TEMPORAL NETWORKS:

STRUCTURE, FUNCTION AND APPLICATIONS

Sandipan Sikdar

TEMPORAL NETWORKS:

STRUCTURE, FUNCTION AND APPLICATIONS

Thesis submitted to the
Indian Institute of Technology, Kharagpur
for award of the degree

of

Doctor of Philosophy

by

Sandipan Sikdar

Under the supervision of

Dr. Animesh Mukherjee
and
Prof. Niloy Ganguly

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

July 2017

CERTIFICATE

This is to certify that the thesis entitled **"Temporal networks: Structure, function and applications"**, *submitted by* **Sandipan Sikdar** *to the Indian Institute of Technology, Kharagpur, for the award of the degree of Doctor of Philosophy, is a record of bona fide research work carried out by him under my supervision and guidance. The thesis, in my opinion, is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute. To the best of my knowledge, the results embodied in this thesis have not been submitted to any other University or Institute for the award of any other Degree or Diploma.*

Animesh Mukherjee
Assistant Professor
CSE, IIT Kharagpur

Niloy Ganguly
Professor
CSE, IIT Kharagpur

Date:

DECLARATION

I certify that

a. The work contained in this thesis is original and has been done by myself under the general supervision of my supervisors.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in writing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

f. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Sandipan Sikdar

ACKNOWLEDGMENTS

I would like to take the opportunity to thank everybody who inspired me, helped me and contributed directly or indirectly in realizing this thesis. I want to thank all my family members who continuously supported throughout my life. I like to thank my father who always remains a great inspiration for me. I like to thank my mother, who was also my childhood teacher and always stands by me for continuous support. My parents did a lot of hard work to support logistics for studies and always encouraged me in every decision. I like to specially thank my uncle, aunt and my little cute brother for their continuous support.

The other persons who made a huge contribution and without whom the thesis would not have been possible are my supervisors, Dr. Animesh Mukherjee and Prof. Niloy Ganguly. The long discussions with them not only helped me in culminating the problems of this thesis, but also helped me to understand the nuances of doing research. His constant support and encouragement during the entire tenure of my PhD motivated me to work harder for this thesis. I would like to say thank for all their support from the core of my heart. I would also like to thank Dr. Sanjukta Bhowmick (University of Nebraska, Omaha), one of my collaborators, who has not only helped me with technical guidance on my research, but more importantly, has taught me several ways of becoming a better researcher. I am very much thankful to all my co-authors who directly contributed in this thesis. I want to specially convey thank to Suhansanu Kumar and Sandipan Sikdar who have remained as good friends and coauthors in a large part of my thesis. I always

consider myself fortunate to be a part of the Complex Network Research Group which includes several good friends and excellent researchers. I want to extend my gratitude towards all my *CNeRG* lab mates Dr. Rajib Maiti, Dr. Saptarshi Ghosh, Dr. Rishiraj Saha Roy, Dr. Sudipta Saha, Kaustav Rudra, Parantapa Bhattacharya, Animesh Srivastava, Abir De, Sourav Dandapat, Sankarshan Mridha, Abhijnan Chakraborty, Madhumita Mallick, Souvik Sur, Suman Maity, Soumajit Pramanik and Mayank Singh. Continuous discussion with my lab mates not only helped me to enrich my knowledge in diverse fields but also was hugely refreshing. I express my sincere gratitude to the faculty members and staffs of the department. I would specially like to thank the members of my Doctoral Scrutiny Committee, Prof. S. Sarkar, Dr. P. Mitra, Prof. J. Mukherjee, Dr. A. Mukherjee, and our Head of the Department, Prof. R. Mall for their invaluable suggestions regarding the thesis. I was supported by Google India PhD Fellowship Award. I am thankful to several organizations such as Google India, ACM IARCS, Yahoo India and to the organizers of conferences such as ACM SIGKDD, ACM CODS, COMAD for financial assistance for attending the conferences.

I like to thank all my teachers throughout my life. I like to specially thank Mr. Dipankar Sannigrahi and Mr. Asish Kar who provided a continuous encouragement and confidence.

Finally, I am grateful to God for his blessings and for giving me the strength to persevere throughout the long and arduous journey towards a Ph.D.

Sandipan Sikdar

Kharagpur, India

Author's Biography

Sandipan Sikdar has received his B-Tech. degree in 2012 from Computer Science and Engineering department of Institute of Engineering and Management, Kolkata.

Publications from the Thesis

1. Sandipan Sikdar, Tanmoy Chakraborty, Soumya Sarkar, Niloy Ganguly and Animesh Mukherjee. " ComPAS: Community Preserving Sampling for Streaming Graphs". (communicated)

2. Sandipan Sikdar, Matteo Marsili and Animesh Mukherjee. "Unsupervised Ranking of Clustering Algorithms by INFOMAX". (communicated)

3. Sandipan Sikdar, Nitesh Sekhar, Matteo Marsili, Niloy Ganguly and Animesh Mukherjee. " On the effectiveness of multiple reviewers in a peer-review system: A case study of two high impact Physics journals". (communicated)

4. Sandipan Sikdar, Matteo Marsili, Niloy Ganguly and Animesh Mukherjee. " Influence of Reviewer Interaction Network on Long-term Citations: A Case Study of the Scientific Peer-Review System of the Journal of High Energy Physics", JCDL, Toronto, Canada, 2017.

5. Marcin Bodych, Niloy Ganguly, Tyll Kruger, Animesh Mukherjee, Rainer Seigmund-Schultze and Sandipan Sikdar. " Threshold based epidemic dynamics in systems with memory", Europhysics Letters, 116.4(2017):48004.

6. Sandipan Sikdar, Matteo Marsili, Niloy Ganguly and Animesh Mukherjee. " Anomalies in the peer-review system: A case study of the journal of High Energy Physics", CIKM, Indianapolis, USA, 2016, .

7. Sandipan Sikdar, Abhijnan Chakraborty, Anshit Choudhury, Gourav Kumar, S. Kumar, Abhijeet Patil, Niloy Ganguly and Animesh Mukherjee. " Identifying and Characterizing Sleeping Beauties on YouTube", CSCW 2016, San Francisco Poster highlights.

8. Sandipan Sikdar, Niloy Ganguly, Animesh Mukherjee. " Time series analysis of temporal networks",ÂăEuropean Physics Journal B topical issue

on Temporal Network Theory and Applications (2016), volume 89(1), 1-11, DOI: 10.1140/epjb/e2015-60654-7.

9. Sandipan Sikdar, Marcin Bodych, Rajib Ranjan Maity, Biswajit Paria, Niloy Ganguly, Tyll Kruger, Animesh Mukherjee. " On segmented message broadcast in dynamic networks", IEEE INFOCOM workshop (Netscicom), HongKong, 2015.

10. Tanmoy Chakraborty, Sandipan Sikdar, Niloy Ganguly, Animesh Mukherjee. " Citation Interactions among Computer Science Fields: A Quantitative Route to the Rise and Fall of Scientific Research",ÂăSocial Network Analysis and Mining (SNAM 2014), Springer, 4:187, pp. 1-18, DOI 10.1007/s13278-014-0187-3.

11. Tanmoy Chakraborty, Sandipan Sikdar, Vihar Tammana, Niloy Ganguly, Animesh Mukherjee. " Computer Science Fields as Ground-truth Communities: Their Impact, Rise and Fall", IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Niagara Falls, Canada, August 25-28, 2013.(Nominated for best paper)

ABSTRACT

To ensure efficient offload using WiFi network, we propose to use WiFi network as a local area CDN where popular files are distributed across the network in a local cache of WiFi APs. It ensures uninterrupted streaming service for users with high mobility. Another important contribution of this work is to propose a storage efficient spatial chunk distribution strategy considering human mobility model. Simulation results show that across a wide range of speed, system does a sizeable offload and in high traffic scenario performance degrades gracefully.

Traffic distribution across network is uneven. With such traffic distribution, ad-hoc association may result in poor system performance. To manage heterogeneity in load, we create a global view of load distribution across network. Moreover, we reduce pressing association control protocol to classical max flow algorithm which ensures maximum device association with knowledge of load distribution. Simulation results show that our proposed protocol can ac-

commodate more devices and provide better fairness in association compared to existing protocols.

As people do share their subscription credential with others, service providers lose their revenue and extra traffic is also introduced into already congested network. In this work, we propose an authentication scheme utilizing our daily activities which reduces shareability substantially. We have chosen a set of daily activities, which users are uncomfortable to share with others, to form authentication challenge. Simulation results show that an authentic user can successfully authenticate in 95% cases while even very close friends can not break-in in more than 5.5% cases.

Keywords: Community analysis, Permanence, Overlapping permanence, Community detection algorithms, Community evolution, Citation networks, Faceted recommendation system

Contents

Table of Contents xv

List of Figures xix

List of Tables xxv

1 Introduction 1
- 1.1 Major Challenges . . . 2
- 1.2 Objectives . . . 3
- 1.3 Constant Communities in Networks . . . 4
- 1.4 Permanence and Network Communities . . . 6
- 1.5 Analyzing Ground-truth Communities . . . 7
- 1.6 Community-based Applications . . . 9
- 1.7 Contributions . . . 11
  - 1.7.1 Constant Communities in Networks . . . 11
  - 1.7.2 Permanence and Network Communities . . . 13
- 1.8 Efficient Offload using WiFi Network . . . 15
  - 1.8.1 Analyzing Ground-truth Communities . . . 15
- 1.9 Challenges . . . 18
  - 1.9.1 Community-based Applications . . . 18
- 1.10 Organization of the Thesis . . . 20

2 Related Work 23
- 2.1 Survey on Community Detection and Evaluation . . . 23
  - 2.1.1 Non-overlapping Community Detection . . . 23
  - 2.1.2 Overlapping Community Detection . . . 31
  - 2.1.3 Community Scoring Metrics . . . 34
- 2.2 Survey on Post-hoc Analysis of Communities . . . 38
  - 2.2.1 Tracking Communities over Time . . . 38
  - 2.2.2 Analyzing Community Evolution in Networks . . . 40
  - 2.2.3 Community Structure in Link Prediction . . . 40
  - 2.2.4 Community Structure in Information Diffusion . . . 41
  - 2.2.5 Community Structure in Recommendation Systems . . . 41

3 Analyzing structural properties of temporal networks 43
3.1 Introduction . . . 43
3.2 Mapping Temporal network to time series . . . 45
3.3 Description of the dataset . . . 46
3.4 Analysis of time series . . . 48
3.4.1 Time domain characteristics . . . 48
3.4.2 Frequency domain analysis . . . 50
3.5 Prediction framework . . . 51
3.5.1 Selecting a window . . . 53
3.6 Prediction results . . . 56
3.6.1 Enhancing the prediction scheme through spectrogram . . . 59
3.7 An attack strategy using prediction framework . . . 60
3.8 Summary of this chapter . . . 61

4 Sampling temporal graphs 63
4.1 Introduction . . . 63
4.2 Problem Definition . . . 65
4.3 Proposed Algorithm: COMPAS . . . 67
4.4 Experimental Setup . . . 79
4.4.1 Sampling algorithms . . . 79
4.4.2 Datasets . . . 79
4.5 Evaluation . . . 80
4.5.1 Community-centric evaluation . . . 81
4.5.2 Graph-centric evaluation . . . 83
4.5.3 Effect of edge ordering and sample size . . . 84
4.6 Applications of compas . . . 85
4.6.1 Ranking community detection algorithms . . . 85
4.6.2 Selecting training set for online learning . . . 86
4.7 Summary of the chapter . . . 87

