accuracy was almost to the exact method.

7.2.3 Recurring links prediction

The appearance of new and recurring links in a temporally evolving network creates two distinct types of problems. One is to predict new or missing links and the other one is to predict recurring links. Considerably, our focus here is on the second one. From the

perspective of O'Madadhain et al. (2005b), networks are described based on two types of relationships: (i) **persistent relations**, where the relationship between two nodes is stable for a long time. This network is modelled by a simple graph and new edges are created between vertices that are at a distance two or higher, e.g., friendship, road network, etc.; and (ii) **discrete events**, where the relationship status is different at every point in time. In this case the edges may appear not only between vertices at distance two or higher but also between vertices that are already connected (repeating/recurring edges). For example, citation, transaction and communication networks etc. The authors were interested in the discrete scenario. They trained the classifiers using node neighbourhood and event-based features to predict the probabilities of all links in a time series setting. Therefore the results are mixed and not differential for recurring links.

Another interesting work was presented by Tylenda et al. (2009). Referring to the above work of O'Madadhain et al. (2005b), the authors realised two main problems in link prediction: (i) prediction of new links; and (ii) prediction of repeated links. They found that the number of repeated links in co-authorship networks is as high as 50 % and can reasonably expect similar or more in other networks. Therefore, they predicted repeated links using a time-aware maximum entropy model. The links were forgotten when they violate the constraints imposed by the model, using temporal weights. As their results report, a simple temporal baseline model (last, count) outperformed all the other methods even the authors time-aware maxent. Likewise as in the case of new links. Purposefully we will discuss in detail about this baseline model in section 7.3.4.1.

Zhu et al. (2016) adapted an incremental approximate variation of block coordinate gradient descent approach in contrast to the global version, which is computationally prohibitive. The authors evaluated the accuracy of their approach in terms of predicting 'all' links (new links and recurring links) and also new links distinctly. Surprisingly, their incremental approach performed weakly than the global counterpart in terms of all links and better in few samples in terms of new links. This shows that the presence of recurring links in 'all' links made the predictions worse.

As learned from the previous works temporal information of the links is very beneficial in predicting links and also natural weights on links, which haven't been explored in any of the works until recently in Moradabadi and Meybodi (2018). Instead, most of the weights assigned to the links are gained from the neighbourhoods of the network which are less beneficial in the case of recurring links. Therefore we propose algorithms by exploiting the above properties of edges.

7.3 Modelling and predicting recurring links

In this section we explain the notations, definitions and the problem statement we are trying to address, followed by our model for predicting links and its complexity analysis. Towards the end of the section we present other baselines for comparisons.

7.3.1 Problem definition

We model an evolving network G as a stream of links/edges $\{e_1, e_2, e_3, e_1, e_4...\} = E$ which is a multiset. Every edge $e_i = (u, v, t)$ is composed of a pair of vertices's from V and a time-stamp t, which indicates the time of occurrence of e_i or simply e which is unique. We assume that the edges are streaming in the order of time-stamps. The basic intuition of the proposed approach is for an undirected multi/weighted graph stream. However, a directed graph can be used with a small modification considering an edge between (u, v) different from (v, u).

Definition 7.3.1 (Recurring links)**.** In a multi-graph stream an edge e is considered recurring if it reoccurs randomly in G at various time-stamps t. Another variable τ is a discrete time-step/time-interval with granularity defined by the user or based on data availability. Initially when $\tau = 0$ we have $G(\tau) = \emptyset$. At every τ, the weight w of an edge e is computed as

$$w(e, \tau) = \#\text{ of occurrences of an } e \text{ in } \tau \tag{7.1}$$

The weights w are incremented in a streaming way sequentially. For weighted graphs w is the weight of e in the time-interval τ.

Definition 7.3.2 (Edge vector stream)**.** Every edge e in G can occur in multiple τ's. Therefore every edge is a temporal vector $\mathbf{a}$ of the state of presence (with a weight w) or its absence in every τ. S is a dynamic sample updating in a streaming fashion. At any time-step, S is given as $S(\tau)$. m is the number of unique edges in $S(\tau)$, i.e. $m = |S(\tau)|$. At every τ, $S(\tau)$ contains an instance of $m \times a$ with updated values. $\hat{m}$ is the number of links to be predicted, which can be varied based on requirement or fixed overtime.

Definition 7.3.3 (Problem statement)**.** Given a temporally evolving network stream until $G(\tau)$, where $\tau = \{0, 1, 2...n\}$, our model aims to predict the probability distribution of edge recurrence in $\tau + 1$.

7.3.2 RLP: Recurring link's prediction using temporal bias and frequency

For predicting recurring links with limited space, efficiency and fastness, we propose our model using the dynamic sampling technique (*SBias*) given by Tabassum and Gama (2018). If we consider picking a uniform random sample from a power law distribution (which most of the real-world networks follow), it will likely result in getting all low degree nodes from the long tail. Techniques such as stratified or systematic sampling (Cochran, 2007) would need extra information apriori, like degree distribution of nodes, total population size, variance etc. Moreover, they need multiple passes on data. Which are expensive on massive networks. On the contrary, the samples generated using (*SBias*) are proved to preserve certain important properties of temporal networks. They are less biased to disconnected components, low degree nodes and follow true network distribution (Tabassum and Gama, 2018). Henceforth we make use of this strategy and moreover it applies to multi-graphs.

RLP incorporates a forgetting function (RLP_f) from the class of memoryless exponential functions to weight edges as in *SBias*. This function is applied on every unique edge e in $G(\tau)$, after every τ to get $\hat{w}(e,\tau+1)$, which is the predicted weight of an edge in $G(\tau+1)$. The function is defined as follows in Equation 2.

$$\hat{w}(e,\tau+1) = w(e,\tau) + (\hat{w}(e,\tau-1))(1-\alpha) \tag{7.2}$$

The parameter α defines the bias rate and typically lies between 0 and 1 inclusive. In general, this parameter α is chosen in an application specific way. When α is 1 we forget all of the previous edge occurrences and only maintain information from the latest τ. When $\alpha = 0$ the function returns the global network G with accumulated weights.

We also experimented with another memoryless exponential function that is exponential smoothing RLP_{ex} in the place of the above Function 7.2, which is a slight variation of it given as

$$\hat{w}(e,\tau+1) = \alpha w(e,\tau) + (\hat{w}(e,\tau-1))(1-\alpha) \tag{7.3}$$

However, this change didn't have any effect on the predictions, except increasing slightly the time and space complexities in the case of exponential smoothing variation. For a particular e, the above function with currying can be given as $\hat{w}_e(\tau+1)$. As this function is applied on every $\mathbf{a}$ in m independently, it explains that for every e the prediction is not based on its neighbourhood or global information. If e does not recur in $\mathbf{a}$ then $\hat{w}_e(\tau+1)$ is monotonically decreasing. The above function is imposed over m edge vector streams giving m time series. Considering every instance of τ as a vector, we determine the

probability distribution of edge recurrences by holding any τ as in Equation 7.4 for RLP_f and respectively for RLP_{ex}.

$$\hat{w}_{\tau+1}(e) = w_{\tau}(e) + (1-\alpha)\hat{w}_{\tau-1}(e) \tag{7.4}$$

7.3.2.1 Link labelling

In Function 7.4, for every τ we observe a power law distribution with a long tail. This long tail if not pruned will be asymptotic to the x-axis and unbounded. The memoryless functions hold the current distribution and discard all the data, but even the current distribution (after every τ) in this case is large enough. Therefore we prune the outcome at every τ using a threshold θ. The point here worth noting is that incase of *SBias* the threshold is only applied on the part $(\hat{w}_e(\tau-1))(1-\alpha)$ of Equation 2, while in *RLP*, we apply the threshold on the total weight i.e., $\hat{w}_e(\tau+1)$ at every τ to predict weights in $\tau+1\cdot$ Therefore the sample generated by *RLP* is less biased to low degree nodes than *SBias*.

The threshold θ can be constant or tuned according to the number of links to be predicted ($\hat{m}$). Here if the updated weight of an edge is greater than or equal to the threshold then the link is expected to recur. In the Equation below $\hat{y}_e$ is the label of link e at a time step τ.

$$\hat{y}_e = \begin{cases} 1 & \text{if } \hat{w}_{\tau}(e) \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

Where θ lies between 0 and 1 when $\hat{w}_{\tau}(e)$ is normalised using a constant or min-max technique. If it is not normalised then θ lies between 0 and the highest $\hat{w}_{\tau}(e)$ in the network, which would save the extra computation and memory space needed for normalising.

A higher value of α retains most recent edges in $S(\tau)$ exponentially in the order of τ, while a lower value tends to include older edges also. Likewise, a higher value of θ would retain more strong and stable edges in $S(\tau)$ and a lower value retains less strong and stable ones also. Therefore, we can tune the parameter according to the requirement or the number of links to be predicted ($\hat{m}$). The model is illustrated with a simple example in Figure 7.1. The weights on the edges are their frequency in the given time-step i.e w which aree not stored. The weights on the edges in the memory is $\hat{w}$. A cross indicates the edge was absent in that τ and the cross in memory indicates the memory is not used.

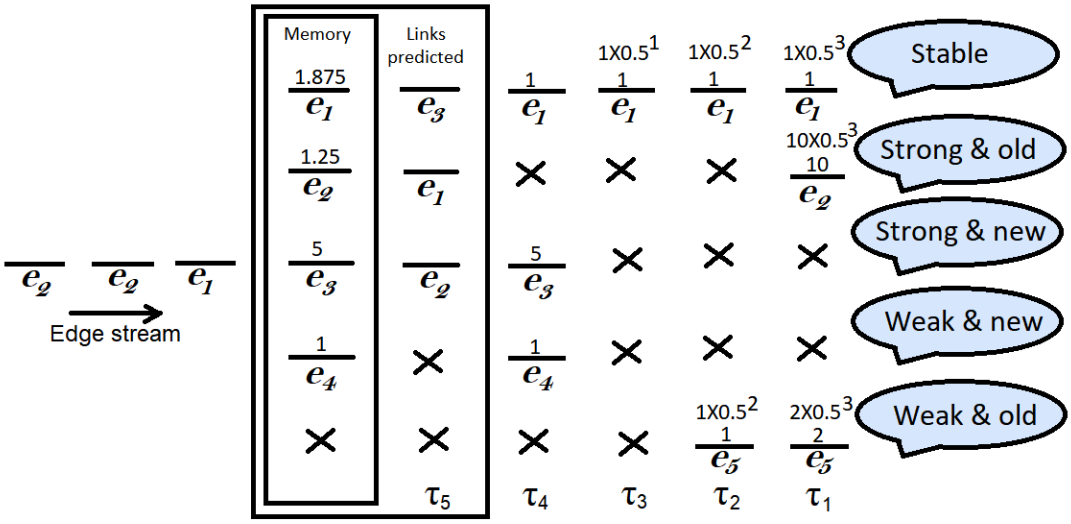


Figure 7.1: Recurring link prediction model with $\alpha = 0.5$ and $\theta = 1.5$

7.3.3 Complexity analysis

After each τ, the edges in the sample with $\hat{w}_\tau(e) \leq \theta$ are deleted from it, retaining stronger and stable edges from m but we still have some weak edges from the latest τ because of the recent bias. Therefore, the sample size $|S(\tau)|$ which is m decreases to $\hat{m}$ at $\tau+1$ by deleting the edges in long tail at every τ. If e appears again in G at a time-step $\tau+i$ it gets added to the $G(\tau)$ again. The space complexity is $O(\hat{m})$ which is the number of edges we need to predict or output size. In the worst case is $O(m)$, which is much less than $|E|$. We use hash maps to store the edges and their weights. So the time complexity is also $O(\hat{m})$ for all the network or $O(1)$ per edge.

7.3.4 Baselines with extensions

We evaluated a number of fast and efficient algorithms based on sampling or weighting network.[1] Note that for all the algorithms used, the sample size is not fixed. We varied the sample size/threshold over time τ to match the number of links to predict. Even in the case of reservoir-based samples the parameter for reservoir size is not treated constant but changing in real-time. When the reservoir/sample size changes the algorithms adapt to the new size. The algorithms are briefly discussed below:

[1] Java code for all the algorithms is available at https://github.com/ShaziaTabassum/Evolving-Multigraph-Stream/tree/master/networks/src/dynamic/sampling.

7.3.4.1 Sort by Last, Count (*SLC*)

This method was employed by Tylenda et al. (2009). It sorts the node pairs by the time of their most recent linkage and the number of links between them. This was proved to be the simplest and most efficient method in the authors' analysis even outperforming their maximum entropy model. We also implemented this in a stream setting. To avoid the sorting complexity, we use the threshold over the distribution as in Section 7.3.2.1.

7.3.4.2 Linear Weighting Scheme (*LWS*)

In this case, all the instances are weighted equally irrespective of their time of existence. We employ this methodology to investigate the significance of the time of occurrence in making predictions. This method treats data as in a static setting. The edges are sorted based on the frequency and the top edges are considered probable candidates for future links. However, to make the global network scalable enough, we processed it as a stream without considering temporal importance and pruned the links by applying threshold over the weights as in Section 7.3.2.1.

7.3.4.3 Space Saving (*SS*)

Space saving algorithm given by Metwally et al. (2005) is a fast and space efficient alternative for approximately finding frequent items from a data stream. This algorithm maintains the partial interest of information as it monitors only a subset K of items from the stream. It maintains counters for every item in the sample and increments its count when the item re-occurs in the stream. If a new item is encountered in the stream it is replaced with an item of the least counter value and its count is incremented. This leads to top K items at any time in the stream. As all the algorithms we used above are based on the frequency of link occurrence, we adapted space saving to find most frequent top K links from the edge stream at every τ to predict the recurring links in $\tau + 1$.

7.3.4.4 Reservoir Sampling for Multi-graphs (*RS-M*)

Reservoir sampling is a random sampling technique for unbounded data streams proposed by Vitter (1985). This algorithm maintains a reservoir with a predefined sample size m. Firstly the reservoir is filled with the initial items from the stream. Every item after that is computed for the probability m/i of being inserted. Where i is the length of the stream exhausted till then. If the probability of the contending item in the stream is greater than the probability of an item in the reservoir which is $1/m$, then uniformly at random an item is picked from the reservoir. The picked item is replaced with the item in the stream. In

case the probability is less the streaming item is discarded. Items in our case are links arriving in the stream.

To do a comparative assessment we present a multi-graph invariant of the reservoir sampling, while the basic notion remains the same. When a recurring link in the stream has a probability to enter the reservoir, it's weight in the reservoir is incremented by one, without modifying any other links. If a link is deleted from the reservoir, the information about its weight is not stored. When a new link is being added to the reservoir it is assigned a weight of 1 initially.

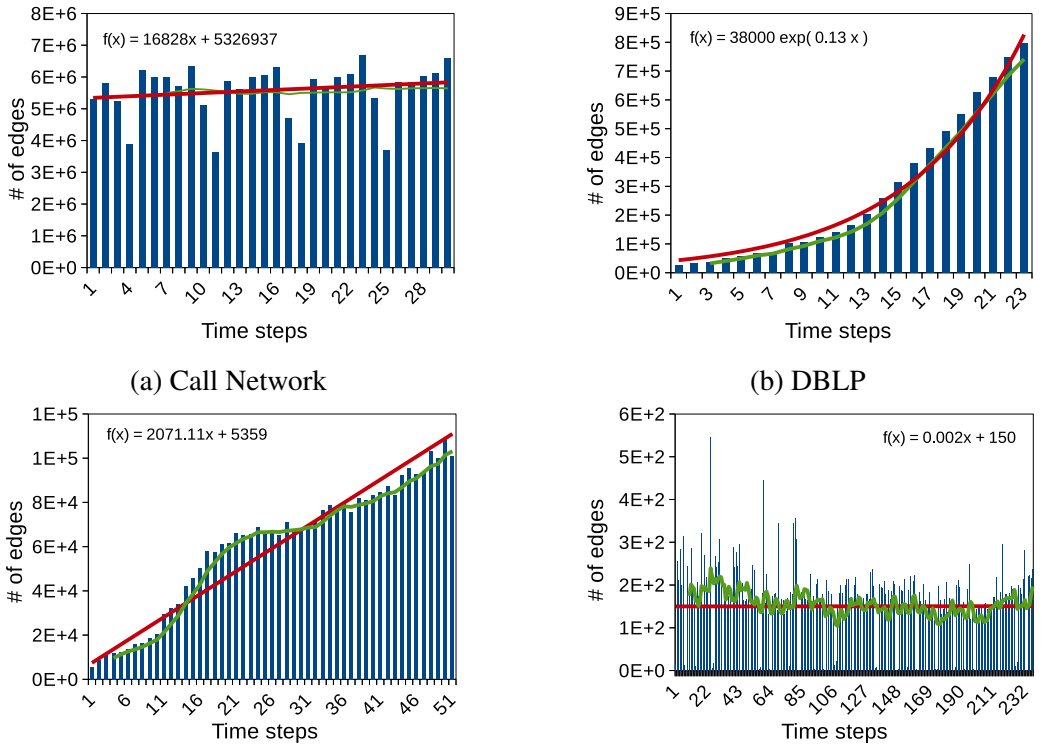
Table 7.1: Networks' properties

Datasets / Description	$\#E$	$\#V$	$\#\tau$	Avg. Degree	Avg. Weighted Degree	Avg. Deg per τ	Avg. Edge Stability	Avg. recur
Call Network	389 994 643	12 213 391	30 (days)	15.570	63.860	0.519	0.072	4.1
DBLP	17 861 201	1 242 068	24 (years)	16.323	28.898	0.223	0.032	1.76
Last.fm band	18 956 166	172 896	51 (months)	10.202	218.780	0.200	0.427	21.27
Radoslaw	82 927	167	237 (days)	69.269	993.140	0.292	0.063	14.34

7.3.4.5 Biased Random Sampling for Multi-graphs (*BRS-M*)

This algorithm is proposed as a biased variant of the Reservoir sampling in Tabassum and Gama (2016b). Unlike the above algorithm where the probability of incoming items diminishes as the stream progresses, this algorithm ensures every item goes into the reservoir irrespective of the stream length. An item from the reservoir is chosen for replacement at random. Therefore, the item insertion is deterministic but deletion is probabilistic. An item staying for a long time in the reservoir has a higher probability of getting out than an item inserted recently. Consequently, the items in the reservoir are not biased to very old items as in *RS-M*.

Considering a multi-graph stream, we use the same technique as the above *RS-M* to extend this algorithm. For every recurring link in the reservoir, its weight is updated. For a new link, the assigned weight is 1 and it replaces a link from the reservoir.



(a) Call Network (b) DBLP

(c) Last.fm band (d) Radoslaw

Figure 7.2: Trend analysis over time. The time series is smoothed using a moving window average (green) and a curve fitted over it to recognize the trend (red).

7.3.5 Experimental set-up

We perform a set of rigorous experiments over several data sets from slowly evolving networks, like DBLP, to high-speed networks, like phone calls, and as large as billions of links over time. We also choose a bipartite network to recommend items based on recurring links. Below we first describe the semantics of the data sets followed by an analysis.

7.3.5.1 Data sets

Here we present the brief description of data sets used and their properties.

Call Network: We use a massive anonymised (Call Detail Records) CDR data provided by a service provider. The average speed of data is 10 to 280 calls per second around mid-night and mid-day. Calls in the data are associated with time-stamps when the call was initiated.

Dblp: This is the collaboration graph of authors of scientific papers from DBLP computer science bibliography which is obtained from KONECT[1] networks. The initial timesteps of the available data are too sparse with just one edge per year. Therefore we considered the 24 years of data towards the end of temporal period. An edge between two authors represents a common publication. Edges are annotated with the date of the publication. There may be multiple edges between two nodes, representing pairs of authors that have written multiple publications together.

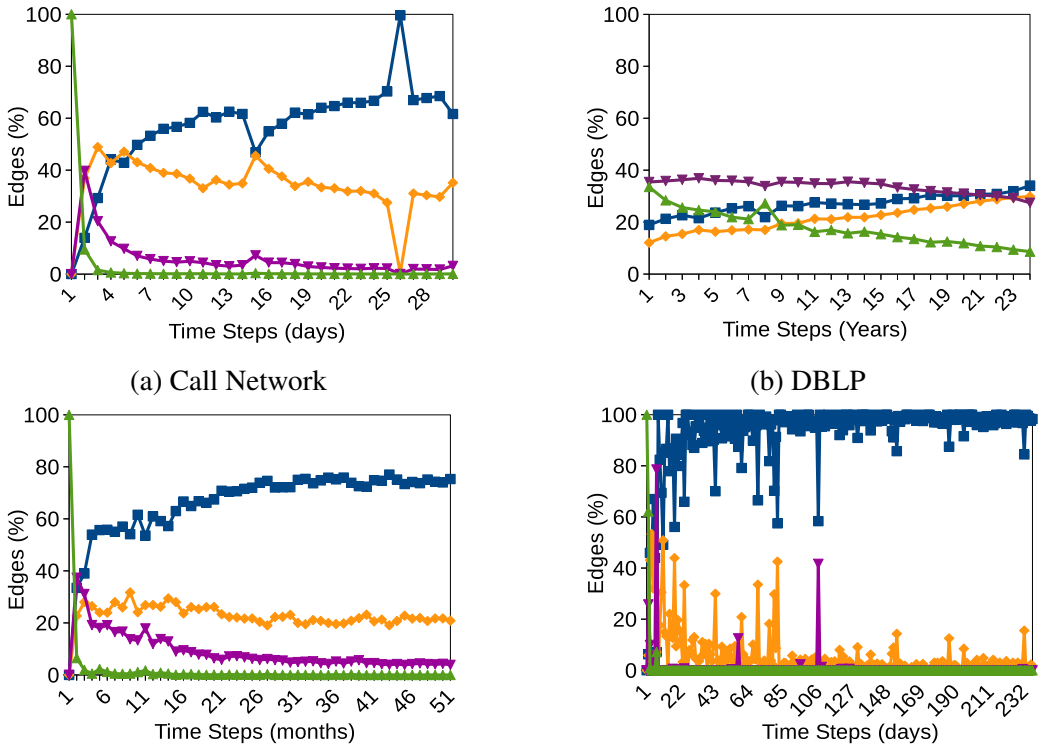
Last.fm band: It is a bipartite network of user–band listening events from the music website last.fm[2]. An edge shows that a user listened to a song of a band at a particular time.

Radoslaw: This is an internal email communication network between employees of a mid-sized manufacturing company[3]. The nodes in the network represent employees. Edges between two nodes are individual emails provided with time stamps.

A description of the data sets used appears in Table 7.1. Remember that we define *Stability of an edge* as the number of time steps it is present in a graph stream over the total number of time steps (Tabassum and Gama, 2018). The average recurrence times of an edge is defined below. The temporal distributions of data are displayed in Figure 7.2.

Average recurrences per link (*Avg. recur*): This is a measure to quantify the number of times a link is recurring in a network on an average. We give this by total number of edges over number of unique edges in the dataset or average weighted degree by average degree of a network. It can also be referred as the mean of multiplicity of all the edges in the multiset E. This value for different networks is given in Table 7.1. For a network that do not have recurring links this value would be equal to 1. The figure shows that on an average the users in telecom call to a same recipient 4 times in a month. In DBLP which has the least avg.recur interprets that the authors didn't do the same collaboration more than once or twice (on average) considering directed edges in the data. We can also observe that the users in Last.fm heard on average 21 times the same band in around 4 years. In Radoslaw the users sent on an average 14 emails to the same person in 237 days. These values indicate the frequency of recurring links in a network.

[1]Data available at http://konect.uni-koblenz.de/networks/dblp_coauthor
[2]Data available at http://konect.uni-koblenz.de/networks/lastfm_band
[3]Data available at http://konect.uni-koblenz.de/networks/radoslaw_email



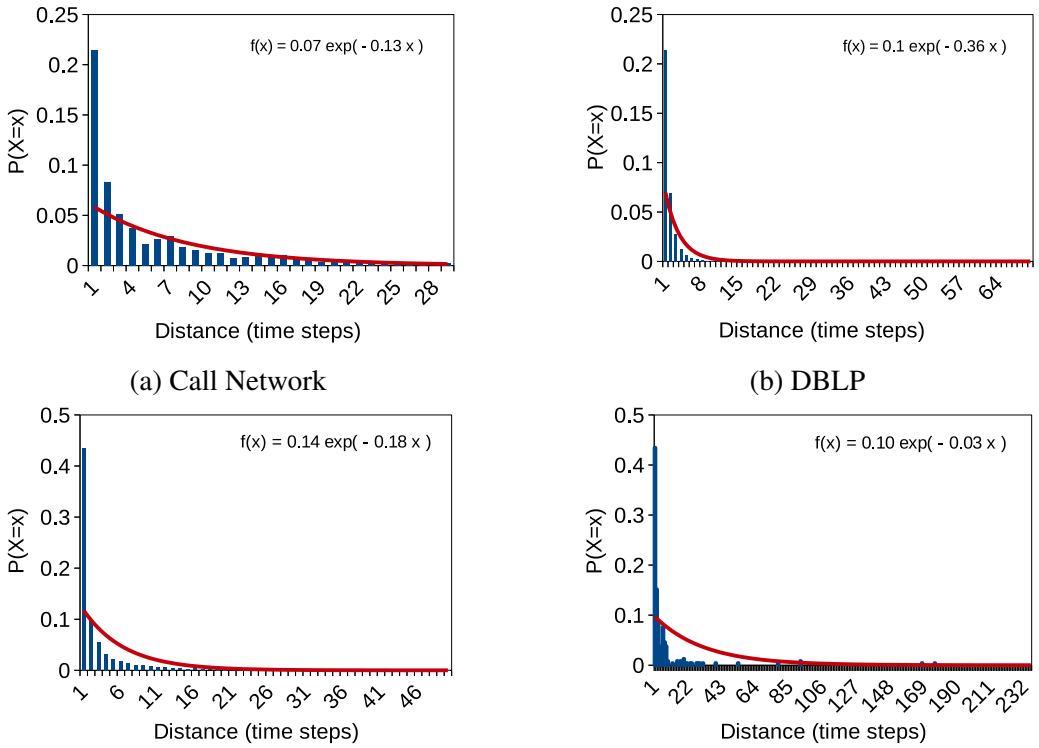
(a) Call Network (b) DBLP

(c) Last.fm band (d) Radoslaw

Figure 7.3: link analysis over time with the multi-graph perspective
■ Repeated links between two old nodes (Recurring)
◆ New links between two old nodes (New)
▼ New links between one old and one new node (New)
▲ New links between two new nodes (New)

7.3.6 Why is it important to predict recurring links?

Tylenda et al. (2009) found that 50 % of the links in the data they analyzed were repeated links. They expected even more in other data sets. We analyzed the links in our data and recognized their evolution over time by classifying them into two types based on the multi-graph topology: (i) recurring links, i.e., its an edge that occurs multiple times between two known/old nodes; and (ii) new links, this can be further subdivided into 3 categories: i.e., its a new link that occurs between: (a) two old nodes, (b) one old node and one new node or (c) two new nodes. Most of the link prediction models deal with the first category of new links (between two old nodes) only, while the other two cases are difficult to predict. In Figure 7.3 we indicate the percentage of recurring or new links at any time step τ corresponding to the links in 0 to $\tau - 1$. One can see that the recurring links in the figure are making the highest percentage and are gradually increasing. In some instances, 100 % of the future links have reoccurred. The analysis reflects the apparent behaviour of users in the networks. In the case of Call network and Radoslaw data, the users are



(a) Call Network (b) DBLP

(c) Last.fm band (d) Radoslaw

Figure 7.4: Recurrence probability of links. The bars (blue) indicate the empirical mass probability of links from τ that has recurred from the previous time-steps at a distance given on x-axis. The curve (red) is a fitted function over it.

more interested in communicating with the old pals than making new contacts. So as in DBLP if two authors publish a paper most probably they will publish a second paper, but in DBLP one can observe that the new connections are also much appreciated than in the other three networks. This may lead to the exponential growth of network size which was shown in the Figure 7.2. Last.fm has a similar trend where the users are interested in listening to the same bands as earlier and fewer users try new bands. Therefore, it is highly significant to predict recurring links as they make a large fraction of future links in most of the real-world networks.

7.3.7 How are the reoccurred links associated with past links?

Chen et al. (2013) found the young links to be more informative than the mid-age links and much more informative than old links in a temporal network. While their influential power differs with data. We analyzed our data sets to check what proportion of the links in the last time step τ have already occurred in the previous time steps $\tau-1, \tau-2, \ldots$. To derive a space efficient model we kept on deleting the edges already found. Figure 7.4

shows the re-occurrence probability of the links in the last time step based on previous time steps for different networks. Distance on x-axis indicates the distance between the current τ and previous $\tau's$, i.e. the distance between τ and $\tau-1$ is 1. One can observe that most recent links to τ have the highest probability of recurrence in it, and the probability is decreasing exponentially as the temporal distance to τ is increasing. We used Maximum Likelihood Estimation (MLE) to find the value of α in the exponential function which differs with data sets.

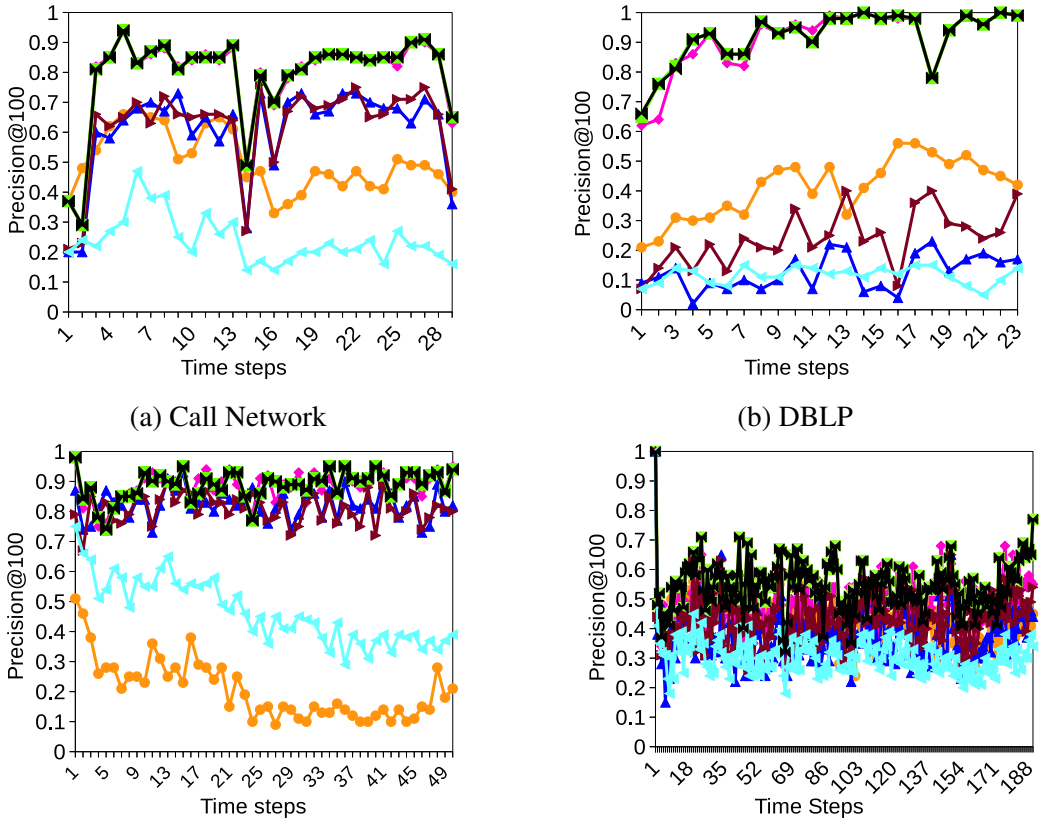
Table 7.2: Recurring link prediction models

Algorithm	Model Abbreviation
Recurring link prediction with forgetting	RLP_f
Recurring link prediction with exponential smoothing	RLP_{ex}
Sorting based on last, count	*SLC*
Linear weighting scheme	*LWS*
Space saving	*SS*
Reservoir sampling for multi-graphs	*RS-M*
Biased random sampling for multi-graphs	*BRS-M*

7.4 Experimental Evaluation

7.4.1 Prequential Evaluation for time-series graphs

To evaluate and compare the given temporal models, we followed prequential or predictive sequential evaluation methodology (Gama et al., 2013; Oliveira et al., 2018). It is one of the standard frameworks for evaluating models in a stream setting. The initial sequence of examples is learned to predict the upcoming examples in the stream. Then an error is calculated using a loss function ($L(y_i, \hat{y}_i)$) between the true values and the predicted values. For every next iteration, the true values are combined with the learned sequence to predict the next upcoming values. The number of examples to be learned can be growing or forgotten using weights. In this work, we implement it over temporal network data. The sequence of items is a time series, with every time-step having a subset of all the links in the network. As given in Equation 7.6, the loss function is calculated for all the $\hat{m}$ number of links predicted at every τ. In the case of *LWS*, we learn the time-series from 0 to $\tau-1$ to predict τ and 0 to τ to predict $\tau+1$. For *RLP*, the predicted links from the previous time-step and the true links of that time-step are used to predict the links in the next time-step. *SLC* uses only the true links from the previous time-step.



(a) Call Network (b) DBLP

(c) Last.fm band (d) Radoslaw

Figure 7.5: Precision@K over time

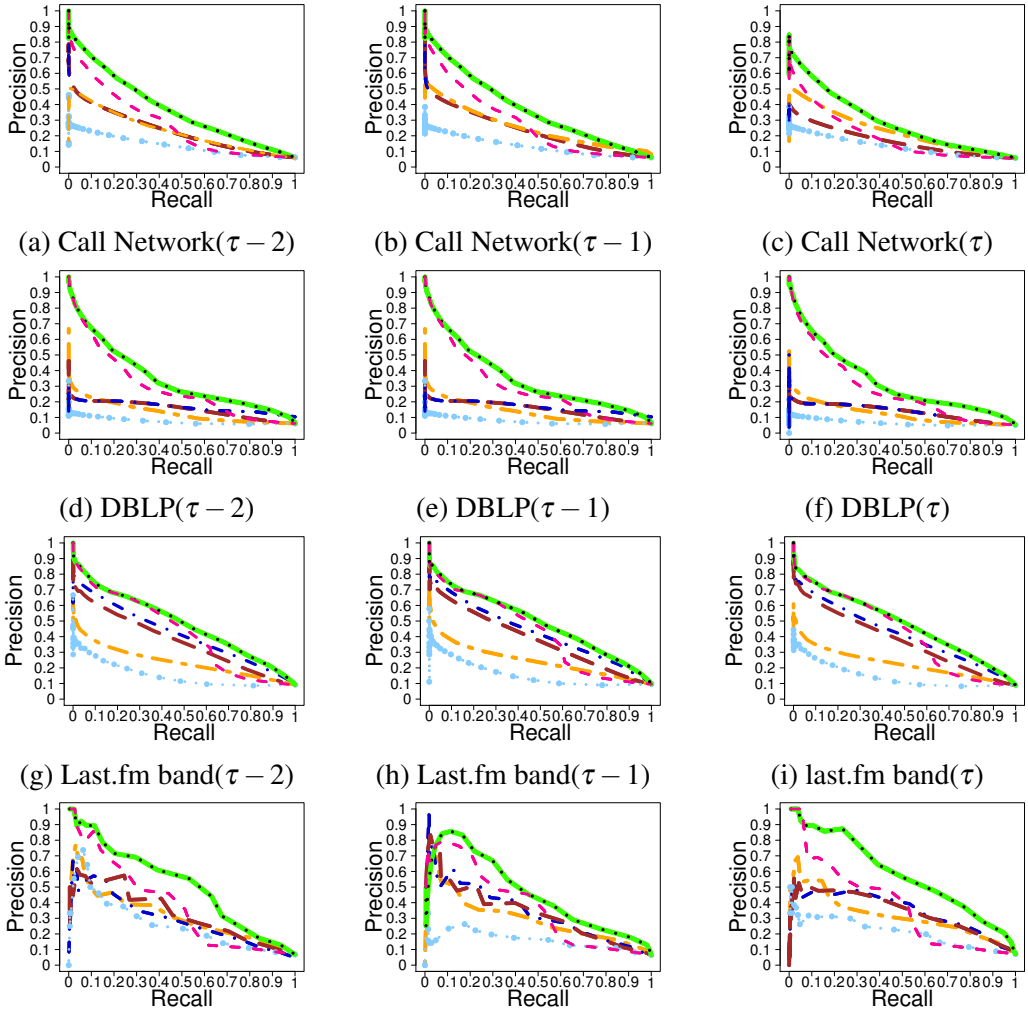
RLP_f RLP_{ex} *SLC* *LWS* *SS* *BRS-M* *RS-M*

For all the other algorithms the updated samples for every time-step are used for making predictions. Therefore we exhibit the results for every time-step in Figure 7.5. The loss function used in this plot is precision@k. In the Figure 7.6, we calculated the area under precision-recall curve (AUCPR) by computing the loss functions (precision and recall) multiple times for each time-step by varying thresholds. More detailed explanations for plots are given in the sections below. To facilitate the comprehensibility of experiments, we listed the abbreviations of models in Table 7.2

$$\varepsilon(\tau) = L(\hat{y}_e(\tau), y_e(\tau)); \text{ where } e = 1, 2, ...\hat{m} \tag{7.6}$$

7.4.2 Temporal evaluation of positive predictive value

To evaluate the preciseness of models over time we fixed the threshold such that the number of links to be predicted at every τ is constant K. Precision@K is a useful metric where they can be millions of predicted items but not all of them are necessary, for example in



(a) Call Network($\tau-2$) (b) Call Network($\tau-1$) (c) Call Network(τ)