5 Diffusion in temporal networks 89
5.1 Introduction . . . 89
5.2 Diffusion model . . . 92
5.3 On Complete graph . . . 93
5.4 On d-regular tree: . . . 101
5.5 Innoculation strategy . . . 102
5.6 Broadcast algorithm based on the diffusion model . . . 104
5.6.1 Agent configuration and network setup . . . 104
5.6.2 Message configuration . . . 104
5.6.3 Transfer protocol . . . 105
5.6.4 Metrics of interest . . . 106
5.6.5 Broadcast algorithms . . . 106
5.6.6 Dynamic topology . . . 108

5.7 Experiment on different network topologies . . . 108
5.7.1 Blind push on different topologies . . . 109
5.7.2 Comparison of different broadcast strategies on Gnutella Topology . 112
5.8 Summary of the chapter . . . 113

6 Application of temporal network analysis in peer-review network 115
6.1 Introduction . . . 115
6.2 Dataset . . . 121
6.3 Reviewer-reviewer interaction network . . . 123
6.3.1 Degree (Deg) . . . 124
6.3.2 Betweenness centrality (BC) . . . 125
6.3.3 Closeness centrality (CC) . . . 125
6.3.4 Clustering coefficient (Clus) . . . 125
6.3.5 PageRank (PR) . . . 126
6.4 Supporting features . . . 126
6.4.1 Paper based features . . . 127
6.4.2 Review report based features . . . 129
6.4.3 Author based features . . . 132
6.4.4 Reviewer based features . . . 134
6.5 Determining the fate of the paper . . . 136
6.6 Irregular cases . . . 137
6.7 Publisher's views . . . 138
6.8 Conclusion . . . 139
6.9 Anomalous behavior . . . 140
6.9.1 Editor . . . 140
6.9.2 Reviewer . . . 144
6.10 Identifying anomalous Editors and Reviewers . . . 149
6.10.1 Editors . . . 149
6.10.2 Reviewers . . . 150
6.11 Profiling Anomalous Reviewers . . . 150
6.12 Conclusion . . . 151
6.13 Single vs multiple referee system . . . 152
6.13.1 Citation . . . 153
6.13.2 Top cited papers . . . 153
6.13.3 Conflict in review reports . . . 154
6.13.4 Impact of discordance . . . 157
6.14 Analyzing reviewer tendencies . . . 158
6.14.1 Factors determining performance of the referees . . . 159
6.14.2 Action with under-performing reviewers . . . 160
6.15 Forming reviewer groups . . . 161
6.15.1 Problem definition . . . 162
6.15.2 Methodology . . . 162
6.15.3 Evaluation . . . 165

6.16 Importance of the editor . . . 168
6.17 Conclusion . . . 171

7 Conclusion and Future Work **173**
7.1 Summary of Contribution . . . 173
7.1.1 Efficient Offload using WiFi Network . . . 173
7.1.2 Managing Heterogeneous Traffic . . . 175
7.1.3 Restricting Unauthorized Traffic . . . 176
7.2 Future Direction . . . 178
7.2.1 Efficient Offload using WiFi Network . . . 178
7.2.2 Managing Heterogeneous Traffic . . . 179
7.2.3 Restricting Unauthorized Traffic . . . 180

Bibliography **181**

A Appendix **201**

List of Figures

3.1 Converting Temporal network to Time series 45

3.2 (a) and (b) denote the status of the network at time t and $t+1$ respectively. For the edge (i, e) in t the corresponding edges emanating from i and e are (i, a) and (e, b). For the edge (x, y) they are (x, c) and (y, d). So the $Edge\_emer_t = \frac{2+2}{2} = \frac{4}{2}$ 46

3.3 (A), (C) and (E) represent the time series plots for INFOCOM 2006, SIGCOMM 2009 and High-school 2011 respectively. (B), (D) and (F) represent the power spectral density (PSD) corresponding to the frequency bins for INFOCOM 2006, SIGCOMM 2009 and High-school 2011 dataset respectively. Bins 1, 2 and 3 corresponds to frequencies <5, 5-15 and >15(Hz) respectively. 49

3.4 (A) and (C) represent the time series plots for High-school 2012, Hospital respectively. (B) and (D) represent the power spectral density (PSD) corresponding to the frequency bins for High-school 2012 and Hospital dataset respectively. Bins 1, 2 and 3 corresponds to frequencies <5, 5-15 and >15(Hz) respectively. 50

3.5 (a) and (b) denote the status of node i at time t and $t+k$ respectively. $NBR(i)_t = \{a, b, c\}$ and $NBR(i)_{t+k} = \{a, d, c, e\}$ $Correlation(i)_k = \frac{NBR(i)_t \bigcap NBR(i)_{t+k}}{NBR(i)_t \bigcup NBR(i)_{t+k}} = \frac{2}{5}$ where $NBR(i)_t \rightarrow$ the set of neighbors of i at time t 52

3.6 The average neighborhood-overlap value at different lags for (A)INFOCOM 2006, (B)SIGCOMM 2009, (C)Highschool 2012, (D)Highschool 2011 and (E)Hospital datasets. 52

3.7 Mean prediction error (%) across different properties for INFOCOM 2006 dataset for different similarity values. The lags corresponding to the similarity value are also provided. 55

3.8 The percentage error distribution of all the properties (time series) for (A) INFOCOM 2006 dataset, (B) SIGCOMM 2009 dataset, (C) High-school 2011. (D) High-school 2012 and (E) Hospital. X-axis represents percentage error and Y-axis represents probability. 55

3.9 (A) LPSD versus mean percentage error for all the properties across all the datasets. 57

3.10 The time series plot for number of active edges. The red and the green ellipses identify two transition and two silent phases respectively 58
3.11 The prediction framework. 58
3.12 Temporal robustness as a function of the fraction of removed nodes for (a) INFOCOM 2006 and (b) SIGCOMM 2009 datasets. 59

4.1 Toy example depicting various conditions handled by ComPAS when a streaming edge arrives. 69
4.2 Illustrative example of 3-clique percolation. Once node g is removed, a 3-clique is placed on node a. The clique percolates and accumulates all the nodes except node f which forms a singleton community along with $\{a, b, c, d, e\}$. 77
4.3 Average D-statistics value across all the topological measures. 83
4.4 Average D-statistics across all the topological measures for (a) different edge ordering and (b) sample size (n) of Youtube graph. 85

5.1 Proposed diffusion model for $k = 2$. 92
5.2 (a) The number of senders versus time steps for different network topologies (For ER random graph p is 0.005, for d-regular graph $d = 5$ and for BA scale-free graph m i.e., number of edges to attach from a new node to existing nodes is 5) and (b) The theoretical estimate and the simulated result for $d-$regular trees. The theoretical eastimate is obtained from equation 17. 93
5.3 $Av(T^*)$ and $Av(t_1)$ versus (a) the number of nodes with message size $k = 4$ and (b) k for fixed $d = 1000$. For both the plots $\hat{E}(t_1) = C * d^{\frac{k-1}{k}}$ where $C = (k!)^{\frac{1}{k}}\Gamma\left(1 + \frac{1}{k}\right)$ (refer to theorem 2). 100
5.4 (a) $Av(T^*)$ and $\hat{E}(T^*)$ (suitably scaled) for different values of k (b)$Av(T^*)$ versus edge probability in Erdos-Renyi random graph for $k = 2$ and $k = 3$. In both cases $Av(T^*)$ is normalized by $\hat{E}(t_1)$ which is $\sqrt{n}$ and $n^{\frac{2}{3}}$ for $k = 2$ and $k = 3$ respectively. 101

5.5 (a) $\frac{t_1^{\hat{p}} - t_1}{t_1}$ versus $\bar{p}$ for a complete graph with 1000 nodes and $k = 4$. (b) Number of nodes at each stage of infection versus time for complete graph of 1000 nodes with $k = 4$. Although the creation of infected nodes is slow initially, the number of partially infected nodes ($0 < l < 4$) increases rapidly. (inset) Magnified version of the same figure. 103
5.6 Pull attempts, successful pulls, push attempts, successful pushes versus time for gnutella1 network 107
5.7 (A) Broadcast time and (B) broadcast wastage versus different values of d for B-P. The parameters values are $n = 200, m = 4, k = 2$. 109
5.8 (A) Broadcast time and (B) broadcast wastage versus different values of d for B-P technique. The parameters values are $n = 200, m = 4, k = 2$. The inset in both the figures show the metrics of interest for the first few values of d to indicate the critical d more appropriately. 110

5.9 (A) Broadcast time and (B) broadcast wastage versus average degree for B-P . The parameters values are $n = 200, m = 4, k = 2$. 110
5.10 Average broadcast time and wastage versus x for gnutella1,gnutella2 and gnutella3 111
5.11 (A) Broadcast time versus message size (B) Wastage versus message size (C) Coverage versus message size for gnutella3 111
5.12 Gain in broadcast time of P-P-G over X-P-P and B-P [Inset shows gain in wastage] for (A)gnutella1 and (B)gnutella2 networks. Note: algo = B-P/X-P-P 111
5.13 Broadcast time for the gnutella snapshots and their sparser variants versus different values of message sizes for (a). gnutella1 and (b). gnutella2 113

6.1 **(Left)** Number of accepted and rejected papers per year from 1997 to 2015. **(Middle)** Citation distribution of both the accepted and the rejected papers. **(Right)** Distribution of number of reviews for accepted and rejected papers. 121
6.2 Peer-review system in JHEP 123
6.3 Cumulative distribution function (CDF) of citations received by the papers (accepted) reviewed by referees in top 25% and bottom 25% reviewers ranked according to (a) degree, (b) betweenness centrality, (c) closeness centrality (d) clustering coefficient values and (e) PageRank in the reviewer-reviewer interaction network. 124
6.4 **(Left)** Fraction of accepted and rejected papers in different citation buckets. **(Middle)** Average number of reviews for accepted and rejected papers in different citation buckets. For both **(Left)** and **(Middle)** bucket sizes are in increasing powers of 2. E.g. $\leq$ 1, 2, ($>$ 2 and $\leq$ 4) and so on. **(Right)** Average citation of the top 20 percentile papers for a given number of rounds of review request. 127
6.5 Average citation versus team size. Note that we segregate the papers based on the teams size and calculate the average citation. 128
6.6 Mean length of referee reports in terms of number of words at different rounds of review. Typically the buckets are < 100, ($\geq 100, < 200$) and so on 129
6.7 Sentiment score versus citations for both accepted and rejected papers for the years 2005, 2007 and 2009. We find similar trends for other years as well. 130
6.8 **(Left)** Mean number of citations per paper versus acceptance ratio. **(Middle)** Mean number of reviews per paper versus acceptance ratio. **(Right)** Mean sentiment score per paper versus acceptance ratio. Note that in each case we use acceptance ratio buckets where buckets correspond to acceptance ratio (≥ 0.1 and < 0.2), (≥ 0.2 and < 0.3) and so on. 132
6.9 Mean citation of the papers versus the average time(in days) between two submission. Note that we use time buckets where buckets correspond to < 100,(≥ 100 and < 200), (≥ 200 and < 300) and so on. 133