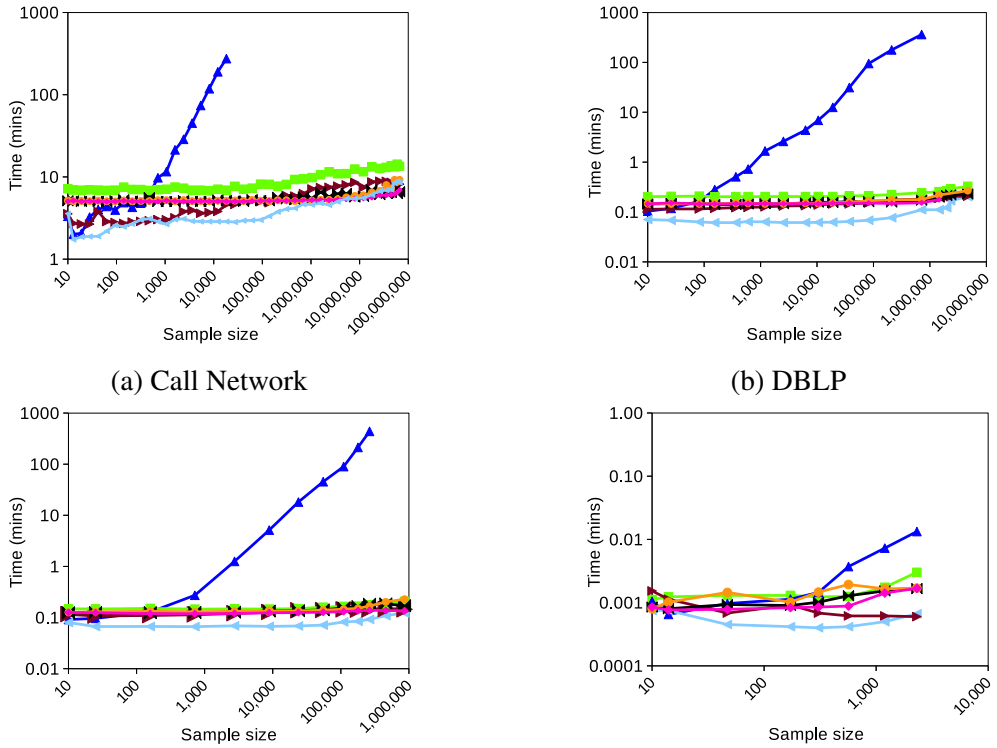
(d) DBLP($\tau-2$) (e) DBLP($\tau-1$) (f) DBLP(τ)

(g) Last.fm band($\tau-2$) (h) Last.fm band($\tau-1$) (i) last.fm band(τ)

(j) Radoslaw($\tau-2$) (k) Radoslaw($\tau-1$) (l) Radoslaw(τ)

Figure 7.6: PR-curves with varying thresholds for different models (⋯⋯) RLP_f, (——) RLP_{ex}, (- - -) *SLC*, (– –) *LWS*, (·····) *SS*, (— —) *BRS-M*, (●···) *RS-M*

the case of search queries, recommendations etc. Figure 7.5 shows the precision@100 over time in different data sets using all the models. To make the K value consistent across all the data sets per τ, we choose it not too small for the largest data set and not too high for the smallest data we have. For all the networks except DBLP, we see a low variance in precision over time. This difference is due to the network evolution trend in DBLP as can be seen in Figure 7.2. The growth of edges in DBLP is exponential over time, while in other networks is linear or constant. We observe poor results with the Radoslaw data (Figure 7.2d) because of lot of missing data. We have one or two time-steps missing every week in the temporal order of data where it shows dips in the performance. However, it is evident that RLP_f and RLP_{ex} exhibits the best average performance over



(a) Call Network (b) DBLP

(c) Last.fm band (d) Radoslaw

Figure 7.7: Running time by increasing output size $\hat{m}$ while the entire network over all time-steps is processed in a streaming way.

RLP_f RLP_{ex} *SLC* *LWS* *SS* *BRS-M* *RS-M*

time.

7.4.3 Prediction efficiency using PR-Curves

Precision-Recall curves evaluate the percentage of true positives among positive predictions with different threshold values. They are known to be more informative than the Receiver Operating Characteristic (ROC) curves for evaluating future classification performance in binary class imbalanced problems. They can be more reliable in this case of sparse data sets with very few links between pairs of nodes and heavily right-skewed distributions. We present in Figure 7.6 the PR-curves for the predictions of the last three time-steps. These results are achieved by performing multiple iterations with varying thresholds per τ. But for each iteration, the threshold is maintained across all models. For RLP_f and RLP_{ex} we present the curves for α with highest AUC-PR. Though the area under the curve for all the other values of α is still higher than that of other algorithms. From the figure, it is evident that *RLP* clearly outperforms in all the data sets and time steps.

7.4.4 Running time evaluation

In Figure 7.7, we report the running cost of all the algorithms. To plot the given figure all the networks were fully processed in the streaming way till the last time step while changing the sample size (m) stored by the algorithms in memory, which is given on x-axis. The experiments were performed on intel core an i7@3.40 GHz processor with 32GB memory. From the figure, it is evident that space saving exhibits a very slow performance in time and increasing exponentially with sample size. All the other algorithms show almost constant time performance even with the increase in size. Reservoir and random sampling are simple and fast techniques, and show the best performance up to a point for comparatively small data sets. As for the very large sample sizes the slope of reservoir and random sampling curves increases while RLP_f performs consistently. While RLP_f and RLP_{ex} gives equal results for ranking predictions in the above experiments, RLP_{ex} is a bit lagging in time efficiency.

7.4.5 Discussion

In this section, we summarized the results and presented possible explanations. We empirically show that our model using *RLP* gives better predictions than other baseline models referred in all the data sets. It performs well with an average precision as good as 100 % in the top 100 predictions as shown in Figure 7.5. *RLP* gives the best predictions with the lastfm.band bipartite data which indicates that it can be beneficial to improve the performance of recommendation models.

One of the crucial questions in the above experiments is the estimation of the value of forgetting factor α. We have observed that for all the values of α between 0 and 1, the results still outperform given baselines. For brevity, we could not display all. However we estimated the value of α using MLE in section 7.3.7 on recurrence probability given in Figure 7.4. The exponential curve is fit from $\tau - 1$ as RLP_f applies α from $\tau - 2$ to predict links in τ (Equation 7.2) while $\tau - 1$ is given weight 1. From our experiments, over time we have found that higher efficiency is attained around $\alpha \pm 0.1$ over all the time steps, but unique to a data set. $\alpha \pm 0.1$ indicated in Figure 7.4 gave the best predictions in the case of respective data sets.

We also notice that the simple random sampling model (*BRS-M*) performs better or as good as the higher time complexity space saving algorithm in finding top links (Figures 7.6, 7.5). It is also apparent that reservoir sampling (*RS-M*) performs very poorly in most of the cases.

Independent of the main results in this chapter, we also found that exploiting temporal information can be particularly beneficial for recurrent link prediction, which is in accor-

dance with previous studies of Tylenda et al. (2009). As *LWS* which is a static model performs weaker than other models exploiting temporal information.

7.5 Chapter Summary

Recurring links make a significant fraction of network links and keep on increasing, which is also empirically demonstrated in this work. We have also observed the probability of a link recurrence decreases exponentially as a function of time. Therefore in this chapter, we presented simple, space and time efficient heuristics to predict recurring links based on the past data, without exploiting the time and space expensive neighbourhood or global information of the network for each node/link. We proposed *RLP* which weights the links exponentially based on their frequency and recency in the stream. It is fast and scalable with $O(1)$ per edge and almost constant running time with the increase in sample size as seen empirically. We performed an extensive experimental comparison of the proposed method against existing link prediction algorithms, using massive real-world data sets from different domains. *RLP* exhibits high precision and recall over previously proven efficient method for recurring links prediction. We also extended and incorporated other state of the art algorithms such as space saving, reservoir sampling and random stream sampling algorithms for link prediction.

For future works, we would like to extend our model for new links' prediction. The analysis of a variety of networks with the perspective of recurrent links has not been carried out so far. We would try to advance the analysis of networks presented in the chapter to understand and model the growth of networks.

Chapter 8

Profiling High Leverage Users for Fraud Detection

Frauds in telephony incur huge revenue losses and cause a menace to both the service providers and legitimate users. Though the problem is growing with augmenting technologies, the works in this area are hindered by the availability of data and confidentiality of approaches. In this work, we deal with the fraud detection problem as generic with different types of unsolicited users. Most of the malicious users in telecommunications have some of the characteristics in common. These characteristics can be defined by a set of features whose values are uncommon for normal users. We made use of graph-based metrics to detect profiles that are significantly far from the common user profiles in a real data log with millions of users. To achieve this we looked for the high leverage points in the 99.9th percentile which identifies users as extreme anomalous points. Further, clustering these points helped distinguish malicious users efficiently and minimized the problem space significantly. Convincingly, the learned profiles of these detected users coincided with fraudulent behaviors.

8.1 Chapter Overview

Profiling is a technique to identify behavioral patterns of users based on some properties available in a specific context. They are also referred to as signatures or patterns in the literature (Alves et al., 2006; Ferreira et al., 2006). These profiles are usually constructed using data available from the past or current (direct or indirect) interactions with the system. The users with similar behavior can share a common profile with a smaller variance relative to dissimilar behaviors. User profiling is exercised in numerous applications of various domains for customization and improvement of services, decision making, recom-

mendation systems, etc. One of the fields where it was primarily applied was in telecommunications (Aghasaryan et al., 2010), where some of the applications include advertising products, promotions, segmentation of packages and catalogs and development of service infrastructure, etc. Besides these, user profiles have been persistently used in detecting fraudulent or unsolicited users (Kou et al., 2004).

Fraudsters realize the widespread use of telecommunications as a potential platform to target victims. The initiation of fraud over Internet technologies, specifically over the telephone networks is cheaper and easier than committing fraud in an offline society (Azad et al., 2018). In the telephone networks, fraudsters can target both the service providers as well as end-users. Frauds targeting service providers could be the form of bypassing a core network using illegal SIM boxes and excessive calling through the compromised gateways (Murynets et al., 2014; Azad et al., 2018). lately, the telephone channel has been the preferred method to target consumers directly by making the massive calling and convincing them of disclosing their private information through social engineering attacks and selling illegal products. Recent statistics on telephony frauds show that answering unsolicited calls would result in the wasting of 20 million man-hours which is equal to the loss of $475 million per year. Another study in 2018 (York, 2018) finds that 1 in every 10 US citizens became a victim of telephone fraud and some even fell twice. It is also estimated that telephone scammers managed to have the benefit of $357 per victim, with aggregated overall fraud benefit of approximately $8.9 billion in total losses (York, 2018). Federal Trade Communication (FTC) has estimated that every year scammers and spammers cause a loss of $8.6 billion annually to the citizens of USA due to frauds, with the majority of them initiated through the telephone (Gupta et al., 2018).

Though fraud detection is one of the important and expensive problems in telecommunications there has been very limited literature in this area (Mohammed Aamir et al., 2019). One of the reasons is the competitive environment which does not let the service providers disclose their models. Besides that the companies and organizations that have been defrauded refrain from revealing the situation due to reputation concerns (Hilas et al., 2015). This is also driven by the lack of public data availability due to privacy concerns.

Frauds are usually discovered from anomalies in data and patterns (Kou et al., 2004). The patterns of fraudulent users are unlike legitimate users in some specific features. Understanding those features that define the variability of users for a specific fraud is important. For example the feature ‘number of calls’ made by the user, which is highly applied in the literature for telecom fraud (Hilas et al., 2015; Hilas and Sahalos, 2005; Kaiafas et al., 2019; Miranda et al., 2019). This feature is significant in terms of many frauds (Gupta et al., 2015). Therefore, we characterize the behavior of frauds in terms of

common features to identify them. We make use of novel social network metrics (out-degree and indegree centralities, weighted outdegree and indegree centralities, recurrence ratio, average call duration and non-reciprocity) that are proved to be significant from our results. We then utilize them to find high leverage points and apply unsupervised learning to profile users. With the above approach we present the following contributions in our work:

- We propose a novel approach for detecting fraudulent users in telecommunication networks using high leverage points.
- Our method applies Mahalanobis distance, which is scale-invariant and considers the correlation between the variables in the data.
- We present new social network features that characterize an unsolicited user profile and evaluate them based on their contribution to it. We also found that the proposed metric recurrence ratio is significant in describing variations in our real data than call duration, which is highly used in fraud detection.
- We make use of network visualization techniques to illustrate the results and understand the patterns of communication between different types of users.

Rest of the chapter is organized as follows: In Section 8.2, we presented a brief review of previous works in this area. It is followed by Section 8.3, where we outlined the data-driven characteristics of telecom frauds which are found in the literature. In Section 8.4, we described the dataset, presented an analysis of the data and detailed the extracted features. The overview of the method for detecting high leverage points is presented in Section 8.5. Grouping the anomalous nodes and profiling their behavior is done in Section 8.6. The discussion about the results is carried out in Section 8.7. An illustration of network visualization is delineated in section 8.8. Finally the conclusions are drawn in Section 8.9.

8.2 Literature Review

Content-based approaches for fraud detection (Lentzen et al., 2011; Iranmanesh et al., 2012) are inconvenient as they need the call content which is not available beforehand. Lately, detecting call patterns from Call Detail Records (CDR) has been an alternative for timely fraud detection (Murynets et al., 2014). There are several feature-based frameworks or models discussed in previous research works that rely on statistics and machine learning. While in statistics the works proposed by Balasubramaniyan et al. (2007); Azad

and Morla (2013) use features such as the number of outgoing calls, the proportion of calls received and duration, based on which they assign a reputation score to the user. Depending on these scores the system may block users or classify them as fraud. In these works, the labels in the data were based on the assumptions on features. Additionally, Miranda et al. (2019) uses some known factors such as unallocated numbers, invalid number ranges, known fraudulent numbers, high-cost destinations etc, besides the derived features. These known factors are specific to the organizations and can be added in most of the models without a major increase in complexity.

In the last decade, there has been a significant increase in the works that make use of extracted features in semi-supervised and unsupervised learning algorithms which are referenced below. Most of them also include features such as time, call type (local or international), etc. besides the features mentioned above (Murynets et al., 2014; Yang et al., 2019). While not always the labeled data is available most of the works above use the crowd for labeling (Yang et al., 2019) or use the expert guidance for evaluating the users identified as malicious (Murynets et al., 2014). On the unsupervised side, Ferreira et al. (2006) employed distance measure between features to detect anomalies on exceeding a certain threshold and generate signature profiles of anomalous points. However, the evaluation was carried out only on two users. Advancement of the above work was presented by Alves et al. (2006) where after detecting the anomalous nodes the signatures were clustered to observe the pattern of users. However, the feature set was poor while only considering the duration and number of calls with different granularity and not considering social network metrics that capture the impact of relationships. Moreover, the results were mainly concentrated on the number of alarms raised with different thresholds while not validating the credibility of the users. Hilas et al. (2015) generated profiles of all the users, then considering them normal they detected changes in their behavior with the help of similarity scores. Kaiafas et al. (2019) presented an analysis of outlier detection techniques based on their effect from data normalization schemes, combination functions, and outlier detection algorithms.

The drawback of the anomaly detection techniques using distance-based clustering algorithms above is that the anomalies though make a little percentage of the data, they are still a considerable number as the data itself is massive. Therefore the anomalous points themselves make a cluster or join the clusters close enough. Moreover, it is unjustifiable to declare a cluster as fraudulent. Therefore we make use of the high leverage points which are extreme outliers to detect unsolicited users in our network.

8.3 Characteristics of Telephony Abuse

In this section, we present the previous works investigating frauds in telephony and characteristics of abuse that are focused on this work. The characteristics were observed from the data-driven approaches in (Balasubramaniyan et al., 2007; Gupta et al., 2015; d'Heureuse et al., 2011).

8.3.1 SPIT

Spam over internet telephony (SPIT) is sending unwanted calls or messages in bulk. As the description suggests they have a relatively very high number of outgoing calls in contrast to a legitimate user. This category includes phishing, vishing, telemarketers, etc. They try to cover as many recipients as possible and are likely to have short duration calls as most of the time their attempt is not successful (Balasubramaniyan et al., 2007; Gupta et al., 2015; d'Heureuse et al., 2011). The work of Balasubramaniyan et al. (2007) is based on the observation that the spammer's call pattern is largely unidirectional while for legitimate users it is bidirectional. That means the spammers receive zero or very fewer calls in comparison to the calls he makes. Not only previous works profile users based on these characteristics but use them to simulate or label data for supervised and semi-supervised learning (Balasubramaniyan et al., 2007; Azad and Morla, 2013).

8.3.2 TDoS

Telephony Denial of Service attacks is flooding of the system with a large number of calls which prevents other users from accessing the system. Again the motive hear is to make a large number of calls within a time frame to keep the system busy (Gupta et al., 2015).

8.3.3 Bypass Fraud

This type of fraud uses SIM boxes to divert the traffic (mostly international calls) over low-cost IP connections. The SIMs used in this endeavor is likely to have similar characteristics of a disproportionately large number of outgoing calls (Murynets et al., 2014). The duration is likely to differ as they are normal users whose calls are being forwarded. Nevertheless, these calls are also unidirectional to a large extent.

Though the above frauds are pursued with different motives they have the above characteristics in common. We exploit all the characteristics discussed above as features that are extracted from the data using social network metrics. Apart from these observations we also suppose that the percentage of repetitive calls to the same users is significantly low

or zero in the above forms of misuse. We call this a recurrence ratio which is explained in section 8.4. The recorded results in this work support our supposition.

8.4 Case Study

We make use of a massive anonymized Call Detail Records (CDR) data provided by a service provider. The data consists of 11,313,111 calls made by 3,486,492 users after removal of service calls which make a high number of the calls received such as emergency numbers etc. Calls in the data are also associated with time-stamps when the call was initiated and the duration in seconds for how long the call was active. The temporal distribution of calls is displayed in Figure 8.1. The plot shows there is a high activity in the day between 9 and 19 hours.

From the attributes available in the data we extract some features based on the social network properties of graph theory. The data spans over 24hrs, therefore, the update period or time period t of the values for the features is 1day. This process can be repeated for every time period. For brevity, we didn't mention t in the equations in the section below which by default means for every t. The features and their description is provided in the section below.

8.4.1 Network Features

In this work, we represent data in the form of a network. Therefore before we explain the derived features, it is important to explain the semantics of the network. A Network is used to represent objects and their relationships in the data. In this case, the objects are users, which are called nodes/vertices and a call between them represents an edge/link. The nodes can also be associated with attributes or features.

Recurring Links: The links in a call network are Recurring i.e occurring again and again over time. It is a unique link that occurs more than once. When a user repeats a call to the same user again over time it is a recurring link. With the above perspective of networks, we derive the following features. The generation of features was done in a streaming way as in Tabassum and Gama (2016b), to cope with the very large data.

8.4.1.1 Degree Centrality (*deg*)

The number of nodes directly connected to a node is said to be its degree centrality. In a call network, the number of adjacent users or unique calls made/received by a specific

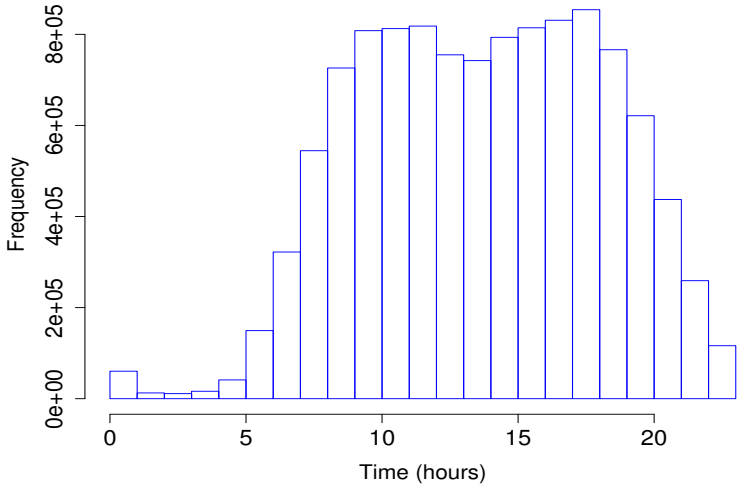


Figure 8.1: Temporal distribution of calls per hour.

user v is his degree centrality ($deg(v)$). To be more precise the number of nodes that make a call to the user in a time period t is called his indegree centrality (deg^-) and the number of nodes who receive a call from a user is called his outdegree centrality (deg^+). We calculated the indegree and outdegree centralities for every user in the network.

Weighted Degree Centrality ($wdeg$): Weighted degree centrality is defined as the sum of the weights attached to the ties connected to a node (Barrat et al., 2004). In our perspective of recurring links, the weight of a link is defined by the number of times it recurred over a time period t. The sum of weights over all the unique calls associated with the user in t is the Weighted Degree Centrality of that user. Specifically, the Weighted Indegree Centrality ($wdeg^-$) is the number of calls received by the user and Weighted Outdegree Centrality ($wdeg^+$) is the number of calls made by the user in t. This feature is not used directly but as an input for the recurrence, ratio explained below.

8.4.1.2 Inverse Recurrence Ratio (Γ)

Typically unsolicited users tend to call a large number of different users, unlike common users who call a group of users repeatedly. Therefore, it is essential to understand the inverse recurrence ratio for a user, which is the ratio of unique calls made by the total number of calls by him. It can be given in the form of an equation for every user v in the network as:

$$\Gamma = \frac{deg(v)}{wdeg(v)} \tag{8.1}$$

The value of Γ lies between 1 and 0 exclusive. When the ratio is 1 it indicates all the links of the user v are unique. If Γ gets closer to 0 the number of recurring links for v gets increasing. $\Gamma = 0$ indicates there is no link between two nodes, which is excluded as we consider the nodes only where an edge exists. In this work, we only use the recurring ratio of out-going links as in equation 8.2.

$$\Gamma = \frac{deg^{+}(v)}{wdeg^{+}(v)} \tag{8.2}$$

8.4.1.3 Average Call Duration $(AvgDur)$

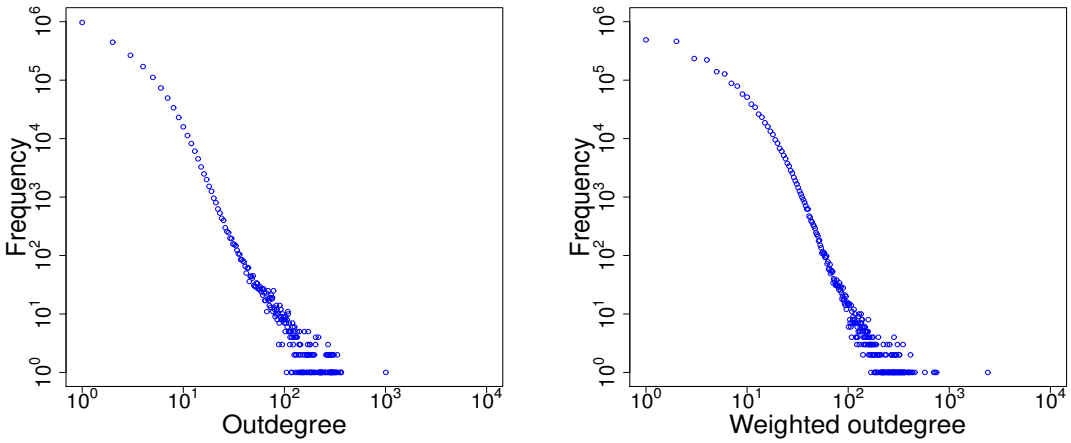
Average call duration is the mean of the duration of all calls made by the user v in a time period t. For every user in the data it is given as $AvgDur(v)$.

8.4.1.4 Non-reciprocity (r)

We give the non-reciprocity of a node in terms of weighted links as in Equation 8.3, which is the number of calls that did not get back to the user from the total number of calls he made. The focus here is on outdegree of users (as explained in the Section 8.3)

$$r = \frac{wdeg^{+}(v) - wdeg^{-}(v)}{wdeg^{+}(v)} \tag{8.3}$$

As most of the users don't get back to the calls from spammers or fraudsters the non-reciprocity of them can be high or equal to 1 in case no calls are received back. If all the number of calls done are equal to the number of calls received then the non-reciprocity reaches 0.

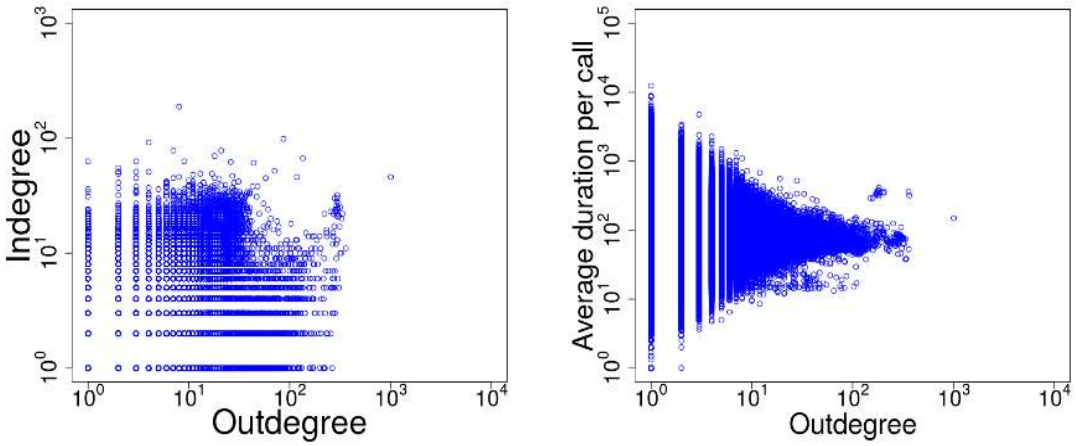


(a) Outdegree Distribution (b) Weighted Outdegree Distribution

Figure 8.2: The outdegree and weightedoutdegree distributions of call network

8.4.2 Call Network Analysis

To understand the underlying distribution and adjust the parameters in the experiments, we carried out an analysis of data that is depicted in this section. The outdegree distribution of the network is given in 8.2. For the long tail distributions given above, the use of the anomaly detection techniques such as box-plots results in almost 50% of the data falling above the third quartile as anomalies with the median between 2 and 3 degrees.



(a) Outdegree vs Indegree (b) Outdegree vs Average duration

Figure 8.3: Correlation between outdegree, indegree and Average duration

The relation between indegree, outdegree of nodes and the duration of calls is shown in Figure 8.3. Figure 8.3(a) portrays reciprocity. There are many sparse regions in both the analysis and we see no correlation between these variables. We also observe from the above figure, for the users making fewer calls, the duration has a high variance. While for the outliers making a high number of calls the duration is very tight.

To analyze the importance of features in the variability of data we carried out Principal Component Analysis (PCA). To avoid bias with the variables we scaled the data as explained below.

Scalling: Some features have large values while the others have smaller. In our data, the minimum outdegree of a node is 0 and a maximum outdegree is 1008, while the range of inverse recurrence ratio is between 0 and 1 and also in the case of other features. Therefore we applied feature scaling or z-scoring which normalizes the range of all the features by subtracting the mean of a feature and dividing the result by its standard deviation.

The PCA of the extracted features is given in Figure 8.4. One can observe that the outdegree and average duration are to some extent inversely correlated. The figure shows outdegree as the most important contributing variable in the components, while the in-

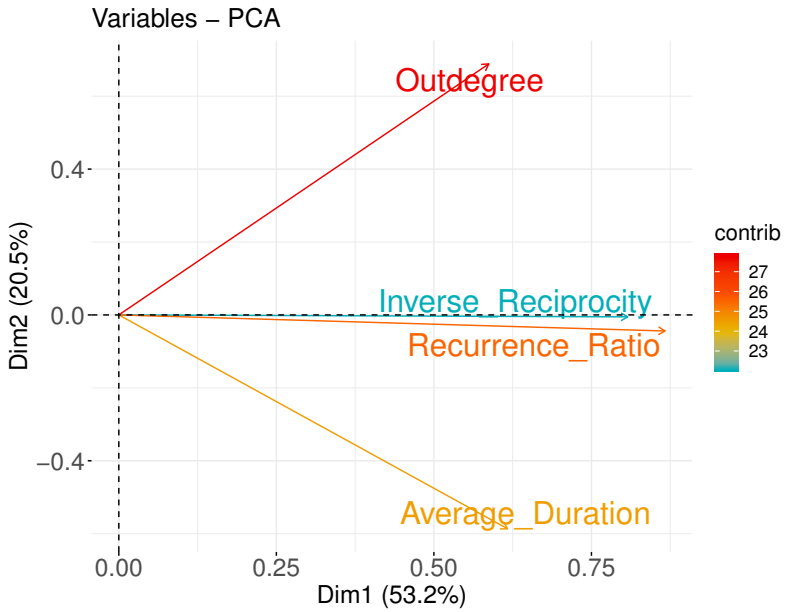


Figure 8.4: Pictorial representation of variables in the given principal components.

verse recurrence ratio follows it. The average duration is less significant than the inverse recurrence ratio while considering the real data we have.

8.5 Identifying Anomalous Nodes

We use the features discussed in the above section to find the anomalous users from the network. **Anomalous users/nodes** are those whose behavior significantly deviates from the common behavior of other nodes in the network. These form patterns that raise suspicion.

8.5.1 Detecting High Leverage Nodes

High leverage points are the outlying and extreme observations in the data and lack neighboring observations (Everitt and Skrondal, 2002). High leverage points are influential in the sense they greatly affect the distribution of independent variables. In our case, we consider them for fraud detection as they are far from the common behavior. We found these high leverage nodes from the anomalous points detected using Robust Mahalanobis distance.

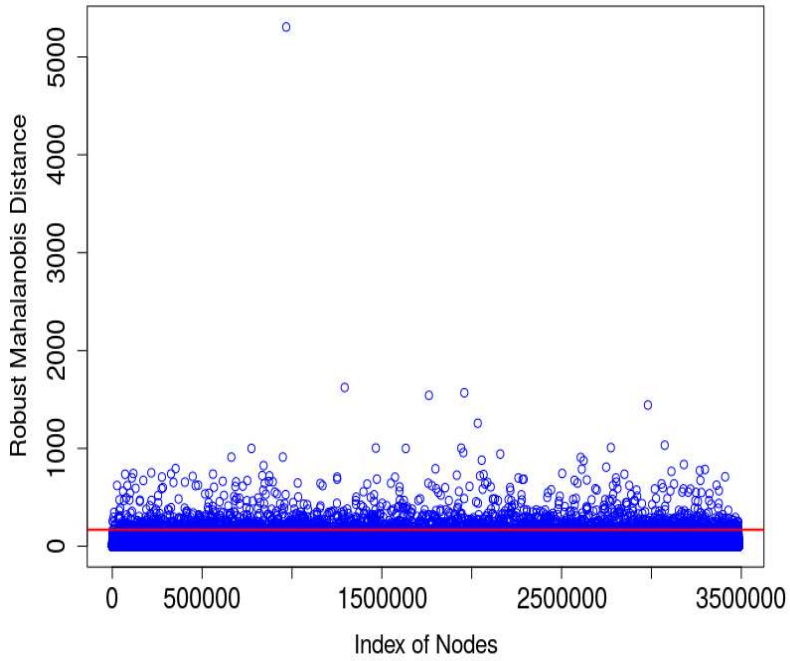


Figure 8.5: Robust Mahalanobis distances of nodes in the network from the centroid and the red line marks the cutoff.

8.5.2 Mahalanobis Distance

Mahalanobis Distance is a statistical metric used to find the distance of a point from a data distribution, based on its measurements in multiple dimensions (Mahalanobis, 1936). For a vector of nodes $\overrightarrow{v} = \{v_1, v_2, v_3, ...v_N\}^T$ with a mean $\overrightarrow{\mu} = \{\mu_1, \mu_2, \mu_3, ...\mu_N\}^T$ and a covariance matrix S from multiple features is given as Equation 8.4.

$$D^2 = (\overrightarrow{v} - \overrightarrow{\mu})^T S^{-1} (\overrightarrow{v} - \overrightarrow{\mu}) \tag{8.4}$$

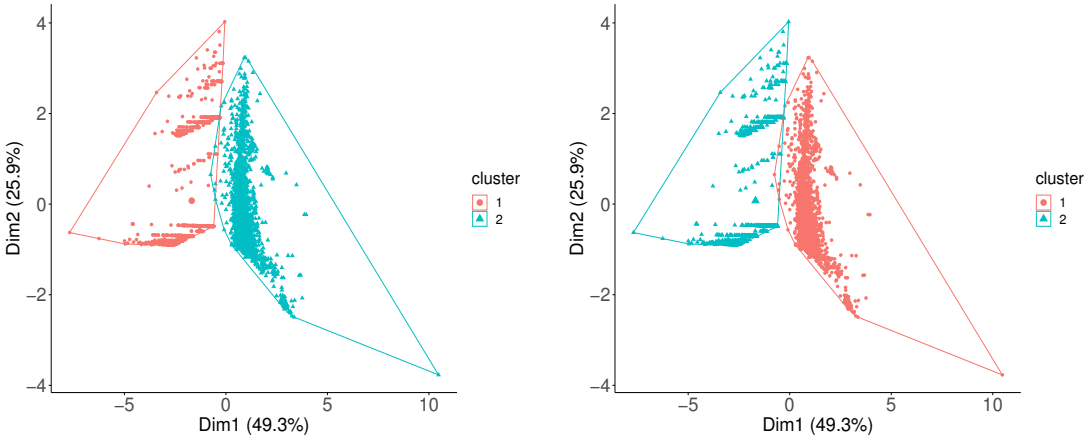
Leverage is closely related to Mahalanobis distance (Schinka et al., 2013). High leverage points are at the farthest from the centroid with high Mahalanobis distances. These distances of the nodes using robust Minimum Covariance Determinant (MCD) (Varmuza and Filzmoser, 2016) is shown in Figure 8.5. To find the extreme outliers we use a cutoff of a fraction of 1-percentile of data, around 0.001, which is depicted as a red line in the given figure. The number of high leverage points detected is 3,450 from the total number of 3,486,492 points. The cutoff can be varied depending on the number of points to be analyzed with a trade-off of memory size.

8.6 Behavioral Profiling

We see that the outliers detected by the Mahalanobis distance have different behaviors. This is due to negative correlation between the variables. To identify the profiles of high leverage outliers who are similar, we apply clustering algorithms.

8.6.1 Clustering

Clustering is a task of grouping data with fewer distances between group members than other groups. By clustering the high leverage points above we do not have to use the entire data for clustering and the profiles of clusters are specific to them. We apply two distance-based clustering algorithms K-means (Hartigan and Wong, 1979) and K-medoids (Reynolds et al., 2006; Schubert and Rousseeuw, 2019).



(a) K-means (b) [K-medoids

Figure 8.6: Clustering of high leverage users

K-means is a popular unsupervised learning algorithm to find patterns in the data. It assigns points to clusters randomly. Then over each iteration minimizes the sum of squares between the points and the center (mean) of assigned clusters. Finally converges giving clusters with similar points. For the experiments, we employed the Kmeans implementation of stats package in R. Any clustering algorithm that works well with sparse data can be applied at this stage. The results of the clusters are shown in Figure 8.6.

K-medoids works very similar to K-means. While K-means uses the means of cluster points as the centers, K-medoids chooses one of the data points as the centroids. These data points are chosen based on their average dissimilarity to other points in their cluster. K-medoids is known to be more robust to noise than K-means (Arora et al., 2016). R

implementation of PAM in the cluster package was used for the experiments. Figure 8.6b represents the clusters using PCA.

Both the clustering algorithms above need an input parameter for the number of clusters. To find the best number of clusters we extracted the Silhouette information by increasing the number of clusters (k). For every $k > 2$ the Silhouette coefficients for the clusters turned negative showing they are assigned to wrong clusters. The time complexity of these algorithms is $O(n*d*k)$. Where n is the number of nodes and d is the number of dimensions and k is the number of clusters. By using high leverage points we decrease k, and n to a significant extent.

Table 8.1: Profile of two groups of high leverage points

Algorithm	Profile	Centroid	Cluster Size	Outdegree	Inverse Recurrence	Non-Reciprocity	Average Duration
K-means	1	mean	2274	69.028	0.816	0.935	78.135
	2	mean	1176	1.113	0.848	0.227	2136.5
K-medoids	1	medoid	2274	55	0.932	1	71.779
	2	medoid	1176	1	1.0	0	1860.0

8.7 Discussion

The two groups are apparent from the results of both the algorithms and the points exactly belong to the same clusters. This is because there is a concrete discrepancy between the outliers detected. Therefore, it leads to two dissimilar profiles identified from the clusters which are given in Table. Profile 2 represents a normal profile where the users make around one call but a very high duration not common with all the users. The users close to profile 1 (cluster 2 in figure 8.6a and cluster 1 in figure 8.6b) on an average make calls to around 70 recipients in one day of which 82% calls are unique and 94% do not call them back. The duration of these calls is less with an average of 78 seconds per call. This profile coincides with the behavior of unsolicited users such as spam, phishing, and telemarketers as outlined in section 8.3. The inverse recurrence ratio of this profile is close to 1, which supports our assumption in section 8.3. This set of identified users can be further scrutinized for a specific fraud or raise an alert to the recipient. The profiles can be updated for every t. On clustering only the anomalous users above the cutoff (in section 8.5), would decrease false positives besides saving space and computations.