6.10 **(Left)** Mean number of citations per paper versus accept ratio. **(Middle)** Mean number of reviews per paper versus accept ratio. **(Right)** Mean sentiment score per paper versus accept ratio. Note that in each case we use accept ratio buckets where the buckets correspond to accept ratio (≥ 0.1 and < 0.2), (≥ 0.2 and < 0.3) and so on. 134
6.11 Mean citation of the papers versus (a) time since the last assignment for the assigned reviewer and (b) time taken by the reviewer to send the report. Note that for both the cases the times are divided into equi-sized buckets. For (a) bucket sizes are 100 each while for (b) it is 25. 135
6.12 CDF of (a) acceptance ratio of authors and (b) accept ratio of reviewers for accepted papers and highly cited rejected papers. 137
6.13 CDF of (a) accept ratio of reviewers, (b) acceptance ratio of authors (c) length of the review text (# words) for rejected papers and low cited accepted papers. 137
6.14 (a) Median Average citation (MAC) versus $MEAT$. $MEAT$ values are bucketed into 12 bins of equal size with range(1, 498.8).(b) MAC versus SRI and (c) MAC versus $RADI$. For both (b) and (c), the x-axis values are bucketed by values corresponding to (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on. 141
6.15 (a) Median Average citation versus SRI. SRI values are bucketed by values corresponding to (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on. (b) RDI versus number of declines. Increasing trend indicates higher the RDI, higher is the number of declines. 141
6.16 (a) Median Average citation (MAC) versus $MRAT$. $MRAT$ values are bucketed into 20 buckets of equal size with range(1,498.8),(b) MAC versus $MRSD$ (c) MAC versus TDI, (d) MAC versus EDI, (e) MAC versus MTD and (f) MAC versus AR. For both (c),(d) and (e), the x-axis values are bucketed by values corresponding to (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on. For (b) and (f) values (x-axis) are divided into 10 buckets of equal size. 142
6.17 (a) Median Average citation versus DFI. DFI values are bucketed by values corresponding to (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on. (b) Number of declines versus the month of the year. 148
6.18 Cumulative distribution function of the average citations for the two sets of editors (anomalous and normal). 149
6.19 Cumulative distribution function of the average citations for the two sets of reviewers (anomalous and normal). 150
6.20 Mean citation profile of the reviewers in the three categories. 151
6.21 Citation distribution of the multi-refereed and single-refereed papers for (Left) accepted and (Right) rejected papers for JHEP dataset. 152
6.22 Citation distribution of the multi-refereed and single-refereed papers for (Left) accepted and (Right) rejected papers for JSTAT dataset. 152

6.23 Fraction of multi-refereed papers in the top k most cited papers where $k = 1, 50, 100, 500, 1000, 2000$. 153
6.24 Fraction of cases where the reviewers disagreed with respect to (1) length (2) sentiment and (3) content for (a) JHEP and (b) JSTAT datasets. 154
6.25 Cumulative distribution function of citations for single refereed papers and concordant multi-refereed papers in terms of (a) length, (b) sentiment and (c) content for JHEP. 157
6.26 Cumulative distribution function of citations for single refereed papers and discordant multi-refereed papers in terms of (a) length, (b) sentiment and (c) content for JSTAT. 158
6.27 Mean citation versus (a) accept ratio (b) assignment delay buckets for the JHEP dataset. Note that the papers are segregated into accept ratio/delay bins and the mean citation is calculated for each bin. Typical bin sizes for accept ratio are < 0.1,(≥ 0.1 and < 0.2) and so on while for delay the sizes are < 100, (≥ 100 and < 200) and so on. 158
6.28 Mean citation versus (a) accept ratio (b) assignment delay buckets for JSTAT dataset. Note that the papers are segregated into accept ratio/delay bins and the mean citation is calculated for each bin. Bin sizes are same as figure 6.27. 160
6.29 Mean citation for papers belonging particular class combination with respect to (a) accept ratio (b) time since last assignment. For example LL would represent a paper reviewed by referees both belonging to class L. . . 161
6.30 Block diagram demonstrating the work flow of our system. 165
6.31 Mean true positive (averaged over accepted and rejected papers) value while recommending a set of reviewer groups for papers across different values of (a) α and (b) crossover rate. Experiments repeated for both JHEP and JSTAT datasets. $k = 15$ 168
6.32 Mean true positive (averaged over accepted and rejected papers) value with recommendation by our method (G_A_based) and the baseline. Results are noted for (a) JHEP and (b) JSTAT. $k = 15$. 169
6.33 True positive value at each point of recommendation of the top 25 percentile (based on citation) accepted papers for JHEP dataset. The papers are sorted by date of submission and x-axis denotes the paper number in the sequence. 170

List of Tables

3.1 Properties of the dataset used. 48
3.2 Network property and the fraction of predictions with percentage error $\leq 20\%$ without (with) spectrogram analysis. The cases where more than 80% of the points have prediction error $\leq 20\%$ have been highlighted in bold font and the cases where on using spectrogram analysis the improvement is more than 5% have been underlined. 57

4.1 Important notations used in this paper. 67
4.2 Summary of the D-statistics (the lower, the better) values of the topological measures for all the datasets. For Youtube we present all the results, while for the rest we provide the average D-statistics and standard deviation (SD). `ComPAS` truns out to the second best algorithm after GA (the most informed static graph sampling algorithm for which the sample is obtained from the aggregated graph and Louvain is run on the sample, thus serving as the strict baseline). Top two values for each average result is highlighted. 81
4.3 NMI between the ground-truth and community structure obtained from individual sampling algorithms for all datasets. 82
4.4 Summary of D-statistics for different graph properties. For Youtube we present all the results, while for the rest we provide only average D-statistics (top three results in each average case are highlighted). 84
4.5 Rank correlation of community detection algorithms based on the performance on the sample (generated from individual sampling methods) and the original graph. (b) Performance of SVM using the training set obtained from sampling methods. 86

5.1 Summary of the notations used. 95

6.1 General information of the dataset. 122
6.2 Mean values of percentages of various categories of words in review reports of high and low cited papers where the means differ significantly. . . 130
6.3 The F-statistics value for all the features used for predicting the long-term citation of the paper. 133
6.4 Features used for detecting anomalies. 147

6.5 Jaccard similarity between the cases identified as discordant by the different metrics. 156
6.6 Proportion of discordant cases (length, sentiment and content) in each reviewer class combination with respect to (accept ratio, time since last assignment). 161
6.7 True positive (TP) and true negative (TN) values across accepted and rejected papers measured for different values of k. Results are reported for JHEP and JSTAT datasets. 169

Chapter 1

Introduction

A complex network is a graph-based representation of the interactions amongst entities that take place in the real world. Examples include social networks such as acquaintance networks [8], collaboration networks [161], technological networks such as the Internet [67] and the World Wide Web [7], and biological networks such as neural networks [229], and metabolic networks [107]. Real networks are not random and they usually exhibit *inhomogeneity* [**?**], indicating the coexistence of order and organization. Furthermore, the distribution of links also shows inhomogeneity, both globally and locally, describing the phenomenon that nodes naturally cluster into groups and links are more likely to connect nodes within the same group. This phenomenon tells us that the organization of such complex network is modular. Network scientists call this organization as the *community structure* of networks. Though there is a lack of consensus in the definition of communities, most popular and well-accepted definition suggests that: communities are the subsets of vertices within which vertex-vertex connections are dense, but between which connections are less dense [**?**]. A figurative sketch and a real-world community structure are shown in Figure **??**. Analysis of such communities is essential to understand the structural and the functional organizations of the network.

1.1 Major Challenges

Detecting communities is of prime importance in sociology, biology and computer science disciplines where systems are often represented as graphs. This problem is very hard and not yet satisfactorily solved, despite the huge effort of a large interdisciplinary community of scientists working on it over the past one and half decades (see [70] for the reviews). Besides this, several other challenges have been encountered during the analysis of community structure in large networks, some of which are as follows:

- The goodness of community detection algorithms (see [74] for a review) is often objectively measured according to how well they achieve the optimization. Modularity [**?**] is a widely accepted metric for measuring the quality of community structure identified by various community detection algorithms. However, a growing body of research have begun to explore the limitations of maximizing modularity for community identification and evaluation; three such limitations include – resolution limit [**?**], degeneracy of solutions and asymptotic growth of the modularity value. Therefore, a new goodness measurement metric needs to be formulated that can overcome (or minimize) such limitations.

- Due to the limitations of the goodness measures (such as modularity) described above, researchers often rely on manual inspection in order to evaluate the detected communities. For each detected community an effort is made to interpret it as a "real" community by identifying a common property or external attribute shared by all the members of the community. Such anecdotal evaluation procedures require extensive manual effort; therefore these are non-comprehensive and are limited to small networks. Therefore, a possible solution would be to find a reliable definition of explicitly labeled ground-truth communities.

- Although there is a large volume of research on community detection, systematic post-hoc analysis of the communities, which can unfold interesting characteristic properties of various real systems, is missing in the literature. For instance, temporal community interactions on a longitudinal scale (i.e, with the progress of time) often unveil the opportunity to analyze the rise and fall of dominant clusters in different time points. This analysis might be helpful in detecting the trending topics in

Twitter, identifying major research fields in different scientific domains, information diffusion among scientific communities [**?**] etc.

Given this scenario, it is clear that we need to develop a better understanding of community structure in various types of large networks. The goal of our research is to study different aspects of community analysis in complex networks that mainly focus on two major directions – (i) identification of realistic communities in different large networks and (ii) leveraging such community structure for developing various applications.

1.2 Objectives

To deal with all the challenges mentioned above, we identify four major issues mentioned below that contribute to different chapters of the thesis.

(i) Investigating the dependence of community detection algorithms on vertex ordering: Here we intend to study the variation of results produced by the algorithms due to different vertex orderings. Moreover, we posit that despite any vertex ordering, there exist some invariant groups in each network whose constituent vertices always remain together. In particular, we ask the following questions – what does the invariance of the results tell us about the network structure? what is the significance of these invariant substructures in a network? how are they related with the actual community structure of a network?

(ii) Formulating a new metric for community analysis: Most of the community scoring functions are global, thus do not imply anything about the vertices of a network. We believe that the individual constituent vertices in a community do not belong to the community with equal strength. Further, there is a lack of a proper quantitative indicator that would entail the true modular structure of a network. For instance, the highest modularity in the Jazz network is 0.45 and that of the Western USA power grid is 0.98 [**?**]. However, it has been observed that Jazz has a much stronger community structure than the power grid [**?**]. Therefore, formulation of a vertex-centric measure for community analysis that correctly indicates the presence of community structure in a network is needed.

Here we intend to ask few fundamental questions pertaining to the community analysis of a network – is the membership of vertices in a community homogeneous (which has been the common consensus so far)? do we need to check the eligibility of a network for community detection prior to running the community detection algorithm? can one formulate a metric that suitably reduces the limitations of the existing metrics for community detection?

(iii) Analyzing real-world community structure: Several works on detecting and tracking communities in a temporal environment have been conducted [74]. However, the interactive patterns of detected communities over a temporal scale still remain unexplored mainly due to the lack of standard ground-truth community structure of a network. The availability of ground-truth communities allows to explore a range of interesting characteristics of a time-varying systems. For example, deep understanding of the connectivity structure in and across ground-truth communities could lead to realistic community detection methods. Here, we focus on a typical real network, *citation network*, whose nodes correspond to scientific articles and links correspond to the citations from citing papers to cited papers. We aim at investigating different aspects of this network such as – how do the communities form in this network? what do the topological features of citation network tell us? what can we learn from them? what kind of trends are observed over-time in these networks? how often do authors publish and collaborate?

(iv) Developing community based applications: Once the community structure is detected from a network, an immediate question might arise that how this information can help us in building real applications. Citation profiles over time can be shown to group in different communities, which can be further used to develop more accurate citation prediction models. Further, it is possible to arrange citations into semantic communities which can facilitate developing a full-fledged faceted recommendation system of scientific articles.

1.3 Constant Communities in Networks

An automatic way of detecting the communities from networks has attracted much attention in recent years and many community detection algorithms have been proposed.

Most of these algorithms are based on the maximization of a quality function known as modularity, which measures difference between the fraction of edges in the network that connect vertices of the same type (within community type) and the expected value of the same quantity in a network with the similar community divisions but random connection between vertices (see Section 2.1.1). Modularity maximization is an NP-hard problem [34], and most algorithms use heuristics. For several reasons related to the modularity, as well as the non-determinism of the algorithms or randomness in initial configuration, such algorithms often produce different partitions of similar quality, and there is no reason to prefer one above another. Besides, such approaches may produce communities with a high modularity in networks which have no community structure, e.g., random networks. This is related to the instability of algorithms: small perturbations of the input graph can significantly influence the output.

Here, we investigate the effect of input ordering on two non-deterministic agglomerative methods for modularity maximization – (i) CNM algorithm [**?**] and (ii) Louvain method [**?**]. Both these methods are based on combining appropriate pairs of vertices to increase modularity. Based on these results, we posit that the permutation of the vertices is a key point for obtaining high modularity. A bad permutation can lead to sub-optimal combination of vertex pairs that in turn can affect the communities obtained. The notion of stability is governed by the inherent compartmental structure of the nodes in a network. Our intuition is based on the fact that some vertices always persist within same communities despite any combinatorial ordering of input edge sequence. Those vertices may have some intrinsic connectivity property that forces them not to share other communities under any circumstance. We call such groups of vertices as *constant communities* and the constituent vertices as *constant vertices*. We observe that if these constant vertices are grouped together in the pre-processing step, it significantly improves the accuracy of hierarchical clustering technique by increasing the modularity. We further analyze the properties of constant communities in order to identify the characteristic that keep them together independent of the order of the vertices in which the community detection algorithm is fed in. In particular, we observe that constant vertices experience minimum "pull" from external nodes in the network. Further, we present a case study on phoneme network and illustrate that constant communities, quite strikingly, form the core functional units of the larger communities.

1.4 Permanence and Network Communities

Community detection algorithms primarily deal with identifying densely-connected units from within large networks. So far, the common consensus in the analysis of the community structure is that the community membership is homogeneous, i.e., each node belongs to one or more communities with equal extent. Therefore, less attention has been paid in analyzing individual vertices in a community, and a community is mostly considered as a whole. Here we argue that the community membership of vertices is *heterogeneous*; where few vertices have more involvement into the community and others have less. To quantify the membership of a vertex, we need a proper local vertex-based metric. Modularity is a widely accepted global metric for measuring the quality of community structure identified by various community detection algorithms. However, a growing body of research have begun to explore the limitations of maximizing modularity for community identification and evaluation; three such limitations include – resolution limit [72], degeneracy of solutions [**?**] and asymptotic growth of the modularity value [**?**].

To address these issues, we here propose a novel vertex-level metric called *permanence* (Perm) for analyzing disjoint communities which is built on the notion of relative pull experienced by a vertex from its neighbors that lie external to its own community. The value of permanence indicates the extent to which a vertex belongs to a community. We show that this metric as compared to other standard measures, namely modularity, conductance and cut-ratio qualifies as a better community scoring function for evaluating the detected community structures from both synthetic and real-world networks. We demonstrate that the process of maximizing permanence produces communities that concur with the ground-truth structure of the networks more accurately than the modularity based and other approaches. Finally, we show that maximizing permanence (named as MaxPerm) can effectively reduce the limitations associated with modularity maximization as well as can indirectly help in inferring the community quality of a network.

Further, we formulate a generalized version of this metric called *overlapping permanence* (abbreviated as OPerm) that, although is developed for overlapping community, translates to the non-overlapping case under special boundary conditions. Note that this is one of the rarest formulations which can be useful for both non-overlapping and overlapping

community analysis. Since every vertex gets scored by this metric, it can be used to rank the vertices within a community as well as can give an overview of the belongingness of nodes in the community. Detailed experimentation demonstrates OPerm's superiority over other state-of-the-art scoring metrics in terms of performance as well as its resilience to minor perturbations. We also present an algorithm, MaxOPerm, to detect communities based on maximizing OPerm. Over a test suite of synthetic and six large real-world networks we show that MaxOPerm outperforms six state-of-the-art algorithms in terms of accurately predicting the ground-truth labels. We also demonstrate that MaxOPerm is resistant to degeneracy of solution. Further, we introduce the resolution limit problem in the context of overlapping communities and show that an algorithm which can maximize OPerm can effectively tackle the problem.

1.5 Analyzing Ground-truth Communities

Most of the existing works on community analysis have concentrated on developing and improving the algorithms for discovering communities. Evaluating the performance of such algorithms is incomplete without comparing the detected output with the actual ground-truth community structure of the network under investigation. However, such ground-truth community structure is limited in number. Moreover, availability of such community structure of a labeled network would unveil the opportunity to investigate its characteristics and functionality thoroughly. To this purpose, we particularly focus on a scientific network, called *citation network*, whose nodes indicate scientific articles and links correspond to the citations. We gather all the papers in computer science domain published in the last fifty years and indexed by Microsoft Academic Search[1]. Each paper comes along with various bibliographic information – the title of the paper, a unique index number, its author(s) etc. Each individual community in a citation network is naturally defined by a research field – i.e., acting as ground-truth. Then we study the interactions among these communities through citations in real time which unfold the landscape of dynamic research trends in the computer science domain over the last fifty years. We quantify the interaction in terms of a metric called *inwardness* that captures the effect

[1] `http://academic.research.microsoft.com/`

of local citations to express the degree of *authoritativeness* of a community (research field) at a particular time instance. Several arguments to unfold the reasons behind the temporal changes of inwardness of different communities are put forward using exhaustive statistical analysis. The measurements (importance of field) are compared with the project funding statistics of NSF and it is found that the two agree to a considerable extent.

As a second step we quantify the interdisciplinarity of a research field through four indicative measures. Three of the indicators, namely *Reference Diversity Index* (*RDI*), *Citation Diversity Index* (*CDI*) and *Membership Diversity Index* (*MDI*) are directly related to the topological structure of the citation network. The last feature called the *Attraction Index* of a field is based on the propensity of the new researchers to start research in a particular field. Further, to check the significance of these features in characterizing interdisciplinarity, we rank the fields based on the value of each of the features separately. Next, we propose an unsupervised classification model that can efficiently cluster the core and the interdisciplinary fields based on the similarity of the feature sets mentioned above. To understand the evolutionary landscape of a core field vis-a-vis an interdisciplinary field, we conduct a case study on one popularly accepted interdisciplinary field (WWW) and one core field (Programming Languages). The results attest to the conclusion that the interdisciplinarity occurs through cross-fertilization of ideas between the fields that otherwise have little overlap as they are studied independently. The conclusion that popularity of the interdisciplinary research now-a-days overshadows the core fields is strengthened on analyzing the core-periphery organization of the citation network at different time periods. We observe that the core region of a domain is gradually dominated by the more applied fields with interdisciplinary fields steadily accelerating towards the core.

The rich citation dataset further allows us to conduct an author-centric analysis. In particular, we analyze the diverse scientific careers of researchers in order to understand the key factors that could lead to a successful career. Essentially, we intend to answer some specific questions pertaining to a researcher's scientific career – what are the local and the global dynamics regulating a researcher's decision to select a new field of research at different points of her entire career? what are the suitable quantitative indicators to measure the diversity of a researcher's scientific career? We propose two entropy-based metrics to measure a researcher's choice of research topics. Experiments with large computer science bibliographic dataset reveal that there is a strong correlation between the diversity of the ca-

reer of a researcher and her success in scientific research in terms of the number of citations. We observe that while most of the researchers are biased toward either adopting diverse research fields or concentrating on very few fields, a majority of the highly cited researchers tend to follow a typical "scatter-gather" policy – although their entire careers are immensely diverse with different types of fields selected at different time periods, they remain focused primarily in at most one or two fields at any particular time point of their career.

1.6 Community-based Applications

The group of homogeneous entities can be useful in several applications. Here we particularly focus on two major applications that are built on the citation networks and publication datasets. Prior to that, we study another important aspect of a scientific article, its growth of citation counts over time after the publication. A common consensus in the literature is that the citation profile of published articles in general follows a universal pattern – an initial growth in the number of citations within the first two to three years after publication followed by a steady peak of one to two years and then a final decline over the rest of the lifetime of the article. This observation has long been the underlying heuristic in determining major bibliometric factors such as the quality of a publication, the growth of scientific communities, impact factor of publication venues etc. We study the citation network once again and notice that the citation count of the articles over the years follows a remarkably diverse set of patterns – a profile with an initial peak (PeakInit), with distinct multiple peaks (PeakMul), that exhibits a peak late in time (PeakLate), that is monotonically decreasing (MonDec), that is monotonically increasing (MonIncr) and that cannot be categorized into any of the above (Oth)). The papers following same citation profile are assumed to form separate community. We systematically investigate the important characteristics of each of these categories.

Then we leverage this category information in order to develop a prediction model that predicts future citation count of a scientific article after a given time interval of its publication. We propose to categorize the complete set of data samples into different subparts each of which corresponds to one type of citation pattern mentioned earlier. This approach is commonly termed as *stratified learning* in the literature where the members of the stratified

space are divided into homogeneous subgroups (aka strata) before sampling. We develop a *two-stage prediction model* – in the first stage, a query paper is mapped into one of the strata using a Support Vector Machine (SVM) approach that learns from a bunch of features related to the author, the venue of the publication and the content of the paper; in the second stage, only those papers corresponding to the strata of the query paper are used to train a Support Vector Regression (SVR) module to predict the future citation count of the query paper. For the same set of features available at the time of publication, the two-stage prediction model remarkably outperforms (to the extent of 50% overall improvement) the well-known baseline model. Our two-stage prediction model produces significantly better accuracy in predicting the future citation count of the highly-cited papers that might serve as an useful tool in early prediction of the seminal papers that are going to be popular in the near future. We also show that including the first few years of citations of the paper into the feature set can significantly improve the prediction accuracy especially in the long term.