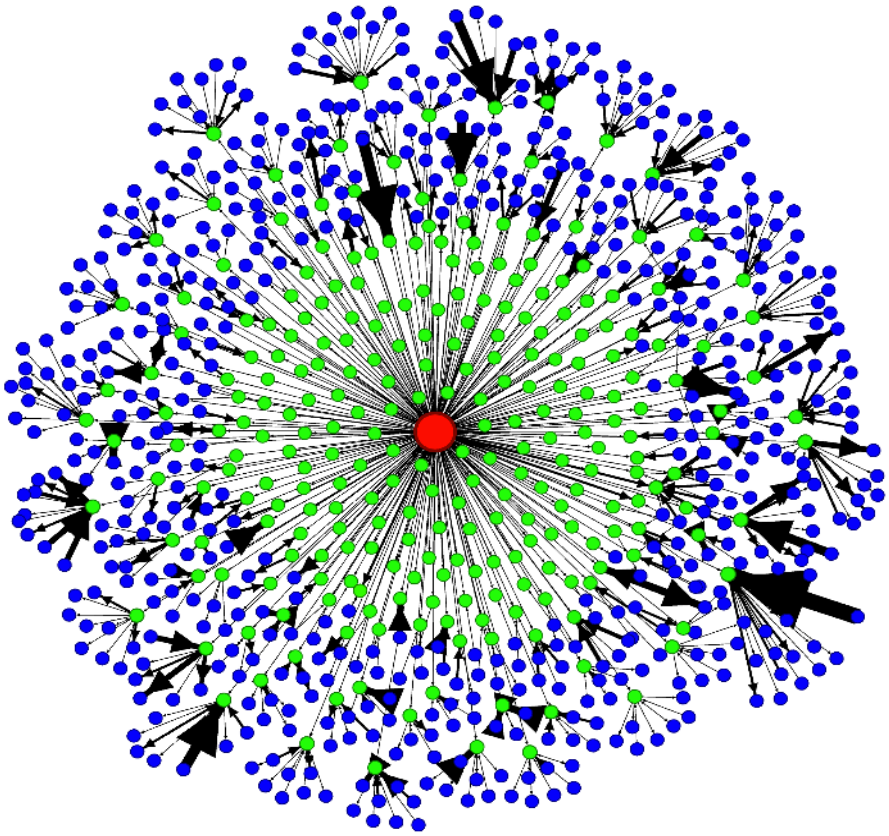


Figure 8.7: An ego network (two levels) of a high leverage user (red). Green indicates common users and the blue nodes represent their networks.

8.8 Social Network Visualisation

To understand the patterns of interactions or relationships between fraudulent users and legitimate users we applied network visualization techniques which is depicted in Figure 8.7. The red node represents an anomalous user identified from our approach. It has only outgoing edges connecting to green nodes which have a normal behavior falling very close to the mean of the distribution. The blue nodes represent the neighbors of green nodes which are not directly connected to the red node. The thickness of the edge represents its weight/strength and the size of the node delineates its degree centrality. As the inverse recurrence ratio of the red node is very high it has negligible links to which it makes repetitive calls. Therefore it is hard to observe any strong links that connect to it. On the contrary, all the other users have low degree centrality and the strength of their connections is high, showing recurrence. This explains the users with the normal profile have a significant proportion of strong connections to whom they make repeated calls while it is quite different in the outliers. From the perspective of green nodes in the figure, we also notice that they only make strong connections with the blue nodes i.e other

legitimate users but not with the red node. There are also no connections between any two green nodes in the figure. That means they are no triangles that form a closed group with the red node.

8.9 Chapter Summary

Call networks are very large networks that need computationally feasible methods for fraud detection. In this work, we present the approach for detecting fraudulent users in a massive network of calls. Our approach uses the clustering of high leverage points in order to save the complexity of clustering algorithms. The results show the generated profiles that uniquely identify and comply with the characteristics of fraudulent behaviors in the literature. We also observe that the two clustering algorithms perform exactly similar in this case as there is a clear distinction of points discovered using our approach. However, the approach is not limited to the outlier detection and clustering techniques used in this work. Nevertheless, the methods should be able to handle multiple variables.

We also propose new sociometric features that have a unique range of values for anomalous user profiles. Additionally, we present an analysis that shows the correlations and patterns in the data. Besides that, the network visualization techniques clearly illustrate the behavior of anomalous nodes.

Part IV

Epilogue

Chapter 9

Research Summary

Here we conclude this dissertation by revisiting the contributions and positioning them in the respective area of research. The contributions of this dissertation are mainly focused in three areas, i.e social network analysis, sampling and mining for large streaming networks. The networks can be evolving or have reccurring links over time. The methods, approaches and techniques presented in this dissertation can be directly applied or can be input or part of other models. In the Analytics part of this thesis, we answered questions like, how to process streaming networks and analyze them incrementally. We also proposed sampling approaches as an alternative to analysing networks incrementally.

Previous sampling techniques do not take into consideration the current demands of real-time evolving networks and recurring links. We draw some important observations while exploiting other sampling techniques, such as the classic reservoir sampling approach is severely biased to very low degree nodes 4 - 7. We proposed a BRS method which is not only simple and fast but super-linearly maintains latest information without having to keep track of order as in sliding windows. When applied on recurring link networks it also maintains frequency of edges. BRS and SS sample stronger communities with high average degree centralities. That shows a better community/clustering structure when compared to RS. Therefore BRS and SS would be more suitable for applications analyzing community structure.

We proposed a one pass memory less dynamic sampling approach SBias for recurring links or weighted networks. Unlike other network sampling approaches it does not maintain nodes with high degree and keep track of their edges. Instead, it preserves edges with high frequency which eventually leads to nodes with high degree. This is also seen empirically from the results. These samples are less biased to low degree nodes contrary to traditional sampling methods. They retain better cluster structure and maintain temporal distribution. The samples also gets close spanning tree as it gets rid of redundant edges.

In this thesis we have defined the recurring links' networks over time and learnt their importance in most of the real world networks. In Chapter 7 we see that the significant fraction of links reoccur and sometimes as high as 100% of the links. We also notice that the recurrence probability of links increases with the decrease of distance in time. While considering the above aspects in a temporal setting we proposed a fast and memory less streaming model for predicting recurring links. Most of the existing algorithms for link prediction have not been evaluated for recurring links, the one's that have been, performed poorly as we refer in Chapter 7. We found our method to perform substantially better than the method that was proven efficient in the literature. For experiments, we adopted prequential evaluation over diversified datasets.

Fraud detection is an another important application of graph analytics. It is an expensive and prevalent problem, moreover evolving with the advancements of technology. While outlining the limitations of current literature, we also observed a set of common characteristics among a number of frauds. These characteristics are based on graph theoretical properties. We extracted them from the CDR networks by stream processing methodology for computing incremental analytics in Chapter 3. Further, we presented a model for detecting extreme outliers whose profiles match the fraudulent behaviors.

Additionally, we made use of visualisation techniques to comprehend and interpret the insights and patterns from user behaviors.

Bibliography

"10 data and analytics trends for 2020 by gartner," *informatech*, Nov 2019. [Online]. Available: https://www.informationweek.com/big-data/ai-machine-learning/10-data-and-analytics-trends-for-2020/d/d-id/1336310

"5 innovative ways to use graph analytics," *Oracle Big Data*, June 2018. [Online]. Available: https://medium.com/oracledevs/5-innovative-ways-to-use-graph-analytics-bacc4f2be521

"A comprehensive guide to real-time big data analytics," *ScienceSoft*, Apr 2018. [Online]. Available: https://www.scnsoft.com/blog/real-time-big-data-analytics-comprehensive-guide

"The digitization of the world from edge to core," *An IDC White Paper – US44413318*, Nov 2018. [Online]. Available: https://www.seagate.com/pt/pt/our-story/data-age-2025/

"Data set to grow 10-fold by 2020 as internet of things takes off," *computerweekly.com*, Apr 2014. [Online]. Available: http://www.computerweekly.com/news/2240217788/Data-set-to-grow-10-fold-by-2020-as-internet-of-things-takes-off

M. A. Abbasi, J. Tang, and H. Liu, "Scalable learning of users' preferences using networked data," in *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, ser. HT '14. New York, NY, USA: ACM, 2014, pp. 4–12.

A. Abraham, A.-E. Hassanien, V. Sná *et al.*, *Computational social network analysis: Trends, tools and research advances*. London: Springer Science & Business Media, 2009.

E. Acar, D. M. Dunlavy, and T. G. Kolda, "Link prediction on evolving data using matrix and tensor factorizations," *ICDM Workshops 2009 - IEEE International Conference on Data Mining*, pp. 262–269, 2009.

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

C. Aggarwal and K. Subbian, "Evolutionary network analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 10, 2014.

C. C. Aggarwal, *Social Network Data Analytics*, 1st ed. Springer Publishing Company, Incorporated, 2011.

C. C. Aggarwal, "On biased reservoir sampling in the presence of stream evolution," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 607–618.

C. C. Aggarwal, *Data streams: models and algorithms*. Springer Science & Business Media, 2007, vol. 31.

C. C. Aggarwal, "Models for incomplete and probabilistic information," in *Managing and Mining Uncertain Data*. Springer, 2009, pp. 1–34.

C. C. Aggarwal and K. Subbian, "Event detection in social streams." in *SDM*, vol. 12. SIAM, 2012, pp. 624–635.

C. C. Aggarwal and K. Subbian, "Event detection in social streams," in *12th SIAM International Conference on Data Mining, USA.*, 2012, pp. 624–635.

C. C. Aggarwal, H. Wang *et al.*, *Managing and mining graph data*. Springer, 2010, vol. 40.

A. Aghasaryan, M.-P. Dupont, and S. Betge-Brezetz, "Applications for telecommunications services user profiling," Jun. 3 2010, uS Patent App. 12/447,593.

T. Ågotnes, "Mec-monitoring clusters' transitions," in *Stairs 2010: Proceedings of the Fifth Starting AI Researchers' Symposium*, vol. 222. Amsterdam: IOS Press, 2010, p. 212.

N. K. Ahmed, J. Neville, and R. Kompella, "Space-efficient sampling from social activity streams," in *Proceedings of the 1st international workshop on big data, streams and heterogeneous source mining: algorithms, Systems, Programming Models and Applications*. ACM, 2012, pp. 53–60.

N. K. Ahmed, J. Neville, and R. Kompella, "Network sampling: From static to streaming graphs," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 2, p. 7, 2014.

N. K. Ahmed, N. Duffield, T. L. Willke, and R. A. Rossi, "On sampling from massive graph streams," *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1430–1441, 2017.

W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. New York, NY: Acm, 2000, pp. 171–180.

E. M. Airoldi and K. M. Carley, "Sampling algorithms for pure network topologies: a study on the stability and the separability of metric embeddings," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 13–22, 2005.

C. G. Akcora and E. Ferrari, "Discovering trust patterns in ego networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. New York, NY: IEEE, 2014, pp. 224–229.

L. Akoglu and B. Dalvi, "Structure, tie persistence and event detection in large phone and sms networks," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*. ACM, 2010, pp. 10–17.

L. Akoglu and C. Faloutsos, "Event detection in time series of mobile communication graphs," in *Army Science Conference*, 2010, pp. 77–79.

L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2010, pp. 410–421.

L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

M. Al Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social network data analytics*. Springer, 2011, pp. 243–275.

M. Al Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social network data analytics*. Springer, 2011, pp. 243–275.

M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

R. Albert and A.-L. Barabási, "Topology of evolving networks: local events and universality," *Physical review letters*, vol. 85, no. 24, p. 5234, 2000.

R. Albert, H. Jeong, and A.-L. Barabási, "Internet: Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.

U. Alon, "Biological networks: the tinkerer as an engineer," *Science*, vol. 301, no. 5641, pp. 1866–1867, 2003.

R. Alves, P. Ferreira, O. Belo, J. Lopes, J. Ribeiro, L. Cortesão, and F. Martins, "Discovering telecom fraud situations through mining anomalous behavior patterns," in *Proceedings of the DMBA Workshop on the 12th ACM SIGKDD*, 2006.

K. Anand, J. Kumar, and K. Anand, "Anomaly detection in online social network: A survey," in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*. IEEE, 2017, pp. 456–459.

J. R. Anderson and L. J. Schooler, "Reflections of the environment in memory," *Psychological science*, vol. 2, no. 6, pp. 396–408, 1991.

V. Arnaboldi, M. Conti, A. Passarella, and F. Pezzoni, "Ego networks in twitter: an experimental analysis," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 3459–3464.

P. Arora, S. Varshney *et al.*, "Analysis of k-means and k-medoids algorithm for big data," *Procedia Computer Science*, vol. 78, pp. 507–512, 2016.

L. Averell and A. Heathcote, "The form of the forgetting curve and the fate of memories," *Journal of Mathematical Psychology*, vol. 55, no. 1, pp. 25–35, 2011.

M. A. Azad and R. Morla, "Caller-rep: Detecting unwanted calls with caller social strength," *Computers & Security*, vol. 39, pp. 219–236, 2013.

M. A. Azad, R. Morla, and K. Salah, "Systems and methods for spit detection in voip: Survey and future directions," *Computers & Security*, vol. 77, pp. 1–20, 2018.

B. Babcock, M. Datar, and R. Motwani, "Sampling from a moving window over streaming data," in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2002, pp. 633–634.

V. Balasubramaniyan, M. Ahamad, and H. Park, "Callrank: Combating spit using call duration, social networks and global reputation." in *CEAS*, 2007.

A.-L. Barabási, *Network science*. New York, NY, USA: Cambridge University Press, 2016.

A.-L. Barabási, *Network science*. Cambridge University Press, 2016.

A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical mechanics and its applications*, vol. 311, no. 3, pp. 590–614, 2002.

B. A.-L. Barabási and E. Bonabeau, "Scale-free," *Scientific American*, vol. 288, no. 5, pp. 50–59, 2003.

A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the national academy of sciences*, vol. 101, no. 11, pp. 3747–3752, 2004.

P. Barson, S. Field, N. Davey, G. McAskie, and R. Frank, "The detection of fraud in mobile phone networks," *Neural Network World*, vol. 6, no. 4, pp. 477–484, 1996.

M. Bastian, S. Heymann, M. Jacomy *et al.*, "Gephi: an open source software for exploring and manipulating networks." *Icwsm*, vol. 8, pp. 361–362, 2009.

N. Benchettara, R. Kanawati, and C. Rouveirol, "Supervised machine learning applied to link prediction in bipartite social networks," in *2010 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2010, pp. 326–330.

A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, "Copycatch: stopping group attacks by spotting lockstep behavior in social networks," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 119–130.

A. Bifet, G. Holmes, B. Pfahringer, and R. Gavaldà, "Mining frequent closed graphs on evolving data streams," in *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11, 2011, pp. 591–599.

M. Bilgic and L. Getoor, "Effective label acquisition for collective classification," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 43–51.

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

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

R. S. Bogartz, "Evaluating forgetting curves psychologically." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 16, no. 1, p. 138, 1990.

P. Bonacich, "Power and centrality: A family of measures," *American journal of sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

E. U. Bond III, B. A. Walker, M. D. Hutt, and P. H. Reingen, "Reputational effectiveness in cross-functional working relationships," *Journal of Product Innovation Management*, vol. 21, no. 1, pp. 44–60, 2004.

S. P. Borgatti, "Structural holes: Unpacking burt's redundancy measures," *Connections*, vol. 20, no. 1, pp. 35–38, 1997.

S. P. Borgatti and M. G. Everett, "Network analysis of 2-mode data," *Social networks*, vol. 19, no. 3, pp. 243–269, 1997.

S. P. Borgatti, M. G. Everett, and L. C. Freeman, "Ucinet for windows: Software for social network analysis," 2002.

B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, pp. 163–177, 2001.

R. L. Breiger, S. A. Boorman, and P. Arabie, "An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling," *Journal of mathematical psychology*, vol. 12, no. 3, pp. 328–383, 1975.

S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Computer networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

B. Bringmann, M. Berlingerio, F. Bonchi, and A. Gionis, "Learning and predicting the evolution of social networks," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 26–35, 2010.

A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," *Computer networks*, vol. 33, no. 1, pp. 309–320, 2000.

C. Buntain and J. Lin, "Burst detection in social media streams for tracking interest profiles in real time," in *39th International ACM SIGIR conference*, 2016.

L. S. Buriol, G. Frahling, S. Leonardi, A. Marchetti-Spaccamela, and C. Sohler, "Computing clustering coefficients in data streams," in *European Conference on Complex Systems (ECCS)*, 2006.

R. S. Burt, *Structural holes: The social structure of competition*. Harvard university press, 2009.

C. Cao, Q. Ni, and Y. Zhai, "An improved collaborative filtering recommendation algorithm based on community detection in social networks," in *Proceedings of the 2015 annual conference on genetic and evolutionary computation*. ACM, 2015, pp. 1–8.

C. Cattuto, M. Quaggiotto, A. Panisson, and A. Averbuch, "Time-varying social networks in a graph database: a neo4j use case," in *First International Workshop on Graph Data Management Experiences and Systems*. New York, NY: ACM, 2013, p. 11.

T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 4, p. 54, 2017.

V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

D. H. Chau, S. Pandit, and C. Faloutsos, "Detecting fraudulent personalities in networks of online auctioneers," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2006, pp. 103–114.

S. Chaudhuri, R. Motwani, and V. Narasayya, "On random sampling over joins," in *ACM SIGMOD Record*, vol. 28, no. 2. ACM, 1999, pp. 263–274.

N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

H.-H. Chen, D. J. Miller, and C. L. Giles, "The predictive value of young and old links in a social network," in *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*. ACM, 2013, pp. 43–48.

S. Choobdar, P. Ribeiro, and F. Silva, "Event detection in evolving networks," in *Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on*. IEEE, 2012, pp. 26–32.

A. Clauset, M. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, p. 066111, 2004.

A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

A. Clauset, C. Moore, and M. E. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

W. G. Cochran, *Sampling techniques*. John Wiley & Sons, 2007.

T. F. Coleman and J. J. Moré, "Estimation of sparse jacobian matrices and graph coloring blems," *SIAM journal on Numerical Analysis*, vol. 20, no. 1, pp. 187–209, 1983.

D. Combe, C. Largeron, E. Egyed-Zsigmond, and M. Géry, "A comparative study of social network analysis tools," *Social Networks*, vol. 2, pp. 1–12, 2010.

M. Cordeiro and J. Gama, *Online Social Networks Event Detection: A Survey*. Cham: Springer International Publishing, 2016, pp. 1–41.

M. Cordeiro, R. P. Sarmento, and J. Gama, "Dynamic community detection in evolving networks using locality modularity optimization," *Social Network Analysis and Mining*, vol. 6, no. 1, p. 15, 2016.

M. Cordeiro, R. P. Sarmento, P. Brazdil, and J. Gama, "Evolving networks and social network analysis methods and techniques," *Social Media and Journalism: Trends, Connections, Implications*, p. 101, 2018.

G. Cormode, V. Shkapenyuk, D. Srivastava, and B. Xu, "Forward decay: A practical time decay model for streaming systems," in *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*. IEEE, 2009, pp. 138–149.

G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine, "Synopses for massive data: Samples, histograms, wavelets, sketches," *Foundations and Trends in Databases*, vol. 4, no. 1–3, pp. 1–294, 2012.

L. d. F. Costa, O. N. Oliveira Jr, G. Travieso, F. A. Rodrigues, P. R. Villas Boas, L. Antiqueira, M. P. Viana, and L. E. Correa Rocha, "Analyzing and modeling real-world phenomena with complex networks: a survey of applications," *Advances in Physics*, vol. 60, no. 3, pp. 329–412, 2011.

C. A. Coulson, "Present state of molecular structure calculations," *Reviews of Modern Physics*, vol. 32, no. 2, p. 170, 1960.

P. R. da Silva Soares and R. B. C. Prudêncio, "Time series based link prediction," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. New York, NY: IEEE, 2012, pp. 1–7.

N. Dakiche, F. B.-S. Tayeb, Y. Slimani, and K. Benatchba, "Tracking community evolution in social networks: A survey," *Information Processing & Management*, vol. 56, no. 3, pp. 1084–1102, 2019.

K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjea, A. A. Nanavati, and A. Joshi, "Social ties and their relevance to churn in mobile telecom networks," in *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*. New York, NY: ACM, 2008, pp. 668–677.