Finally, we arrange citations into semantic communities based on the relation of a cited paper with the citing paper. We use this grouping to propose for the first time a framework of faceted recommendation for scientific articles, *FeRoSA* which apart from ensuring quality retrieval of scientific articles for a particular query paper, also efficiently arranges the recommended papers into different facets (categories). Our methodology is based on a principled framework of random walks where both the citation links and the content information are systematically taken into account in recommending the relevant results. First, citation links are categorized into four classes/facets, namely Background, Alternative Approaches, Methods and Comparison. Following this, for a particular query paper, we collect an initial pool of papers containing nearest citation-based neighborhoods and papers having high content-similarity with the query paper, and make an induced graph individually for each facet. Next, a random walk with restarts is performed from the query paper on each of the induced subgraphs and a ranked list of papers is obtained. We further prepare another ranked list of papers based on the content similarity. The final ranking is obtained in a principled way by combining multiple ranked lists. Our method is easy to implement and has very elegant and principled way of retrieving the relevant results irrespective of the choice of the facets. Human experts are asked to judge the recommendations of the competing systems. Experimental results show that our system outperforms the baseline systems with respect to different standard measures which are

used to evaluate a recommendation system. In terms of overall precision, FeRoSA achieves an improvement of 29.5% compared to the best competing system. We also evaluate and compare the results separately for different facets (average overall precision of 0.65) and model parameters to have a thorough understanding of the performance of the system.

1.7 Contributions

In this thesis, we consider *community analysis in complex network* as a prime objective, which has been one of the active research topic for quite some time in different branches of science including computer science, physics, mathematics and biology. Despite a large volume of research in this area, few fundamental problems have remained unanswered or have not been solved satisfactorily. Here we attempt to analyze such problems. Moreover, we focus on *citation network* and study different structural and functional aspects of this network. Finally, we design two applications based on the publication dataset which leverage the community information of the underlying network. A brief report (which we shall elaborate in the forthcoming chapters) on these studies and the results obtained thereby, are presented below.

1.7.1 Constant Communities in Networks

Although enormous effort has been devoted to design efficient community detection algorithms, most of these algorithms follow a general framework – these algorithms try to optimize certain objective functions (such as modularity) by grouping vertices, which results in the partitioning of the vertices in the network. However, most of these algorithms are highly dependent on the ordering in which the vertices are processed as a result of which the algorithms produce different outputs in different iterations for a particular network. An exhaustive study of this phenomenon reveals the following interesting results:

(a) We conduct this experiment on a set of scale-free networks and observe that while the vertex orderings produce very different set of communities, some groups of vertices

are always allocated to the same community for all different orderings. We define the group of vertices that remain invariant as *constant community* and the vertices that are part of the constant communities as *constant vertices*.

(b) Although constant communities are detected using the outputs obtained from certain community detection algorithms, we notice that these groups are the invariant part of a network, irrespective of the heuristic being used to detect the communities.

(c) Another issue that has not been studied earlier is whether a network at all contains community structure or not. For instance, a random network or a grid network does not have strong community structure as compared to the ring of cliques. Therefore, we propose a metric, called *sensitivity* (based on the number of constant communities within a network) which efficiently demonstrates how community-like a network is. Later in Chapter 4, we use this metric to measure the *degeneracy of solutions* of an algorithm, which although has been studied several times, is quantified here for the first time.

(d) Constant communities are quite different from the actual community structure of a network. For instance, constant communities do not always have more internal connections than external connections. Rather, the strength of the community is determined by the number of different external communities to which it is connected. Therefore, we characterize constant vertices by a metric called *relative pull*, which indicates that the constant vertices do not experience a significant "pull" from any of the external communities that will cause them not to migrate, and, therefore, their propensity to remain within their own communities is high.

(e) Further, we show that if these constant communities are identified prior to any community detection, and each constant community is combined into a super-vertex, it not only increases the efficiency of any community detection algorithm, but also reduces the variability of the final output.

(f) Finally, we conduct a case study on a specific type of labeled linguistic network constructed from the speech sound inventories of the world's languages. We discover constant communities from this network and observe that each such graph represents a natural class, i.e., a set of consonants that have a large overlap of the features. Such groups are frequently found to appear together across languages.

1.7.2 Permanence and Network Communities

Motivated by the earlier study on constant communities, we further investigate the community structure of real-world networks. Since many real-world communities are based on subjective measurements (as opposed to a formal definition), often the optimum value of the parameters are successful in identifying only a fraction of the "ground-truth" communities. Moreover, as observed in the phenomena of resolution limit [72] and degeneracy of solutions [**?**], the optimum parameter value sometimes produces intuitively incorrect solutions in ideal networks. As a response to these issues, new metrics are being regularly proposed [**?**, 95], that either produce more accurate results on a certain subclass of networks and/or can address some of these inherent problems.

Despite the on-going research in this area, an important question is whether it is always reasonable to assign every individual vertex to a community. Not all networks possess community structures of equal strength. For example, a network composed of several sparsely connected dense cliques will have strong communities whereas a grid will not have any community structure at all, and between these two extremes there exist communities of different strength as per the network structure. As of now, there is no community detection metric that can measure to what extent a vertex is a part of a community. One of the reasons for this deficiency is that the optimum value of the parameters such as modularity is not exactly related to whether the network possesses a strong community, but rather tries to identify the best community assignment, for any given network. Indeed, most algorithms output a set of communities regardless of whether the network (such as a grid) possesses a community structure or not. A corollary to this problem is that given a suboptimal answer we cannot estimate how close we are to the correct result and in the absence of ground-truth community structure, we cannot even judge whether the obtained answer is reliable or not. These are serious limitations for a field that regularly encounters new applications and datasets.

To manage this huge mobile data traffic, we need to explore all the avenues that come across. We have identified three major issues from the scope discussed above related to mobile data traffic management as our objective of this thesis.

Efficient Offload using WiFi Network: Cellular network is becoming heavily

congested. Capacity of 3G network is becoming inadequate to carry current traffic load, hence increasingly stress is given towards deployment of 4G network. About 4G network, Allen Nogee, a principal analyst at the In-Stat research firm commented that "They're not really for speed, they're not really for voice, they're for capacity"[2]. A more immediate solution towards reducing congestion of overloaded cellular network is by offloading it to the Wi-Fi network. In this thesis, we work to explore the opportunity of enhancing Wi-Fi offloading performance through *data caching*. An interesting statistics of YouTube is that 10% of contents are accessed for 80% of times. That means even if we can do some special arrangement about this popular file then traffic load can reduce heavily. Hence one important objective of this thesis is to devise a distributed caching policy of popular files to enhance Wi-Fi offload performance.

Managing Heterogeneous Traffic: From different study of human mobility models, it is known that traffic across network is not evenly distributed. Simple RSSI based association strategy will overload some access points while most of the access points remain under utilized and many users will remain unassociated. One objective of this thesis is to develop an association control protocol which can handle the uneven load distribution and accommodate maximum clients.

Restricting Unauthorized Traffic: There are many paid services which grant access to valuable content like movies, news, songs etc. People used to share subscription credential of such services either under social pressure or to reduce per head subscription charge. As an effect this increases unauthorized traffic as well as loss of revenue for service providers. Hence, the final objective of this thesis is to restrict unauthorized traffic by means of restricting password sharing.

[2]http://it-code-news.blogspot.in/2010/03/it-news-headlines-ars-technica-30032010.html

1.8 Efficient Offload using WiFi Network

To accommodate the huge cellular traffic, cellular network providers are depending significantly on Wi-Fi network, so that they can offload less priority cellular data to Wi-Fi network. According to Cisco's prediction, Wi-Fi traffic will be dominating over Cellular network traffic by the end of 2018. Fig. **??** shows the prediction of offloaded traffic in Wi-Fi network.

1. We identify a precise rank order among the vertices within a community by arranging them into a core-periphery structure based on OPerm; this rank order can be further used as an input for various other applications (e.g., initiator selections in message spreading).

2. Maximizing Perm (maximizing OPerm) is more successful in finding ground-truth communities as compared to state-of-the-art algorithms.

3. Community detection using maximizing Perm (maximizing OPerm) can overcome the problem related to resolution limit, degeneracy of solutions, in many networks. Moreover, the value of Perm (OPerm) is relatively independent of the size of the network.

1.8.1 Analyzing Ground-truth Communities

Even though modeling network communities is a fundamental problem, our understanding of networks at the level of these communities has been relatively less. Moreover, the lack of reliable ground-truth makes the evaluation of such models extremely difficult. Here we study the connectivity structure of ground-truth communities of a real network, citation network of computer science domain whose nodes correspond to the scientific articles and

links correspond to the citations. Our work is based on a large scale citation network where we can reliably define the notion of ground-truth communities. In this network, each paper (node) is marked by its relevant research field; thus citation interactions among papers within a same research field are relatively higher than across fields. These fields therefore act as ground-truth communities in the network. The availability of the reliable ground-truth communities has a profound effect, such as it allows us to understand the connectivity structure of the ground-truth communities and the interaction among these communities that has the potential to portray a significantly better picture of the underlying systems.

(a) To start with, we first study the temporal interaction among communities in citation network by defining a metric called "authoritativeness" which measures the impact of a community in a particular time period. These patterns of interaction, when analyzed carefully, reveal various interesting elements that are either directly or indirectly related to the overall decline in the interest in a field followed by the rise of interest in another. One of the most striking observations is that in almost all cases, the field constituting the current "hottest" area of research within the domain is overtaken in the immediate future by its strongest competitor.

(b) We further investigate the cause of such focus shifts from different and possibly orthogonal directions and observe that (a) the density of high impact publications within a field plays a pivotal role in pulling as well as sustaining the field at the forefront, (b) certain fields produce a huge number of citations (i.e., act as hubs) for a particular field and, thereby, push it to the forefront; an abrupt fall in the number of such received citations, in many cases, triggers the decline of the field currently at the forefront, (c) inception of seminal papers in a field might trigger the emergence of a field at the forefront, and (d) the extent of team work (both within and across continents) in the form of joint publications seem to significantly contribute to the shape of the evolutionary landscape.

(c) A careful analysis of the funding trends by NSF (National Science Foundation of the United States of America) shows that our results correlate very well with the number of proposals submitted in each field while they correlate moderately well with the actual funding decisions.

A common consensus among researchers is that interdisciplinarity is one of the key factors in doing research at current times. However, a pertinent question deals with identifying appropriate indicators of interdisciplinarity. Using a set of citation based indicators, here we investigate the evolution of the extent of interdisciplinary research in computer science. For this, we study the citation network from different orthogonal directions, namely citation and reference patterns of a paper, overlapping membership of the papers in different research communities, inclination of the researchers to adopt new fields, and propose several indices to quantify the degree of interdisciplinarity of a field. The new indices of interdisciplinarity corroborate with the hypothesis that the emergence of interdisciplinarity occurs through cross-fertilization of ideas between the sub-fields that otherwise have little overlap as they are studied independently. At the end, we analyze the core-periphery organization of citation networks and arrive to the conclusion that with the advancement of interdisciplinary research, the core part of the network is also changing from theoretical towards more applied fields of research. Some of our observations are as follows.

(a) The practice of interdisciplinarity in citations occurs mainly between related scientific communities, and this phenomenon has been witnessed to tremendously increase over the last few years.

(b) Few fields such as Data Mining, WWW, Natural Language Processing, Computational Biology, Computer Vision, Computer Education provide clear indications of interdisciplinarity in terms of all the metrics proposed here.

(c) We develop an unsupervised classification algorithm using these metrics to identify the core and the interdisciplinary fields.

(d) Core-periphery analysis on the citation network shows that the interdisciplinary fields are accelerating steadily toward the core of computer science domain.

(e) For already very interdisciplinary fields, such as Data Mining, the indicators may have a certain “saturation” effect forcing it towards the core region of the computer science domain.

1.9 Challenges

With this new trend and new avenue a set of challenges also becomes prominent. In this section, we discuss important challenges of Wi-Fi network.

(a) The average behavior indicates that a researcher tends to adopt few research fields in her entire research career, and she seems to prefer to work simultaneously on all of them together.

(b) A highly-cited researcher tends to work in many fields over her entire career but remains confined to one or few fields in each time window. However, the number of such researchers is very less in our dataset.

(c) The researchers who have tried various fields in the entire career as well and in each successive time period, get low citations.

1.9.1 Community-based Applications

Once the community structure of a network is detected, a natural question would be as to how can we use this information in designing real systems. We use publication dataset, citation network and the community structure, and design two applications – future citation count prediction of a paper after publication and faceted recommendation system for scientific articles. The major contributions from this study are mentioned below.

1. We first start analyzing the citation profile of the papers and reveal six different patterns – a profile with an initial peak (PeakInit), with distinct multiple peaks (PeakMul), that exhibits a peak late in time (PeakLate), that is monotonically decreasing (MonDec), that is monotonically increasing (MonIncr) and that can not be categorized into any of the above (Oth)).

2. While analyzing the characteristic of these categories, we observe that most of the papers in PeakInit (64.35%) and MonDec (60.73%) categories are published in conferences, whereas papers belonging to PeakLate (60.11%) and MonIncr (74.74%) categories are mostly published in journals. Hence, if a publication starts receiving greater attention or citations at a later part of its lifetime, it is more likely to be published in a journal and vice versa.

3. We observe that papers in MonDec are vastly affected by the self-citation phenomenon, i.e., around 35% of papers in MonDec would have been in the 'Oth' category had it not been due to the self-citations. The result also agrees with the observation that MonIncr category is least affected by self-citations, followed by PeakLate, PeakMul and PeakInit in that order.

4. We study the stability of each category by analyzing the migration of papers from one category to others over time. We observe that apart from the Oth category, MonDec seems to be the most stable, which is followed by PeakInit. However, papers which are assumed to fall in Oth category quite often turn out to be MonIncr papers in the later time periods.

5. We analyze the core-periphery organization of the citation network and observe that PeakMul category gradually leaves the peripheral region over time and mostly occupies the innermost shells. PeakInit and MonDec show almost similar behavior with a major proportion of papers in inner cores in the initial year but gradually shifting towards peripheral regions. On the other hand, MonIncr and PeakLate show expected behavior with their proportion increasing in the inner shells over time indicating their rising relevance as time progresses.

6. Our proposed framework for future citation count prediction incorporates a stratified learning approach in the traditional framework which in turn remarkably enhances the overall performance of the prediction model.

7. Our two-stage model produces significantly better accuracy in predicting the future citation count of the highly-cited papers that might serve as an useful tool in early prediction of the seminal papers that are going to be popular in the near future.

8. The faceted recommendation system, FeRoSA is primarily built on the semantic

annotation of citations in citation network. While evaluating the system based on expert judgment, FeRoSA achieves an overall precision (OP) of 0.65, 29.5% higher than the next best system. Thus, the recommendations generated by our framework are found to be of high quality even if the method is very simple to implement.

9. FeRoSA also achieves a reasonably high precision for the query papers with low citations (OP of 0.57 with the next best system having an OP of 0.46).

10. We observe once again that although FeRoSA is designed for faceted recommendation, it significantly outperforms the baselines due to an inherent stratification (dividing the general graph into facet-wise subgraphs) which leads to a better ranking.

1.10 Organization of the Thesis

The thesis is organized into seven chapters.

Chapter 2 presents a detailed literature survey on the state-of-the-art in community analysis for different networks and their usage in different applications.

Chapter 3 centers around our first objective of constant communities in complex networks. We detect constant communities in a brute-force manner and study their structural properties. We show that identifying constant communities prior to any community detection enhances the performance of any community detection algorithms.

Chapter 4 investigates in detail our second objective, i.e., formulation of permanence and overlapping permanence for community analysis. We further develop two community detection algorithms using these metrics.

Chapter 5 explains our third objective of analyzing the ground-truth community structure of citation network. We study three subproblems pertaining to citation network. First, we unfold the rise and fall of scientific research in computer science domain over last fifty years. Second, we propose four metrics to quantify the degree of interdisciplinarity of a research field. Third, we study the field adoption process of a researcher over her entire research career.

Chapter 6 presents our final objective of designing different community-based applications. In particular, we design two systems: (i) future citation count prediction of a scientific article after publication, and (ii) a faceted paper recommendation system for scientific articles.

Chapter 7 concludes the thesis by summarizing the contributions and pointing to a few topics of future research that have opened up from this work.

Chapter 2

Related Work

In this chapter, we discuss relevant studies related to the objectives of this thesis. Particularly, the literature review is conducted in two broad directions: first, we shall describe the metrics and methods used in community detection, and second, we shall elaborate the analysis of community structure and its usage in various applications.

2.1 Survey on Community Detection and Evaluation

In this section, we survey the current literature on the community identification problem and other closely related problems. First, we review the work on identifying non-overlapping and overlapping communities in different networks. Following this, we present various metrics used to evaluate the community structures.

2.1.1 Non-overlapping Community Detection

A wide spectrum of community detection methods have been proposed to detect disjoint communities from static networks. Interested readers are encouraged to read the following survey papers: Fortunato [70], Lancichinetti, Fortunato [124], Harenberg et al. [89]. All

these algorithms can be roughly divided into the following categories.

Traditional Methods

(i) Graph partitioning: The problem of graph partitioning consists of dividing the vertices in different groups of predefined size, such that the number of edges lying between the groups is minimal. The number of edges running between clusters is called *cut size*. There are several algorithms that can do a good job, even if their solutions are not necessarily optimal [112, 181]. Another popular technique is the spectral bisection method [19], which is based on the properties of the spectrum of the Laplacian matrix. Graphs can be also partitioned by minimizing measures that are affine to the cut size, like *conductance* [31], *ratio cut* [232] and *normalized cut* [202]. Algorithms for graph partitioning are not good for community detection, because it is necessary to provide as input the number of groups and in some cases even their sizes, about which in principle has no prior information.

(ii) Hierarchical clustering: Most of the real-world graphs have a hierarchical structure, i.e., display several levels of grouping of the vertices, with small clusters included within large clusters, which are in turn included in larger clusters, and so on. In such cases, one may use hierarchical clustering algorithms [91], i.e. clustering techniques that reveal the multilevel structure of the graph. Hierarchical clustering techniques can be classified in two categories: Agglomerative (bottom-up) and Divisive (top-down) algorithms. Hierarchical clustering has the advantage that it does not require a prior knowledge of the number and size of the clusters. However, it does not provide a way to discriminate between many partitions obtained by the procedure, and to choose that or those partitions which better represent the community structure of the graph. The results of the method depend on the specific similarity measure adopted. The procedure also yields a hierarchical structure by construction, which is rather artificial in most cases, since the graph at hand may not have a hierarchical structure at all [163].

(iii) Partitional clustering: Partitional clustering assumes that the number of clusters is predefined, say k. The points are embedded in a metric space, so that each vertex is a point and a distance measure is defined between pairs of points in the space. The distance is a measure of dissimilarity between vertices. The goal is to separate the points in k clusters

so as to maximize/minimize a cost function based on distances between points and/or from points to *centroids*. Few such functions include minimum k-clustering, k-clustering sum, k-center, k-median. The most popular partitional technique in the literature is k-means clustering [144]. Extensions of k-means clustering to graphs have been proposed by some authors [26, 96]. The limitation of partitional clustering is the same as that of the graph partitioning algorithms: the number of clusters must be specified at the beginning, the method is not able to derive it.

(iv) Spectral clustering: Spectral clustering includes all methods and techniques that partition the set of vertices into clusters by using the eigenvectors of matrices or other matrices derived from it. In particular, the objects could be points in some metric space, or the vertices of a graph. Spectral clustering consists of a transformation of the initial set of objects into a set of points in space, whose coordinates are elements of eigenvectors. The set of points is then clustered via standard techniques, like k-means clustering. The first contribution on spectral clustering was by Donath and Hoffmann [60]. There are three popular methods of spectral clustering: unnormalized spectral clustering and two normalized spectral clustering techniques, proposed by Shi and Malik [202] and by Ng et al. [166] respectively. However, Nadler and Galun [155] discussed the limitations of this method such as it cannot successfully cluster datasets that contain structures at different scales of size and density.

Divisive Algorithms

The philosophy of divisive algorithms is to detect the edges that connect vertices of different communities and remove them, so that the clusters get disconnected from each other. The most popular algorithm is the one proposed by Girvan and Newman [**?**]. The method is historically important, because it marked the beginning of a new era in the field of community detection. Here edges are selected according to the values of measures of *edge betweenness centrality*. Tyler et al. proposed a modification of the Girvan-Newman algorithm, to improve the speed of the calculation [219]. Another fast version of the Girvan-Newman algorithm has been proposed by Rattigan et al. [185]. Here, a quick approximation of the edge betweenness values is carried out by using a network structure index, which consists of a set of vertex annotations combined with a distance

measure. In this line, gradually two community detection algorithms have been proposed for overlapping community detection, namely the concept of vertex splitting [178] and CONGA (Cluster Overlap Newman-Girvan Algorithm) [85].

Modularity-based Algorithms

Modularity (introduced by Newman and Girvan [**?**]) is by far the most used and best known quality function. It is based on the idea that a random graph is not expected to have a cluster structure, so the actual strength of clusters is revealed by the comparison between the actual density of edges in a subgraph and the density one would expect to have in the subgraph if the vertices of the graph were attached regardless of community structure. This expected edge density depends on the chosen null model, i.e., a copy of the original graph retaining some of its structural properties but not community structure. Modularity can then be written as follows:

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(C_i, C_j) \tag{2.1}$$

where the sum runs over all pairs of vertices, A is the adjacency matrix, m the total number of edges of the graph, k_i the degree of vertex i, the δ-function yields one if vertices i and j are in the same community ($C_i = C_j$), zero otherwise. By assumption, high values of modularity indicate good partitions. All clustering techniques that require modularity, directly and/or indirectly can be classified as follows.

(i) Greedy techniques: The first algorithm devised to maximize modularity was a greedy method proposed by Newman [164]. It is an agglomerative hierarchical clustering method, where groups of vertices are successively joined to form larger communities such that modularity increases after the merging. Later on, Clauset et al. [**?**] proposed more efficient data structure like *max-heaps* to make Newman's algorithm faster. Danon et al. [54] suggested to normalize the modularity variation ΔQ produced by the merger of two communities by the fraction of edges incident to one of the two communities, in order to favor small clusters. Wakita and Tsurumi [226] noticed that, due to the bias towards large communities, the fast algorithm by Clauset et al. is inefficient, because it yields

very unbalanced dendrograms. Another trick to avoid the formation of large communities was proposed by Schuetz and Caflisch [198]. A different greedy approach has been introduced by Blondel et al. [**?**] (mostly known as *Louvain algorithm*), for the general case of weighted graphs. The method consists of two phases. First, it looks for "small" communities by optimizing modularity locally. Second, it aggregates nodes of the same community and builds a new network whose nodes are the communities obtained in the first stage. These steps are repeated iteratively until a maximum of modularity is attained. The modularity maxima found by the method are better than those found with the greedy techniques by Clauset et al. [**?**] and Wakita and Tsurumi [226].