P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY: ACM, 2001, pp. 57–66.

S. N. Dorogovtsev and J. F. Mendes, "Scaling properties of scale-free evolving networks: Continuous approach," *Physical Review E*, vol. 63, no. 5, p. 056125, 2001.

J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical review E*, vol. 72, no. 2, p. 027104, 2005.

D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 2, p. 10, 2011.

A. Dutot, Y. Pigné, and F. Guinand, "The graphstream java dynamic graph library."

N. d'Heureuse, S. Tartarelli, and S. Niccolini, "Analyzing telemarketer behavior in massive telecom data records," in *Trustworthy Internet*. Springer, 2011, pp. 261–271.

D. Easley and J. Kleinberg, *Networks, Crowds and Markets: Reasoning about a Highly Connected World*. New York, NY: Cambridge University Press, 2010.

H. Ebbinghaus, *Memory: A contribution to experimental psychology*. University Microfilms, 1913, no. 3.

H. Ebbinghaus, *Urmanuskript "Ueber das Gedächtniss" 1880*. Passau: Passavia Universitätsverlag, 1983.

P. S. Efraimidis, "Weighted random sampling over data streams," in *Algorithms, Probability, Networks, and Games*. Springer, 2015, pp. 183–195.

A. Epasto, S. Lattanzi, V. Mirrokni, I. O. Sebe, A. Taei, and S. Verma, "Ego-net community mining applied to friend suggestion," *Proceedings of the VLDB Endowment*, vol. 9, no. 4, pp. 324–335, 2015.

P. Erdos and A. Renyi, "On the evolution of random graphs," *Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.

W. K. Estes, "The problem of inference from curves based on group data." *Psychological bulletin*, vol. 53, no. 2, p. 134, 1956.

M. Everett and S. P. Borgatti, "Ego network betweenness," *Social networks*, vol. 27, no. 1, pp. 31–38, 2005.

B. Everitt and A. Skrondal, *The Cambridge dictionary of statistics*. Cambridge University Press Cambridge, 2002, vol. 44.

J. Fairbanks, D. Ediger, R. McColl, D. A. Bader, and E. Gilbert, "A statistical framework for streaming graph analysis," in *IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, ser. ASONAM '13, 2013, pp. 341–347.

M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *ACM SIGCOMM computer communication review*, vol. 29, no. 4. New York, NY: ACM, 1999, pp. 251–262.

S. d. S. Fernandes, H. F. Tork, and J. M. P. d. Gama, "The initialization and parameter setting problem in tensor decomposition-based link prediction," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Los Alamitos, CA, 2017, pp. 99–108, © 2017 IEEE.

P. Ferreira, R. Alves, O. Belo, and L. Cortesão, "Establishing fraud detection patterns based on signatures," in *Industrial Conference on Data Mining*. Springer, 2006, pp. 526–538.

S. Fortunato, "Community detection in graphs," *Physics Report*, vol. 486, pp. 75–174, 2010.

L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

L. C. Freeman, "Centered graphs and the structure of ego networks," *Mathematical Social Sciences*, vol. 3, no. 3, pp. 291–304, 1982.

L. C. Freeman, "Turning a profit from mathematics: The case of social networks," *Journal of Mathematical Sociology*, vol. 10, no. 3-4, pp. 343–360, 1984.

B. Furht, *Handbook of social network technologies and applications*. New York, NY: Springer Science & Business Media, 2010.

J. Gama, *Knowledge discovery from data streams*. CRC Press, 2010.

J. Gama, *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 2010.

J. Gama and M. M. Gaber, *Learning from data streams*. Springer, 2007.

J. Gama, R. Sebastião, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Machine learning*, vol. 90, no. 3, pp. 317–346, 2013.

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

J. O. Garcia, A. Ashourvan, S. Muldoon, J. M. Vettel, and D. S. Bassett, "Applications of community detection techniques to brain graphs: Algorithmic considerations and implications for neural function," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 846–867, 2018.

D. Garlaschelli and M. I. Loffredo, "Patterns of link reciprocity in directed networks," *Physical review letters*, vol. 93, no. 26, p. 268701, 2004.

L. Getoor and C. P. Diehl, "Link mining: A survey," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

F. Göbel and A. Jagers, "Random walks on graphs," *Stochastic processes and their applications*, vol. 2, no. 4, pp. 311–336, 1974.

L. A. Goodman, "Snowball sampling," *The annals of mathematical statistics*, pp. 148–170, 1961.

M. Granovetter, *Getting a job: A study of contacts and careers*. Chicago, IL: University of Chicago Press, 1995.

M. S. Granovetter, "The strength of weak ties," *American journal of sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

C. D. Green, "Classics in the history of psychology: An internet resource," *York University, Toronto, Ontario. ISSN*, pp. 1492–3713, 2000.

O. Green, R. McColl, and D. A. Bader, "A fast algorithm for streaming betweenness centrality," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. IEEE, 2012, pp. 11–20.

X. Gu, L. Akoglu, and A. Rinaldo, "Statistical analysis of nearest neighbor methods for anomaly detection," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 921–10 931.

R. Guimera and L. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, pp. 895–900, 2005.

C. Gunavathi, R. S. Priya, and S. Aarthy, "Big data analysis for anomaly detection in telecommunication using clustering techniques," in *Information Systems Design and Intelligent Applications*. Springer, 2019, pp. 111–121.

İ. Güneş, Ş. Gündüz-Öğüdücü, and Z. Çataltepe, "Link prediction using time series of neighborhood-based node similarity scores," *Data Mining and Knowledge Discovery*, vol. 30, no. 1, pp. 147–180, 2016.

M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.

P. Gupta, B. Srinivasan, V. Balasubramaniyan, and M. Ahamad, "Phoneypot: Data-driven understanding of telephony threats." in *NDSS*, 2015.

P. Gupta, R. Perdisci, and M. Ahamad, "Towards measuring the role of phone numbers in twitter-advertised spam," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS '18. New York, NY, USA: ACM, 2018, pp. 285–296. [Online]. Available: http://doi.acm.org/10.1145/3196494.3196516

S. Gupta, R. M. Anderson, and R. M. May, "Networks of sexual contacts: Implication for the pattern of spread," *Aids*, vol. 3, no. 12, 1989.

R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," *International Journal of Information Management*, vol. 45, pp. 289–307, 2019.

A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Laboratory (LANL), Tech. Rep., 2008.

A. Hajibagheri, G. Sukthankar, and K. Lakkaraju, "Leveraging network dynamics for improved link prediction," *arXiv preprint arXiv:1604.03221*, 2016.

R. A. Hanneman and M. Riddle, "Introduction to social network methods," 2005.

J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

R. Hassanzadeh, R. Nayak, and D. Stebila, "Analyzing the effectiveness of graph metrics for anomaly detection in online social networks," in *International Conference on Web Information Systems Engineering*. Springer, 2012, pp. 624–630.

M. He and S. Petoukhov, *Mathematics of Bioinformatics: Theory, Methods and Applications*. John Wiley & Sons, 2011, vol. 19.

D.-I. O. Hein, D.-W.-I. M. Schwind, and W. König, "Scale-free networks," *Wirtschaftsinformatik*, vol. 48, no. 4, pp. 267–275, 2006.

O. Heller, W. Mack, and J. Seitz, "Replikation der ebbinghaus' schen vergessenskurve mit der ersparnismethode: Das behalten und vergessen als funktion der zeit," *Zeitschrift für Psychologie mit Zeitschrift für angewandte Psychologie*, vol. 199, no. 1, pp. 3–18, 1991.

C. S. Hilas and J. N. Sahalos, "User profiling for fraud detection in telecommunication networks," in *5th International conference on technology and automation*, 2005, pp. 382–387.

C. S. Hilas, P. A. Mastorocostas, and I. T. Rekanos, "Clustering of telecommunications user profiles for fraud detection and security enhancement in large corporate networks: a case study," *Applied Mathematics & Information Sciences*, vol. 9, no. 4, p. 1709, 2015.

O. Hinz, B. Skiera, C. Barrot, and J. U. Becker, "Seeding strategies for viral marketing: An empirical comparison," *Journal of Marketing*, vol. 75, no. 6, pp. 55–71, 2011.

V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *Journal of the american Statistical association*, vol. 97, no. 460, pp. 1090–1098, 2002.

R. Hoffmann, P. v. R. Schleyer, and H. F. Schaefer, "Predicting molecules—-more realism, please!" *Angewandte Chemie International Edition*, vol. 47, no. 38, pp. 7164–7167, 2008.

J. Hollmén *et al.*, *User profiling and classification for fraud detection in mobile communications networks*. Helsinki University of Technology, 2000.

P. Holme, "Analyzing temporal networks in social media," *Proceedings of the IEEE*, vol. 102, no. 12, pp. 1922–1933, 2014.

P. Holme and J. Saramaki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.

Z. Huang, "Link Prediction Based on Graph Topology: The Predictive Value of Generalized Clustering Coefficient," *Workshop on Link Analysis: Dynamics and Static of Large Networks (LinkKDD2006)*, 2006.

Z. Huang and D. K. Lin, "The time-series link prediction problem with applications in communication surveillance," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 286–303, 2009.

B. A. Huberman and L. A. Adamic, "Evolutionary dynamics of the world wide web," *arXiv preprint cond-mat/9901071*, 1999.

B. A. Huberman and L. A. Adamic, "Internet: growth dynamics of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 131–131, 1999.

T. IDÉ and H. KASHIMA, "Eigenspace-based anomaly detection in computer systems," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04, 2004, pp. 440–449.

R. Ihaka and R. Gentleman, "R: a language for data analysis and graphics," *Journal of computational and graphical statistics*, vol. 5, no. 3, pp. 299–314, 1996.

T. Indow, *Retention curves of artificial and natural memory: Tight and soft models*. School of Social Sciences, University of California, 1993.

S. A. Iranmanesh, H. Sengar, and H. Wang, "A voice spam filter to clean subscribers' mailbox," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2012, pp. 349–367.

M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "Inferring strange behavior from connectivity pattern in social networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 126–138.

R. R. Junuthula, K. S. Xu, and V. K. Devabhaktuni, "Evaluating link prediction accuracy on dynamic networks with added and removed edges," *CoRR*, vol. abs/1607.07330, 2016.

K. Juszczyszyn, K. Musial, and M. Budka, "Link prediction based on subgraph evolution in dynamic social networks," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. New York, NY: IEEE, 2011, pp. 27–34.

G. Kaiafas, C. Hammerschmidt, R. State, C. D. Nguyen, T. Ries, and M. Ourdane, "An experimental analysis of fraud detection methods in enterprise telecommunication data using unsupervised outlier ensembles," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 37–42.

M. Kas, K. M. Carley, and L. R. Carley, "Incremental closeness centrality for dynamically changing social networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 1250–1258.

M. Kas, M. Wachs, K. M. Carley, and L. R. Carley, "Incremental algorithm for updating betweenness centrality in dynamically growing networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 33–40.

H. Kashima and N. Abe, "A parameterized probabilistic model of network evolution for supervised link prediction," in *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 2006, pp. 340–349.

L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

R. Kaur and S. Singh, "A survey of data mining and social network analysis based anomaly detection techniques," *Egyptian informatics journal*, vol. 17, no. 2, pp. 199–216, 2016.

T. R. Kepe, E. C. de Almeida, and T. Cerqueus, "Ksample: Dynamic sampling over unbounded data streams," *Journal of Information and Data Management*, vol. 6, no. 1, p. 32, 2015.

B. S. Khan and M. A. Niazi, "Network community detection: A review and visual survey," *arXiv preprint arXiv:1708.00977*, 2017.

H. Kim and R. Anderson, "Temporal node centrality in complex networks," *Phys. Rev. E*, vol. 85, p. 026107, Feb 2012.

J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

R. Klinkenberg, "Learning drifting concepts: Example selection vs. example weighting," *Intelligent Data Analysis*, vol. 8, no. 3, pp. 281–300, 2004.

D. Knoke and S. Yang, *Social network analysis*. Sage, 2008, vol. 154.

W. J. Koros and R. T. Chern, "Separation of gaseous mixtures using polymer membranes," *Handbook of separation process technology*, pp. 862–953, 1987.

G. Kossinets and D. J. Watts, "Empirical analysis of an evolving social network," *science*, vol. 311, no. 5757, pp. 88–90, 2006.

Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 2. IEEE, 2004, pp. 749–754.

N. Kourtellis, G. D. F. Morales, and F. Bonchi, "Scalable online betweenness centrality in evolving graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2494–2506, 2015.

D. Krackhardt, "Graph theoretical dimensions of informal organizations," in *Computational organization theory*. Psychology Press, 2014, pp. 107–130.

P. Krapivsky, G. Rodgers, and S. Redner, "Degree distributions of growing networks," *Physical Review Letters*, vol. 86, no. 23, p. 5401, 2001.

V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J.-H. Cui, and A. G. Percus, "Reducing large internet topologies for faster simulations," in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*. Springer, 2005, pp. 328–341.

J. Kunegis, "Konect: the koblenz network collection," in *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 2013, pp. 1343–1350.

M. Latapy, C. Magnien, and N. Del Vecchio, "Basic notions for the analysis of large two-mode networks," *Social networks*, vol. 30, no. 1, pp. 31–48, 2008.

K. Lei, Y. Liu, S. Zhong, Y. Liu, K. Xu, Y. Shen, and M. Yang, "Understanding user behavior in sina weibo online social network: a community approach," *IEEE Access*, vol. 6, pp. 13 302–13 316, 2018.

E. A. Leicht, P. Holme, and M. E. Newman, "Vertex similarity in networks," *Physical Review E*, vol. 73, no. 2, p. 026120, 2006.

D. Lentzen, G. Grutzek, H. Knospe, and C. Porschmann, "Content-based detection and prevention of spam over ip telephony-system design, prototype and first results," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.

J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 631–636.

J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 177–187.

J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, p. 5, 2007.

J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.

J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY: ACM, 2008, pp. 462–470.

J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

J. Leskovec, D. Huttenlocher, and J. M. Kleinberg, "Predicting Positive and Negative Links in Online Social Networks," *Conference on World Wide Web (WWW '10)*, pp. 641–650, 2010.

J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 631–640.

K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*. Australian Computer Society, Inc., 2005, pp. 333–342.

M. H. Levinson, "Linked: The new science of networks," *et Cetera*, vol. 61, no. 1, p. 170, 2004.

J. Li, A. Ritter, and D. Jurafsky, "Inferring user preferences by probabilistic logical reasoning over social networks," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1411.2679

K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu, "Improving one-class svm for anomaly detection," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)*, vol. 5. IEEE, 2003, pp. 3077–3081.

X. Li, X. Wu, S. Xu, S. Qing, and P.-C. Chang, "A novel complex network community detection approach using discrete particle swarm optimization with particle diversity and mutation," *Applied Soft Computing*, vol. 81, p. 105476, 2019.

Z. L. Li, X. Fang, and O. R. L. Sheng, "A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions," *CoRR*, vol. abs/1511.01868, 2015.

D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

D. Liben-Nowell and J. Kleinberg, "The Link Prediction Problem for Social Networks," *Proceedings of the Twelfth Annual ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 556–559, 2003.

R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 243–252.

Y. Lim, M. Jung, and U. Kang, "Memory-efficient and accurate sampling for counting local triangles in graph streams: from simple to multigraphs," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 1, p. 4, 2018.

D. Lin, "An information-theoretic definition of similarity," in *In Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 296–304.

T. Z. Linyuan Lü, "Link Prediction in Complex Networks: A Survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

C. Liu, J. Wang, H. Zhang, and M. Yin, "Mapping the hierarchical structure of the global shipping network by weighted ego network analysis," *International Journal of Shipping and Transport Logistics*, vol. 10, no. 1, pp. 63–86, 2018.

S. Liu, L. Li, C. Faloutsos, and L. M. Ni, "Mobile phone graph evolution: Findings, model and interpretation," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 323–330.

G. R. Loftus, "Evaluating forgetting curves." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 11, no. 2, p. 397, 1985.

L. Lü and T. Zhou, "Link prediction in weighted networks: The role of weak ties," *EPL (Europhysics Letters)*, vol. 89, no. 1, p. 18001, 2010.

L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Physical Review E*, vol. 80, no. 4, p. 046122, 2009.

H. H. Ma, S. Gustafson, A. Moitra, and D. Bracewell, "Ego-centric network sampling in viral marketing applications," in *Mining and Analyzing Social Networks*. Springer, 2010, pp. 35–51.

P. C. Mahalanobis, "On the generalized distance in statistics," National Institute of Science of India, 1936.

J.-P. Malrieu, "Quantum chemistry and its unachieved missions," *Journal of Molecular Structure: THEOCHEM*, vol. 424, no. 1, pp. 83–91, 1998.

M. Marjan, N. Zaki, and E. A. Mohamed, "Link prediction in dynamic social networks: A literature review," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*. IEEE, 2018, pp. 200–207.

P. Matuszyk, J. Vinagre, M. Spiliopoulou, A. M. Jorge, and J. Gama, "Forgetting techniques for stream-based matrix factorization in recommender systems," *Knowledge and Information Systems*, vol. 55, no. 2, pp. 275–304, 2018.

J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 4, 2014.

M. McGlohon, L. Akoglu, and C. Faloutsos, "Statistical properties of social networks," in *Social network data analytics*. Springer, 2011, pp. 17–42.

A. Metwally, D. Agrawal, and A. El Abbadi, "Efficient computation of frequent and top-k elements in data streams," in *Database Theory-ICDT 2005*. Springer, 2005, pp. 398–412.

S. Milgram, "The small world problem," *Psychology Today*, vol. 1, pp. 61–67, 1967.

S. Milgram, "The small world problem," *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.

C. Miranda, M. Arora, K. Thirumalaiappan, and B. Dowd, "System and method for real time fraud analysis of communications data," Apr. 25 2019, uS Patent App. 16/090,720.

A. Mohammed Aamir, A. Muhammad AJmal, C. Mario Parreno, H. Feng, and M. Aad Van, "Consumer-facing technology fraud: Economics, attack methods and potential solutions," *Future Generation Computer Systems*, 2019.

B. Moradabadi and M. R. Meybodi, "Link prediction in weighted social networks using learning automata," *Engineering Applications of Artificial Intelligence*, vol. 70, pp. 16–24, 2018.

J. L. Moreno, *Who shall survive*. JSTOR, 1934, vol. 58.

H. Mouss, D. Mouss, N. Mouss, and L. Sefouhi, "Test of page-hinckley, an approach for fault detection in an agro-alimentary production system," in *Control Conference, 2004. 5th Asian*, vol. 2. IEEE, 2004, pp. 815–818.

C.-H. Mu, J. Xie, Y. Liu, F. Chen, Y. Liu, and L.-C. Jiao, "Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks," *Applied Soft Computing*, vol. 34, pp. 485–501, 2015.

G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *GI/ITG Workshop MMBnet*, 2007, pp. 13–14.

T. Murata and S. Moriyasu, "Link prediction based on structural properties of online social networks," *New Generation Computing*, vol. 26, no. 3, pp. 245–257, 2008.

B. B. Murdock Jr, "The retention of individual items." *Journal of experimental psychology*, vol. 62, no. 6, p. 618, 1961.

J. M. Murre and J. Dros, "Replication and analysis of ebbinghaus' forgetting curve," *PloS one*, vol. 10, no. 7, p. e0120644, 2015.

J. M. Murre, A. G. Chessa, and M. Meeter, "A mathematical model of forgetting and amnesia," *Frontiers in psychology*, vol. 4, p. 76, 2013.

I. Murynets, M. Zabarankin, R. P. Jover, and A. Panagia, "Analysis and detection of simbox fraud in mobility networks," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1519–1526.

H. Nasiri, S. Nasehi, and M. Goudarzi, "Evaluation of distributed stream processing frameworks for iot applications in smart cities," *Journal of Big Data*, vol. 6, no. 1, p. 52, 2019.

T. O. Nelson, "Savings and forgetting from long-term memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 10, no. 5, pp. 568–576, 1971.

M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, no. 2, p. 025102, 2001.

M. E. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.

M. E. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, p. 026126, 2003.

M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

M. E. Newman, "Detecting community structure in networks," *The European Physical Journal B*, vol. 38, no. 2, pp. 321–330, 2004.

M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora, *Temporal Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. Graph Metrics for Temporal Networks, pp. 15–40.

Z. Niu, S. Shi, J. Sun, and X. He, "A survey of outlier detection methodologies and their applications," in *International Conference on Artificial Intelligence and Computational Intelligence*. Springer, 2011, pp. 380–387.

K. Noto, C. Brodley, and D. Slonim, "Frac: a feature-modeling approach for semi-supervised and unsupervised anomaly detection," *Data mining and knowledge discovery*, vol. 25, no. 1, pp. 109–133, 2012.

M. Oliveira and J. Gama, "An overview of social network analysis," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 99–115, 2012.

M. Oliveira, L. Torgo, and V. S. Costa, "Evaluation procedures for forecasting with spatio-temporal data," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 703–718.

J. O'Madadhain, J. Hutchins, and P. Smyth, "Prediction and ranking algorithms for event-based network data," *ACM SIGKDD explorations newsletter*, vol. 7, no. 2, pp. 23–30, 2005.

J. O'Madadhain, J. Hutchins, and P. Smyth, "Prediction and ranking algorithms for event-based network data," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 23–30, 2005.

T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social networks*, vol. 32, no. 3, pp. 245–251, 2010.

D. Ortiz-Arroyo, *Discovering sets of key players in social networks*. Springer, 2010.

M. Osborne, A. Lall, and B. Van Durme, "Exponential reservoir sampling for streaming language models." in *ACL (2)*, 2014, pp. 687–692.

E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *arXiv preprint physics/0506133*, 2005.

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

S. Pan, J. Wu, X. Zhu, and C. Zhang, "Graph ensemble boosting for imbalanced noisy graph stream classification," *IEEE transactions on cybernetics*, vol. 45, no. 5, pp. 954–968, 2014.

P. Panzarasa, T. Opsahl, and K. M. Carley, "Patterns and dynamics of users' behavior and interaction: Network analysis of an online community," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 5, pp. 911–932, 2009.

A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "Fast and accurate link prediction in social networking systems," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2119–2132, 2012.

M. Papagelis, G. Das, and N. Koudas, "Sampling online social networks," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 3, pp. 662–676, 2013.

S. Parthasarathy, Y. Ruan, and V. Satuluri, "Community discovery in social networks: Applications, methods and emerging trends," in *Social network data analytics*. Springer, 2011, pp. 79–113.

J. P. Perdew, A. Ruzsinszky, L. A. Constantin, J. Sun, and G. I. Csonka, "Some fundamental issues in ground-state density functional theory: a guide for the perplexed," *Journal of Chemical Theory and Computation*, vol. 5, no. 4, pp. 902–908, 2009.

F. S. F. Pereira, S. Amo, and J. Gama, "Evolving centralities in temporal graphs: a twitter network analysis," in *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, 2016.

F. S. F. Pereira, S. de Amo, and J. Gama, *On Using Temporal Networks to Analyze User Preferences Dynamics*. Cham: Springer International Publishing, 2016, pp. 408–423.

F. S. F. Pereira, S. de Amo, and J. Gama, "Detecting events in evolving social networks through node centrality analysis," *Workshop on Large-scale Learning from Data Streams in Evolving Environments co-located with ECML/PKDD*, 2016.

F. S. F. Pereira, S. Tabassum, S. Amo, and J. Gama, "Processing evolving social networks for change detection based on centrality measures," in *Learning from Data Streams in Evolving Environments*. Cham: Springer, 2018.

F. S. Pereira, S. de Amo, and J. Gama, "Evolving centralities in temporal graphs: a twitter network analysis," in *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, vol. 2. IEEE, 2016, pp. 43–48.

F. S. Pereira, S. de Amo, and J. Gama, "On using temporal networks to analyze user preferences dynamics," in *International Conference on Discovery Science*. Springer, 2016, pp. 408–423.

F. S. Pereira, S. Tabassum, J. Gama, S. de Amo, and G. M. Oliveira, "Processing evolving social networks for change detection based on centrality measures," in *Learning from data streams in evolving environments*. Springer, 2019, pp. 155–176.

L. Peterson and M. J. Peterson, "Short-term retention of individual verbal items." *Journal of experimental psychology*, vol. 58, no. 3, p. 193, 1959.

D. Q. Phung, S. Venkatesh *et al.*, "Preference networks: Probabilistic models for recommendation systems," in *Proceedings of the sixth Australasian conference on Data mining and analytics-Volume 70*. Australian Computer Society, Inc., 2007, pp. 195–202.

P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *ISCIS*, vol. 3733, 2005, pp. 284–293.

A. Popescul and L. H. Ungar, "Statistical relational learning for link prediction," in *IJCAI workshop on learning statistical models from relational data*, vol. 2003. Citeseer, 2003.

M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks," *Notices of the AMS*, vol. 56, no. 9, pp. 1082–1097, 2009.

L. Postman and D. A. Riley, "Degree of learning and interserial interference in retention: A review of the literature and an experimental analysis." *University of California Publications in Psychology*, 1959.

A. Potgieter, K. A. April, R. J. Cooke, and I. O. Osunmakinde, "Temporality in link prediction: Understanding social complexity," *Emergence: Complexity and Organization*, vol. 11, no. 1, p. 69, 2009.

D. Price, "Networks of scientific papers," *Science*, vol. 149, pp. 510–515, 1965.

D. d. S. Price, "A general theory of bibliometric and other cumulative advantage processes," *Journal of the Association for Information Science and Technology*, vol. 27, no. 5, pp. 292–306, 1976.

S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: a survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.

A. Rapoport, "Spread of information through a population with socio-structural bias i: Assumption of transitivity," *Bulletin of Mathematical Biophysics*, vol. 15, pp. 523–533, 1953.

E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

R. Raymond and H. Kashima, "Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs," in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2010, pp. 131–147.

U. Rebbapragada, P. Protopapas, C. E. Brodley, and C. Alcock, "Finding anomalous periodic time series," *Machine learning*, vol. 74, no. 3, pp. 281–313, 2009.

W. J. Reed and M. Jorgensen, "The double pareto-lognormal distribution—a new parametric model for size distributions," *Communications in Statistics-Theory and Methods*, vol. 33, no. 8, pp. 1733–1753, 2004.

A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith, "Clustering rules: a comparison of partitioning and hierarchical clustering algorithms," *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 4, pp. 475–504, 2006.

S. A. Rice, "The identification of blocs in small political bodies," *American Political Science Review*, vol. 21, no. 3, pp. 619–627, 1927.

M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY: ACM, 2002, pp. 61–70.

T. Ritter, "The networking company: antecedents for coping with relationships and networks effectively," *Industrial marketing management*, vol. 28, no. 5, pp. 467–479, 1999.

G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 35, 2018.

G. Rossetti, M. Berlingerio, and F. Giannotti, "Scalable link prediction on multidimensional networks," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 979–986.

P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event detection in activity networks," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14, 2014, pp. 1176–1185.

D. C. Rubin, "On the retention function for autobiographical memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 21, no. 1, pp. 21–38, 1982.

D. C. Rubin and A. E. Wenzel, "One hundred years of forgetting: A quantitative description of retention." *Psychological review*, vol. 103, no. 4, p. 734, 1996.

D. C. Rubin, S. Hinton, and A. Wenzel, "The precise time course of retention." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 25, no. 5, p. 1161, 1999.

G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

A. E. Sariyuce, K. Kaya, E. Saule, and U. V. Catalyurek, "Incremental algorithms for closeness centrality," in *Big Data, 2013 IEEE International Conference on*. IEEE, 2013, pp. 487–492.

P. Sarkar and A. W. Moore, "Dynamic social network analysis using latent space models," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 31–40, 2005.

P. Sarkar, D. Chakrabarti, and M. Jordan, "Nonparametric link prediction in dynamic networks," *arXiv preprint arXiv:1206.6394*, 2012.

A. D. Sarma, S. Gollapudi, and R. Panigrahy, "Estimating pagerank on graph streams," *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 13, 2011.

R. Sarmento, M. Oliveira, M. Cordeiro, J. Gama, and S. Tabassum, "Social network analysis of streaming call graphs," in *Big Data Analysis: New Algorithms for a New Society*. Springer, 2015, p. In Press.

R. Sarmento, M. Oliveira, M. Cordeiro, S. Tabassum, and J. Gama, "Social network analysis in streaming call graphs," in *Big Data Analysis: New Algorithms for a New Society*. Springer, 2016, pp. 239–261.

D. Savage, X. Zhang, X. Yu, P. Chou, and Q. Wang, "Anomaly detection in online social networks," *Social Networks*, vol. 39, pp. 62–70, 2014.

T. Schank and D. Wagner, *Approximating clustering-coefficient and transitivity*. Universität Karlsruhe, Fakultät für Informatik, 2004.

J. A. Schinka, W. F. Velicer, and I. B. Weiner, *Handbook of psychology: Research methods in psychology, Vol. 2*. John Wiley & Sons Inc, 2013.

E. Schubert and P. J. Rousseeuw, "Faster k-medoids clustering: improving the pam, clara, and clarans algorithms," in *International Conference on Similarity Search and Applications*. Springer, 2019, pp. 171–187.

P. Schuetz and A. Caflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement," *Physical Review E*, vol. 77, no. 4, p. 046112, 2008.

J. Scott, "Social network analysis," *Sociology*, vol. 22, no. 1, pp. 109–127, 1988.

J. Scott, *Social network analysis*. Sage, 2012.

R. Sebastião, M. M. Silva, R. Rabiço, J. Gama, and T. Mendonça, "Real-time algorithm for changes detection in depth of anesthesia signals," *Evolving Systems*, vol. 4, no. 1, pp. 3–12, 2013.

M. Seshadri, S. Machiraju, A. Sridharan, J. Bolot, C. Faloutsos, and J. Leskove, "Mobile call graphs: beyond power-law and lognormal distributions," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 596–604.

S. Shaik, "Is my chemical universe localized or delocalized? is there a future for chemical concepts?" *New Journal of Chemistry*, vol. 31, no. 12, pp. 2015–2028, 2007.

R. Shang, J. Bai, L. Jiao, and C. Jin, "Community detection based on modularity and an improved genetic algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 5, pp. 1215–1231, 2013.

H.-W. Shen and X.-Q. Cheng, "Spectral methods for the detection of network community structure: a comparative analysis," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 10, p. P10020, 2010.

J. Shetty and J. Adibi, "The enron email dataset database schema and brief statistical report," *Information sciences institute technical report, University of Southern California*, vol. 4, no. 1, pp. 120–128, 2004.

J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, p. 107, 2000.

H. A. Simon, "A note on jost's law and exponential forgetting," *Psychometrika*, vol. 31, no. 4, pp. 505–506, 1966.

N. J. Slamecka and B. McElree, "Normal forgetting of verbal lists as a function of their degree of learning." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 9, no. 3, p. 384, 1983.

M. A. Smith, B. Shneiderman, N. Milic-Frayling, E. Mendes Rodrigues, V. Barash, C. Dunne, T. Capone, A. Perer, and E. Gleave, "Analyzing (social media) networks with nodexl," in *Proceedings of the fourth international conference on Communities and technologies*. ACM, 2009, pp. 255–264.

P. R. Soares and R. B. Prudêncio, "Proximity measures for link prediction based on temporal events," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6652–6660, 2013.

H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 2009, pp. 322–335.

T. Sørensen, *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*, ser. Biologiske Skrifter // Det Kongelige Danske Videnskabernes Selskab. I kommission hos E. Munksgaard, 1948.

S. Spiegel, J. Clausen, S. Albayrak, and J. Kunegis, "Link prediction on evolving data using tensor factorization," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Berlin - Heidelberg: Springer, 2011, pp. 100–110.

M. R. Srilatha P, "Similarity Index based Link Prediction Algorithms in Social Networks: A Survey," *Journal of Telecommunications and Information Technology*, vol. 2, pp. 87–94, 2016.

M. P. Stumpf, C. Wiuf, and R. M. May, "Subnets of scale-free networks are not scale-free: sampling properties of networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 12, pp. 4221–4224, 2005.

H. Sun, J. Huang, X. Zhang, J. Liu, D. Wang, H. Liu, J. Zou, and Q. Song, "Incorder: Incremental density-based community detection in dynamic networks," *Knowledge-Based Systems*, vol. 72, pp. 1–12, 2014.

J. Sun and Y. Zhu, "Microblogging personalized recommendation based on ego networks," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, vol. 1. New York, NY: IEEE, 2013, pp. 165–170.

Y. Sun, J. Tang, J. Han, M. Gupta, and B. Zhao, "Community evolution detection in dynamic heterogeneous information networks," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*. ACM, 2010, pp. 137–146.

I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *International conference on networked digital technologies*. Springer, 2012, pp. 135–145.

S. Tabassum, "Social network analysis of mobile streaming networks," in *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, vol. 2. IEEE, 2016, pp. 20–25.

S. Tabassum and J. Gama, "Sampling ego-networks with forgetting factor," in *IEEE Workshop on High Velocity Mobile Data Mining*, 2016, p. In Press.

S. Tabassum and J. Gama, "Evolution analysis of call ego-networks," in *International Conference on Discovery Science*. Springer, 2016, pp. 213–225.

S. Tabassum and J. Gama, "Sampling evolving ego-networks with forgetting factor," 2016.

S. Tabassum and J. Gama, "Social network analysis of mobile streaming networks," in *IEEE Conference on Mobile Data Mining, PhD Forum*, 2016, p. In Press.

S. Tabassum and J. Gama, "Sampling massive streaming call graphs," in *ACM Symposium on Advanced Computing*, 2016, pp. 923–928.

S. Tabassum and J. Gama, "Sampling evolving ego-networks with forgetting factor," in *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, vol. 2. IEEE, 2016, pp. 55–59.

S. Tabassum and J. Gama, "Biased dynamic sampling for temporal network streams," in *International Conference on Complex Networks and their Applications*. Springer, 2018, pp. 512–523.

S. Tabassum and J. Gama, "Sampling massive streaming call graphs," in *Proceedings of the 2016 ACM Symposium on Applied Computing*, ser. SAC '16. New York, NY: ACM, 2016, pp. 923–928.

S. Tabassum, A. A. Muhammad, and J. Gama, "Profiling high leverage points for fraud detection in telecommunication networks," *in journal submission*.

S. Tabassum, F. S. Pereira, S. Fernandes, and J. Gama, "Social network analysis: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 5, p. e1256, 2018.

M. Takaffoli, R. Rabbany, and O. R. Zaïane, "Incremental local community identification in dynamic social networks," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013, pp. 90–94.

J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 807–816.

D. M. J. Tax, "One-class classification: Concept learning in the absence of counter-examples." 2002.

M. Thelwall, "Interpreting social science link analysis research: A theoretical framework," *Journal of the American Society for Information Science and Technology*, vol. 57, pp. 60–68, 2006.

H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," 2006.

T. Tylenda, R. Angelova, and S. Bedathur, "Towards time-aware link prediction in evolving social networks," in *Proceedings of the 3rd workshop on social network mining and analysis*. ACM, 2009, p. 9.

B. V and M. A., "Pajek - program for large network analysis," *Connections*, vol. 21, pp. 47–57, 1998.

G. Valkanas and D. Gunopulos, "Event detection from social media data." *IEEE Data Eng. Bull.*, vol. 36, no. 3, pp. 51–58, 2013.

G. G. Van de Bunt, M. A. Van Duijn, and T. A. Snijders, "Friendship networks through time: An actor-oriented dynamic statistical network model," *Computational & Mathematical Organization Theory*, vol. 5, no. 2, pp. 167–192, 1999.

K. Varmuza and P. Filzmoser, *Introduction to multivariate statistical analysis in chemometrics*. CRC press, 2016.

R. Vilalta and S. Ma, "Predicting rare events in temporal domains," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE, 2002, pp. 474–481.

B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM workshop on Online social networks*. ACM, 2009, pp. 37–42.

B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2Nd ACM Workshop on Online Social Networks*, ser. WOSN '09. ACM, 2009, pp. 37–42.

B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, 1985.

K. Wakita and T. Tsurumi, "Finding community structure in mega-scale social networks," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 1275–1276.

C. Wang, V. Satuluri, and S. Parthasarathy, "Local probabilistic models for link prediction," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. New York, NY: IEEE, 2007, pp. 322–331.

D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. Acm, 2011, pp. 1100–1108.

H. Wang, X. Shi, Y. Li, H. Chang, W. Chen, J. Tang, and E. Martins, "User profile management for personalized telecom service," in *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*. New York, NY: IEEE, 2008, pp. 1087–1092.

P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, 2015.

X. Wang, M. Wang, and J. Han, "Accds: A criminal community detection system based on evolving social graphs," in *International Conference on Conceptual Modeling*. Springer, 2018, pp. 44–48.

S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

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

C.-P. Wei and I.-T. Chiu, "Turning telecommunications call details to churn prediction: a data mining approach," *Expert systems with applications*, vol. 23, no. 2, pp. 103–112, 2002.

W. Wei and K. M. Carley, "Measuring temporal patterns in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, p. 9, 2015.

W. Wei, J. Erenrich, and B. Selman, "Towards efficient sampling: Exploiting random walk strategies," in *AAAI*, vol. 4, 2004, pp. 670–676.

B. Wellman, "Are personal communities local? a dumptarian reconsideration," *Social networks*, vol. 18, no. 4, pp. 347–354, 1996.

J. T. Wixted, "Analyzing the empirical course of forgetting." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 16, no. 5, p. 927, 1990.

R. S. Woodworth and H. Schlosberg, *Experimental psychology*. Oxford and IBH Publishing, 1955.

G. Wu, Z. Zhao, G. Fu, H. Wang, Y. Wang, Z. Wang, J. Hou, and L. Huang, "A fast knn-based approach for time sensitive anomaly detection over data streams," in *International Conference on Computational Science*. Springer, 2019, pp. 59–74.

H. Wu, J. Cheng, S. Huang, Y. Ke, Y. Lu, and Y. Xu, "Path problems in temporal graphs," *Proceedings of the VLDB Endowment*, vol. 7, no. 9, pp. 721–732, 2014.