(ii) Simulated annealing: Simulated annealing [116] is a probabilistic procedure for global optimization used in different fields and problems. It was first employed for modularity optimization by Guimera et al. [87]. Its standard implementation combines two types of moves: *local moves*, where a single vertex is shifted from one cluster to another, taken at random; *global moves*, consisting of mergers and splits of communities. Splits can be carried out in several distinct ways. The best performance is achieved if one optimizes the modularity of a bipartition of the cluster, taken as an isolated graph. Global moves reduce the risk of getting trapped in local minima and they have proven to lead to much better optima than using simply local moves [148].

(iii) Extremal optimization: Extremal optimization is a heuristic search procedure proposed by Boettcher and Percus [29], in order to achieve an accuracy comparable with simulated annealing, but with a substantial gain in computer time. It is based on the optimization of local variables, expressing the contribution of each unit of the system to the global function being studied. This technique was used for modularity optimization by Duch and Arenas [62]. Generally, this technique maintains a good trade-off between accuracy and speed, although it sometimes leads to poor results on large networks with many communities [70].

(iv) Other optimization strategies: Agarwal and Kempe [2] suggested maximization of modularity within the framework of mathematical programming. Chen et al. [46] used integer linear programming to transform the initial graph into an optimal target graph consisting of disjoint cliques, which effectively yields a partition. Berry et al. [24] formulated the problem of graph clustering as a *facility location problem*, that attempts to minimize a cost function based on a local variation of modularity. Lehmann and Hansen [131]

optimized modularity via *mean field annealing* [176]. Genetic algorithms [98] have also been used to optimize modularity.

Modifications of Modularity

In the most recent literature on graph clustering several modifications and extensions of modularity can be found. Modularity can be easily extended to graphs with weighted edges [162], directed graphs [132]. Kim et al. [114] proposed a different definition based on diffusion on directed graphs, inspired by Google's PageRank algorithm. Rosvall and Bergstrom raised similar objections [194]. Gaertler et al. [78] introduced quality measures based on modularity's principle of the comparison between a variable relative to the original graph and the corresponding variable of a null model. Another generalization of modularity was recently suggested by Arenas et al. [12]. Expressions of modularity for bipartite graphs were suggested by Guimera et al. [88] and Barber [18]. However, community detection using modularity has certain issues including resolution limit, degeneracy of solutions and asymptotic growth [**?**]. To address these issues, multi-resolution versions of modularity [13] were proposed to allow researchers to specify a tunable target resolution limit parameter. He et al. [95] considered different community densities as good quality measures for community identification, which do not suffer from resolution limits. Furthermore, Lancichinetti and Fortunato [125] stated that even those multi-resolution versions of modularity are not only inclined to merge the smallest well-formed communities, but also to split the largest well-formed communities; some of these problems were addressed and partially resolved by Chan et al. [44] recently.

Dynamic Algorithms

Here we describe methods employing processes running on the graph, focusing on spin-spin interactions, random walks and synchronization.

(i) Spin models: The Potts model is among the most popular models in statistical mechanics [236]. It describes a system of spins that can be in different states. Based on this idea, Reichardt and Bornholdt [186] proposed a method to detect communities that

maps the graph onto a zero-temperature q-Potts model with nearest-neighbor interactions. In another work, Son et al. [206] have presented a clustering technique based on the *Ferromagnetic Random Field Ising Model* (FRFIM).

(ii) Random walk: Random walks [103] can also be useful to find communities. If a graph has a strong community structure, a random walker spends a long time inside a community due to the high density of internal edges and consequent number of paths that could be followed. Zhou [248] used random walks to define a distance between pairs of vertices: the distance d_{ij} between i and j is the average number of edges that a random walker has to cross to reach j starting from i. A different distance measure between vertices based on random walks was introduced by Latapy and Pons [179] where the distance is calculated from the probabilities that the random walker moves from a vertex to another in a fixed number of steps. Hu et al. [100] designed a graph clustering technique based on a signaling process between vertices, somewhat resembling diffusion. Dongen, in his PhD thesis, described the *Markov Cluster Algorithm* (MCL) [222].

Statistical Inference based Methods

Statistical inference aims at deducing properties of data sets, starting from a set of observation and model hypotheses. If the data set is a graph, the model, based on hypotheses on how vertices are connected to each other, has to fit the actual graph.

(i) Generative models: Most of the methods adopted Bayesian inference [235], in which the best fit is obtained through the maximization of a likelihood (generative models). Hastings [92] chose a *planted partition model* of network with communities. Newman and Leicht [165] proposed a similar method based on a mixture model and the expectation-maximization technique. Another technique similar to that by Newman and Leicht was designed by Ren et al. [188] based on the *group fractions*. Maximum likelihood estimation was used by Čopič et al. [53] to define an axiomatization of the problem of graph clustering and its related concepts. Hofman and Wiggins [97] proposed a general Bayesian approach to the problem of graph clustering. The main limitation of these methods comes from high memory requirements.

(ii) Information theoretic approach: The modular structure of a graph can be considered as a compressed description of the graph to approximate the whole information contained in its adjacency matrix. Rosvall and Bergstrom [192] envisioned a communication process in which a partition of a graph in communities represents a synthesis of the full structure that a signaler sends to a receiver, who tries to infer the original graph topology from it. The same idea is the basis of an earlier method by Sun et al. [208], which was originally designed for bipartite graphs evolving in time. In a recent paper, Rosvall and Bergstrom [194] pursued the same idea of describing a graph by using less information than that encoded in the full adjacency matrix. The goal is to optimally compress the information needed to describe the process of information diffusion across the graph. Chakrabarti [38] has applied the minimum description length principle to put the adjacency matrix of a graph into the (approximately) block diagonal form representing the best trade-off between having a limited number of blocks, for a good compression of the graph topology, and having very homogeneous blocks, for a compact description of their structure.

Other Methods

Here we describe some algorithms that do not fit in the previous categories. Raghavan et al. [182] designed a simple and fast method based on *label propagation*. The main advantage of the method is the fact that it does not need any information on the number and the size of the clusters. It does not need any parameter, either. In a recent paper, Tibély and Kertész [216] showed that the method is equivalent to finding the local energy minima of a simple zero-temperature kinetic Potts model. A recent methodology introduced by Papadopoulos et al. [172], called *Bridge Bounding*, is similar to the L-shell algorithm, but here the cluster around a vertex grows until one "hits" the boundary edges. Another method, where communities are defined based on a local criterion, was presented by Eckmann and Moses [63]. Long et al. [140] have devised an interesting technique that is able to detect various types of vertex groups, not necessarily communities. Zarei and Samani [244] remarked that there is a symmetry between community structure and anti-community (multipartite) structure, when one considers a graph and its complement, whose edges are the missing edges of the original graph.

2.1.2 Overlapping Community Detection

There has been a class of algorithms for network clustering, which allow nodes belonging to more than one community. As discussed in [238], we shall discuss the proposed algorithms by categorizing them into five classes.

Clique Percolation Algorithms

The clique percolation method (CPM) is based on the assumption that a community consists of overlapping sets of fully connected subgraphs and detects communities by searching for adjacent cliques. *CFinder* is the implementation of CPM, whose time complexity is polynomial in many applications [170]. However, it also fails to terminate in many large social networks. Following this, CPMw [68] introduces a subgraph intensity threshold for weighted networks. Only k-cliques with intensity larger than a fixed threshold are included into a community. Instead of processing all values of k, SCP [120] finds clique communities of a given size. Despite their conceptual simplicity, an usual criticism is that CPM-like algorithms are more like pattern matching rather than finding communities since they aim to find specific, localized structure in a network.

Link Partitioning Algorithms

On the other hand, few algorithms trying to partition links instead of nodes to discover community structure have also been explored. A node in the original graph is called overlapping if links connected to it are put in more than one cluster. Ahn et al. [6] proposed a method where links are partitioned via hierarchical clustering of edge similarity. Evans [66] projected the network into a weighted *line graph*, whose nodes are the links of the original graph, then applied the node partitioning algorithm. *CDAEO* [237] provides a post-processing procedure to determine the extent of overlapping. Kim and Jeong [113] extended the map equation method [194] to the line graph, which encodes the path of the random walk on the line network under the Minimum Description Length principle.

Local Expansion and Optimization Algorithms

Algorithms utilizing local expansion and optimization rely on growing a natural community or a partial community [126]. Baumes et al. [20] proposed a two-step process: first, nodes are ranked according to some criterion, then the process iteratively removes highly ranked nodes until small, disjoint cluster cores are formed. Lancichinetti et al. [124] proposed an algorithm called *LFM* which expands a community from a random seed node to form a natural community until a fitness function becomes locally maxima. Havemann et al. proposed MONC [94] which uses the modified fitness function of LFM that allows a single node to be considered a community by itself. Lancichinetti et al. further proposed *OSLOM* [128] that tests the statistical significance of a cluster [27] with respect to a global null model (i.e., the random graph generated by the configuration model [151] during community expansion).

Chen et al. [42] proposed selecting a node with maximal node strength based on two quantities: belonging degree and the modified modularity. Cazabet et al. [37] proposed *iLCD* which is capable of detecting both static and temporal communities. Given a set of edges created at some time step, iLCD updates the existing communities by adding a new node if its number of second neighbors and number of robust second neighbors are greater than expected values.

Seeds are very important for many local optimization algorithm. A clique has been shown to be a better alternative over an individual node as a seed. Shen et al. [201] in their algorithm *EAGLE* used the agglomerative framework to produce a dendrogram. Similar to EAGLE, *GCE* [130] identifies maximum cliques as seed communities.

Fuzzy Detection

Fuzzy community detection algorithms quantify the strength of association between all pairs of nodes and communities. Nepusz [157] modeled the overlapping community detection as a nonlinear constrained optimization problem which can be solved by simulated annealing methods. Zhang et al. [245] proposed an algorithm based on the spectral clustering framework [**?**]. There is another algorithm called *FOG* [58] which tries to infer groups based on link evidence. Similar mixture models can also be constructed as a generative

model for nodes [77]. In *SSDE* [145], the network is first mapped into a d-dimensional space using the spectral clustering method. A Gaussian Mixture Model (GMM) is then trained via Expectation-Maximization algorithm. The number of communities is determined when the increase in log-likelihood of adding a cluster is not significantly higher than that of adding a cluster to random data which is uniform over the same space.

Non-negative Matrix Factorization (NMF) is a feature extraction and dimensionality reduction technique in machine learning that has been adapted to community detection. Zhang et al. [246] replaced the feature vector used in NMF with the diffusion kernel, which is a function of the Laplacian of the network. Later Zarei et al. [243] showed that the result would be better if the matrix is defined by the correlation matrix of the columns of the Laplacian. Recently, Yang and Leskovec [**?**] proposed BIGCLAM which is also based on NMF approach.

Ding et al. [58] extended the *affinity propagation clustering algorithm* [76] for overlapping community detection, in which clusters are identified by representative exemplars. First, nodes are mapped as data points in the Euclidean space via the commute time kernel (a function of the inverse Laplacian). The similarity between nodes is then measured by the cosine distance.