D. Wulsin, J. Blanco, R. Mani, and B. Litt, "Semi-supervised anomaly detection for eeg waveforms using deep belief nets," in *2010 Ninth International Conference on Machine Learning and Applications*. IEEE, 2010, pp. 436–441.

E. W. Xiang, "A Survey on Link Prediction Models for Social Network Data," *Science And Technology*, 2008.

J. Xu and H. Chen, "Criminal network analysis and visualization," *Communications of the ACM*, vol. 48, no. 6, pp. 100–107, 2005.

Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, "Evaluating link prediction methods," *Knowledge and Information Systems*, vol. 45, no. 3, pp. 751–782, 2015.

Y. Yang, Y. Xu, Y. Sun, Y. Dong, F. Wu, and Y. T. Zhuang, "Mining fraudsters and fraudulent strategies in large-scale mobile social networks," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

X. Ying, X. Wu, and D. Barbará, "Spectrum based fraud detection in social networks," in *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 2011, pp. 912–923.

P. Yong, L. Xiaodong, and J. He, "Link prediction in heterogeneous networks based on tensor facto-rization," *Open Cybernetics & Systemics Journal*, vol. 8, pp. 316–321, 2014.

M. N. York, "Estimated 24.9m americans lost $8.9b in phone scams," 01, December 2018. [Online]. Available: http://tiny.cc/ia98bz

R. Zafarani, M. A. Abbasi, and H. Liu, *Social media mining: an introduction*. New York, NY: Cambridge University Press, 2014.

C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1409–1416.

J. Zhang, K. Zhu, Y. Pei, G. Fletcher, and M. Pechenizkiy, "Clustering-structure representative sampling from graph streams," in *International Workshop on Complex Networks and their Applications*. Springer, 2017, pp. 265–277.

P. Zhao, C. Aggarwal, and G. He, "Link prediction in graph streams," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 553–564.

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

L. Zhu, D. Guo, J. Yin, G. Ver Steeg, and A. Galstyan, "Scalable temporal latent space inference for link prediction in dynamic social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2765–2777, 2016.

Appendix A

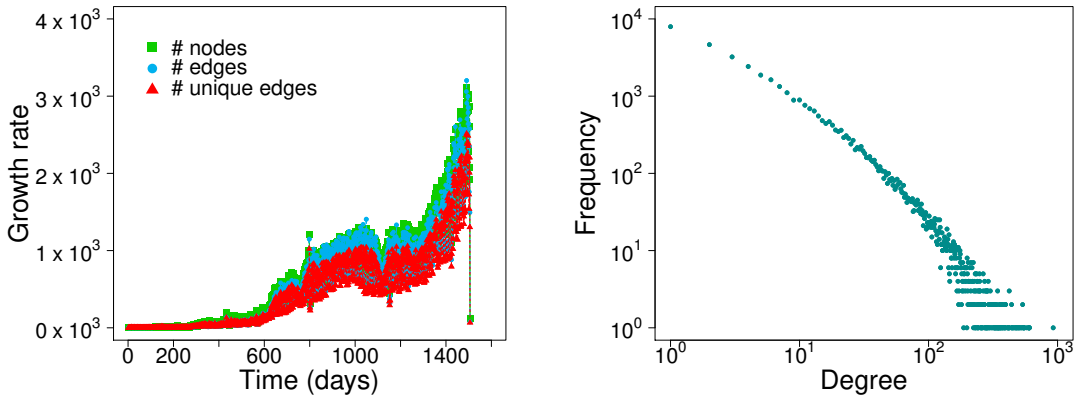
Summary of Datasets and Source Codes

For carrying out research in the area of evolving graph streams we considered datasets from various domains. The context specific characteristics of different domains helped us generalize our methods and validate their efficacy. While some chapters (4 and 8) are specific to an application in a single domain, which has its own importance as generalised methods do not always fit in specific scenarios.

It was a laborious task to find publicly available big network datasets associated with temporal information and recurring links which is the main focus of this work. We got the datasets which are available at the KONECT (Kunegis, 2013) and snap repositories. The details are given below along with the representation of data in the form of a network and its semantics. Besides the public datasets, we also obtained the Call Detail Record logs (CDR) from a service provider. CDR is the largest data set we have and its description is also provided below. As all the networks mentioned below contain recurring links, we show their growth rate in terms of total number of edges and also unique edges in each dataset to exhibit scale. The nodes and their degrees in these networks are not normally distributed which increases the complexity unlike the applications of most of the statistical techniques.

- **Facebook Wall Posts:** This dataset is a small subset of users posts on other users wall on Facebook (FB)[1] networks. These wall posts are connected by two users (nodes), the one who posts and the other whose wall it is posted on. The dataset contains 876,993 directed edges (posts) and 46,952 nodes with a reciprocity of 62.5% and average degree 37.357 edges/vertex. The detailed description of data is also available at Viswanath et al. (2009a). The data spans from 2004 to 2009. We accessed the data as a stream of wall posts (edges) associated with time stamps

[1] Data available at http://konect.uni-koblenz.de/networks/facebook-wosn-wall

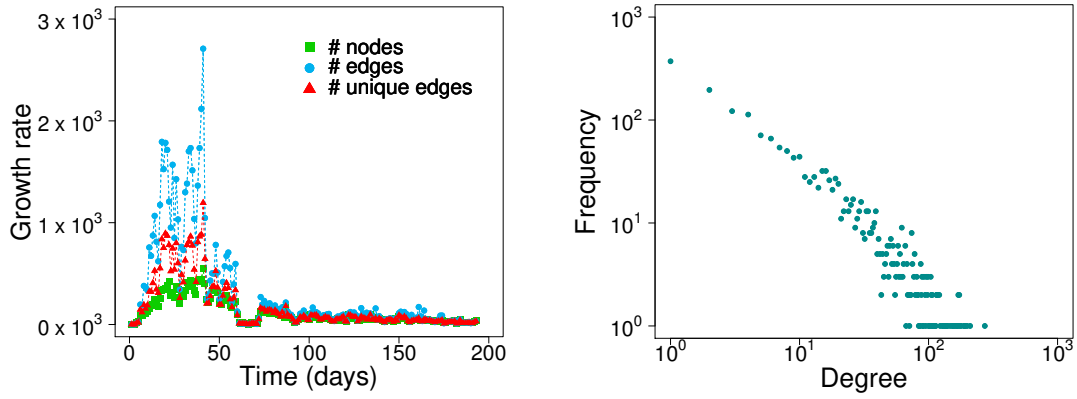


(a) Temporal Distribution (b) Degree Distribution

Figure A.1: Facebook wall post network

in order which is detailed in chapter 5. The temporal and degree distributions are given in Figure A.1

- **CollegeMsg:** This dataset is a network of users from an online social network at the University of California, Irvine[2]. Users could look for the profiles of other users on the network and then start conversations. An edge (u, v, t) means that user u sent a private message to user v at time t. There are 59835 edges and 1899 nodes in this dataset. The growth rate over time and degree distribution is displayed in figure A.2.



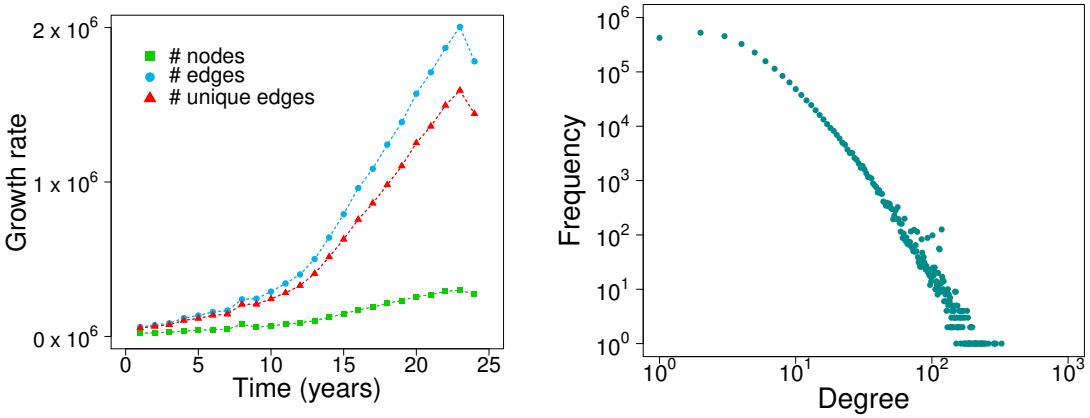
(a) Temporal Distribution (b) Degree Distribution

Figure A.2: CollegeMsg network

- **DBLP:** This is a collaboration network of authors from DBLP computer science bibliography gained from KONECT[2] networks as well. The nodes represent authors

[2]Data available at https://snap.stanford.edu/data/CollegeMsg.html
[2]Data available at http://konect.uni-koblenz.de/networks/dblp_coauthor

and edges are collaborations between them over time. From the available dataset we considered the last 24 years of data as we cleaned the missing data of just one edge for a year. The network consist of 18,986,618 edges and 1,314,050 nodes with an average degree of 28.898 edges/vertex. It is also a multi-graph as two authors can collaborate for more than one paper. The data ranges from 1990 to 2013 and the edges are associated with the time of publication given in years. The growth rate and degree distribution of nodes given in figure A.3



(a) Temporal Distribution

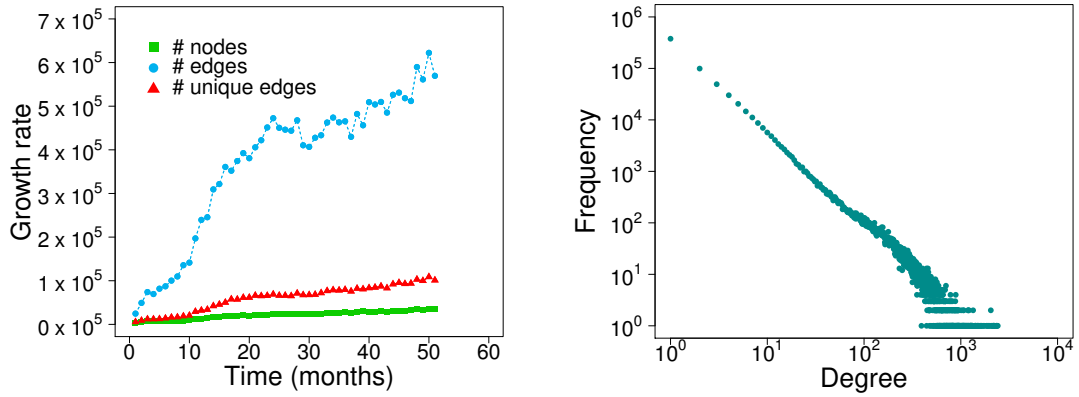
(b) Degree Distribution

Figure A.3: DBLP co-authorship network

- **Last.fm band:** This is a bipartite network of users and the bands they listened from a music website Last.fm. available at KONECT[2]. The users can listen to same bands again creating recurring links over time. We considered a total number of 176061 nodes and 18956166 edges for 51 months. The analysis given in figure A.4

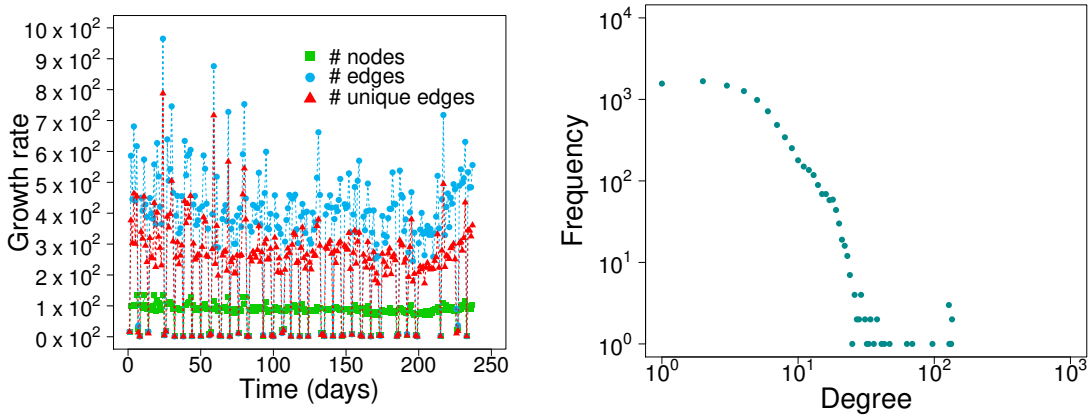
- **Radoslaw Email:** This is the internal email communication network between employees of a mid-sized manufacturing company from KONECT[2]. The network is directed and nodes represent employees. The left node represents the sender and the right node represents the recipient. Edges between two nodes are individual emails associated with time. Number of nodes is 167 and edges 82927 on aggregated scale for 237 days in 2010 (figure A.5. Reciprocity of nodes is 87.6%.

[2]Data available at http://http://konect.uni-koblenz.de/networks/lastfm_band
[2]Data available at http://konect.uni-koblenz.de/networks/radoslaw_email



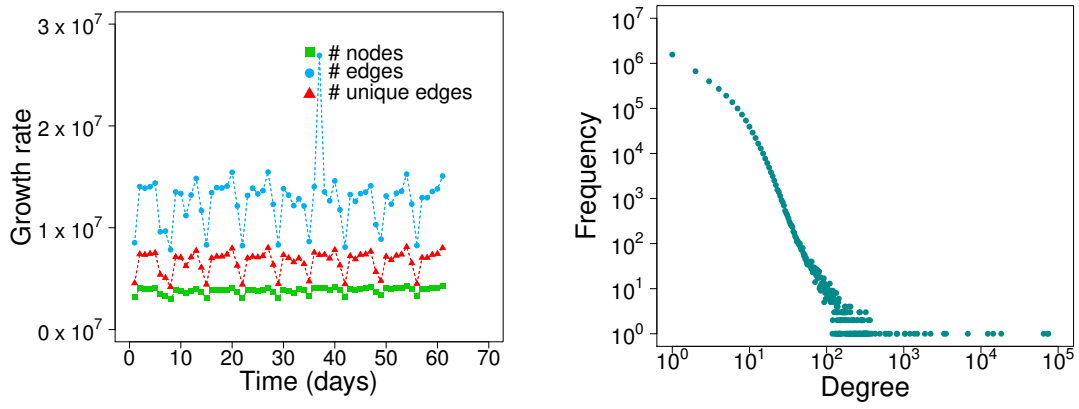
(a) Temporal Distribution (b) Degree Distribution

Figure A.4: Last.fm band network



(a) Temporal Distribution (b) Degree Distribution

Figure A.5: Radoslaw email network



(a) Temporal Distribution (b) Degree Distribution

Figure A.6: Telecommunications call network

- **Call Network:** CDR's (Call Detail Records) from telecommunications are one of the largest and fastest data streams with stupendous mass of information hidden. We have a call network with 800 million calls (edges) on an aggregated scale made by 15 million subscribers (nodes). The network is generated from anonymised CDR's from a service provider (WeDo Technologies). The average speed of data is 10 to 280 calls per second around mid-night and mid-day. Streaming approach is highly feasible for this kind of rapidly evolving data. Nodes correspond to callers and callees. The edges between them represent calls. The edges are bidirectional correspond to incoming and outgoing calls. Calls in the data are associated with timestamps when the call was initiated and also duration, direction and type of call. These attributes can be used for weighting edges. As it is a network with recurring links/calls, the weights on edges can also be mapped to the frequency of a call. We process this network as a stream of calls in the order of time in part II and III of this thesis. The figure A.6 illustrates the activity in the network per day for two months. One can observe the spike in the number of edges but not unique edges which indicates an event on that day. The figure also presents the degree distribution which explains that more number of users make less calls while few users making larger proportion of calls.

The source codes for the algorithms implemented in Chapters 4 to 7 is provided in GitHub (https://github.com/ShaziaTabassum/Evolving-Multigraph-Stream/)

Appendix B

Bibliographical Contributions

Main Publications

Tabassum, S., Veloso, B., & Gama, J. (2020). On fast and scalable recurring link's prediction in evolving multi-graph streams. Network Science. Cambridge University Press.

Tabassum, S., Pereira, F. S., Fernandes, S., & Gama, J. (2018). Social network analysis: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(5), e1256.

Tabassum, S., & Gama, J. (2018, December). Biased dynamic sampling for temporal network streams. In International Conference on Complex Networks and their Applications (pp. 512-523). Springer, Cham.

Tabassum, S., Pereira, F. S., & Gama, J. (2018). Knowledge Discovery from Temporal Social Networks. Intelligent Informatics, 10.

Tabassum, S., & Gama, J. (2016, October). Evolution analysis of call ego-networks. In International conference on discovery science (pp. 213-225). Springer, Cham.

Tabassum, S. (2016, June). Social network analysis of mobile streaming networks. In 2016 Workshop on High Velocity Mobile Data Mining in 17th IEEE International Conference on Mobile Data Management (MDM) (Vol. 2, pp. 20-25). IEEE.

Tabassum, S., & Gama, J. (2016, June). Sampling evolving ego-networks with forgetting factor. In 2016 Workshop on High Velocity Mobile Data Mining in 17th IEEE

international conference on mobile data management (MDM) (Vol. 2, pp. 55-59). IEEE.

Tabassum, S., & Gama, J. (2016, April). Sampling massive streaming call graphs. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 923-928). ACM.

Tabassum, S., Pereira, F. S., Fernandes, S., & Gama, J. (2018). Cover Image, Volume 8, Issue 5. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(5), e1281.

S., Tabassum, Ajmal, M., & Gama, J. Profiling High Leverage Users for Fraud Detection in Telecom Data Networks. Manuscript submitted for a journal review.

Coauthored Publications

Pereira, F. S., Tabassum, S., Gama, J., de Amo, S., & Oliveira, G. M. (2019). Processing evolving social networks for change detection based on centrality measures. In Learning from data streams in evolving environments (pp. 155-176). Springer, Cham.

Ajmal, M., Bag, S., Tabassum, S., & Hao, F. (2017). privy: Privacy preserving collaboration across multiple service providers to combat telecoms spam. IEEE transactions on emerging topics in computing.

Sarmento, R., Oliveira, M., Cordeiro, M., Tabassum, S., & Gama, J. (2016). Social network analysis in streaming call graphs. In Big Data Analysis: New Algorithms for a New Society (pp. 239-261). Springer, Cham.

Veloso, B., Tabassum, S., Carlos, M., Raphael, E., Raul, A., & Gama, J. Interconnect Bypass Fraud Detection: a Case Study. Manuscript submitted for a journal review.

Tutorials and events participation

Tutorial: Sampling Evolving Networked Data @ 7IMM (2018) 7th Iberian Mathematical Meeting.

Tutorial: Knowledge Discovery from Temporal Social Networks @ SDM (2018) SIAM International Conference on data Mining.

Workshop Chair: Evolving Networks @ DSAA (2017) The 4th IEEE International Conference on Data Science and Advanced Analytics.

Organizing Committee: Discovery Challenge @ EPIA (2017) European Conference on Artificial Intelligence.

Organizing Committee: MobDM (Workshop on high velocity mobile data mining) @ 17TH IEEE International Conference on Mobile Data Management (MDM'16)