Agent-based and Dynamical Algorithms

The label propagation algorithm [182] in which nodes with same label form a community, has been extended to overlapping community detection by allowing a node to have multiple labels. Gregory proposed *COPRA* [**?**] in which each node updates its belonging coefficient by averaging the coefficients from all its neighbors at each time step in a synchronous fashion. Xie et al. [239] developed *SLPA* which is a general speaker-listener based information propagation process. A game-theoretic framework is proposed in Chen et al. [45], in which a community is associated with a Nash local equilibrium. A process in which particles walk and compete with each other to occupy nodes is presented by Breve et al. [35]. Different from SLPA and COPRA, this algorithm takes a semi-supervised approach. It requires at least one labeled node per class.

Other Methods

CONGO [85] extends Girvan and Newman's divisive clustering algorithm [**?**] by allowing a node to split into multiple copies. Gregory [86] also proposed to perform disjoint detection algorithms on the network produced by splitting the node into multiple copies using the split betweenness. Zhang et al. [247] proposed an iterative process that reinforces the network topology and propinquity that is interpreted as the probability of a pair of nodes belonging to the same community. The propinquity between two vertices is defined as the sum of the number of direct links, number of common neighbors and the number of links within the common neighborhood. Kovács et al. [118] proposed an approach focusing on centrality-based influence functions.

2.1.3 Community Scoring Metrics

Another important aspect of community detection is to evaluate the detected community structure. If we know the actual community structure of a network, it would be easier to evaluate the detected communities just by comparing them with the actual community structure. However, most of the time, collecting the actual ground-truth community structure is difficult, and therefore we rely on the structural property of the community structure. In this section, we first describe such topology-based community evaluation metrics and then briefly mention few popular validation metrics that are used to compare the detected community with the ground-truth structure.

Topology-based Community Evaluation Metric

Several metrics for evaluating the quality of community structure have been introduced. The most popular and widely accepted is Modularity [**?**] (see Equation 4.1). Recently, Fortunato and Barthelemy [73] presented a *resolution limit* problem of modularity, essence of which is that optimizing modularity will not find communities smaller than a threshold size, or weight [25]. The threshold depends on the total number (or total weight) of edges in the network and on the degree of interconnectedness between communities. Moreover, Good et

al. [**?**] showed another problem of Modularity called *degeneracy of solutions* that this measure admits an exponential number of high-modularity but structurally distinct solutions from a single graph. They also studied the limiting behavior of maximizing modularity for one model of infinitely modular networks (*asymptotic growth*), showing that it depends strongly both on the size of the network and on the number of modules it contains, i.e., as we add more modules to the network, the height of the modularity function converges to 1. To address the resolution limit problem, multi-resolution versions of modularity [13] were proposed to allow researchers to specify a tunable target resolution limit parameter. Lambiotte [123] proposed different types of multi-resolution quality functions to tackle resolution limit problem. Dongxiao et al. [95] considered different community densities as good quality measures for community identification, which do not suffer from resolution limits.

In the context of overlapping community evaluation, people attempted to redefine modularity for overlapping community structure. Shen et al. [201] introduced EQ, an adaptation of Newman's modularity function designed to support overlapping communities. The equation for EQ strongly resembles the original modularity function as follows:

$$EQ = \frac{1}{2m} \sum_{c \in C} \sum_{i \in c, j \in c} \frac{1}{O_i O_j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \tag{2.2}$$

where m is the number of edges in the graph, C is the set of communities, and O_v is the number of communities to which the node v belongs. The presence of an edge between two nodes v and w is represented as the value in the corresponding position of the adjacency matrix A_{vw}.

On the other hand, recently Lázár et al. [129] provided a more complex and potentially more accurate evaluation of the goodness of an overlapping community structure as follows:

$$Q_{ov} = \frac{1}{K} \sum_{r=1}^{K} \left[\frac{\sum_{i \in c_r} \frac{\sum_{j \in c_r, i \neq j} A_{ij} - \sum_{j \notin c_r} A_{ij}}{di \cdot s_i}}{n_{c_r}} \cdot \frac{n^e_{c_r}}{\binom{n_{c_r}}{2}} \right] \tag{2.3}$$

where K is the number of communities, n_{c_r} is the number of nodes and $n^e_{c_r}$ is the number of edges that the rth cluster c_r contains respectively, d_i is the degree of node i, s_i denotes the number of clusters where i belongs to and A is the adjacency matrix. Note that, since the density of clusters containing one single node (when n_{c_r} = 1) is not defined (because

$\binom{1}{2}$ is undefined), the modularity value is set to be zero.

Ahn et al. [6] described two simple measures to quantify the quality of a community structure. The first one is *Community Coverage* which simply counts the fraction of nodes that belong to at least one community of three or more nodes. A size of three is chosen since it is the smallest nontrivial community. This measure provides an estimate of how much of the network is analyzed. The second measure is *Overlap Coverage* which counts the average number of memberships in nontrivial communities (size at leas three) that nodes are given.

Ground-truth Based Community Validation Metrics

Evaluating the quality of a detected partitioning or cover is nontrivial, and extending evaluation measures from disjoint to overlapping communities is rarely straightforward. In this section, we discuss some of the popular evaluation metrics which are often used to compare the detected partition with the ground-truth communities.

(i) Purity (PU): The Purity measure [**?**] is historically the first one used in the context of community detection. Let us assume that $X = \{x_1, x_2, ..., x_I\}$ and $Y = \{y_1, y_2, ..., y_J\}$ be the two partitions of the same set. To denote the cardinalities, we use n for the total number of elements in the partitioned set, and $n_{ij} = |x_i \cap y_j|$ for the intersection of two parts. We also note $n_{i+} = |x_i|$ and $n_{+j} = |y_j|$ the part size. The purity of a part x_i relative to the other partition Y is expressed as $PU(x_i, Y) = \max_j \frac{n_{ij}}{n_{i+}}$. The total purity of partition X relative to partition Y is obtained as follows: $PU(X, Y) = \sum_i \frac{n_{i+}}{n} PU(x_i, Y)$.

It is important to notice the purity is not a symmetric measure. Therefore, the usual approach is to take the harmonic mean of $PU(X, Y)$ and $PU(Y, X)$. The upper bound is 1, it corresponds to a perfect match between the partitions. The lower bound is 0 and indicates the opposite.

(ii) Rand Index (RI): The Rand Index [183] is a way of comparing disjoint clustering solutions that is based on pairs of the objects being clustered. Two solutions are said to agree on a pair of objects if they each put both objects into the same cluster or each into

different clusters. The Rand Index can then be formalized as follows:

$$RI = \frac{(a+d)}{N} \tag{2.4}$$

where N is the number of pairs of objects, a is the number of times the solutions agree on putting a pair in the same cluster and d is the number of times the solutions agree on putting a pair in different clusters. That is, the Rand Index is the number of pairs that are agreed on by the two solutions divided by the total number of pairs.

An improvement to the Rand Index is the *Adjusted Rand Index* (ARI) [101] which adjusts the level of agreement according to the expected amount of agreement based on chance.

(iii) Omega Index: The Omega Index [52] builds on both the Rand Index and Adjusted Rand Index by accounting for disjoint solutions and correcting for chance agreement. The Omega Index considers the number of clusters in which a pair of objects is together. The observed agreement between two partitions $S1$ and $S2$ is calculated by: $Obs(S1, S2) = \sum_{j=0}^{min(J,K)} A_j/N$, where J and K represent the maximum number of clusters in which any pair of objects appears together in partitions 1 and 2, respectively, A_j is the number of the pairs agreed by both partitions to be assigned to number of clusters j, and N is again the number of pairs of objects. The expected agreement is given by: $Exp(S1, S2) = \sum_{j=0}^{min(J,K)} N_{j1}N_{j2}/N^2$, where N_{j1} is the total number of pairs assigned to number of clusters j in partition 1, and N_{j2} is the total number of pairs assigned to number of clusters j in partition 2. The Omega Index is then calculated as

$$Omega(S1, S2) = \frac{Obs(S1, S2) - Exp(S1, S2)}{1 - Exp(S1, S2)} \tag{2.5}$$

The highest possible score of 1 indicates that two solutions perfectly agree on how each pair of objects is clustered.

(iv) Normalized Mutual Information (NMI): The problem of comparing different community structures can be overcome by computing the Normalized Mutual Information (NMI) [224]. Let C be the confusion matrix. Also let N_{ij} (elements of the confusion matrix C) be the number of nodes in the intersection of the original community i and the generated community j. If C_A denotes the number of the communities in the ground truth, C_B the number of the generated communities by an arbitrary approach, N_i the sum of

row i, N_j the sum of column j, and N the sum of all elements in C, then the NMI score between the ground truth partition A, and the generated partition B can be computed as shown in the following equation.

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij} log\frac{N_{ij}N}{N_i N_j}}{\sum_{i=1}^{C_A} N_i log\frac{N_i}{N} + \sum_{j=1}^{C_B} N_j log\frac{N_j}{N}} \tag{2.6}$$

The values of NMI range between 0 and 1 where 0 refers to no match with the ground truth and 1 refers to a perfect match. Recently, McDaid et al. [**?**] also provided a modified version of NMI, called *ONMI* for evaluating overlapping community structures.

However, Labatut [122] argued that these measures are not completely relevant in the context of network analysis, because they ignore the network connectivity. He proposed the modified versions of these measures where misplacing high degree vertices would incur higher penalty compared to low degree vertices. The modified formulations of NMI, ARI and Purity are the weighted versions, namely Weighted-NMI (W-NMI), Weighted-ARI (W-ARI) and Weighted-Purity (W-PU).

2.2 Survey on Post-hoc Analysis of Communities

In this section, we survey the current literature pertaining to the analysis of detected communities and how this community information can be used in the development of various systems.

2.2.1 Tracking Communities over Time

In real world, the membership of communities tend to change gradually. Backstrom et al. observed this on the communities of LiveJournal users and communities of conference publications on DBLP [16]. So it is important not only to detect communities but also to track the changes in membership over time. The questions are: which community in

one snapshot metamorphoses into anther in the next snapshot? How and what fraction of membership changes in between? The problem of tracking communities is motivated by a problem in behavioral ecology in studying animals that live in fission-fusion societies such as zebra and the Asiatic wild ass [209]. A natural question is which group that we observe today is the same as that was previously observed. Groups in this setting are manifestations of perpetual communities. Inversely, a community is a consistent string of groups seen on different days. This is the problem of tracking communities over time. Loosely speaking, it is about how to string different groups from the same day into communities which span over multiple days.

There is a handful of work specifically on the problem of tracking communities over time. Berger-Wolf and Saia [23] proposed a framework which defines communities as independent local patterns. There, a community (or, metagroup) is a sequence of groups which have sufficiently high similarity. The similarity between two groups is the number of common members normalized by the sizes of the two groups. Characteristics of communities are studied via community-based statistical measures such as number of all possible communities, their sizes and life spans. Also, they proposed an approach to study the survival of the communities via finding a critical set of groups whose removal leaves only short-lived communities.

Spiliopoulou et al. [207] proposed a framework, called *MONIC*, for tracking communities over time. The framework utilizes a similarity function of groups at different time steps. The function takes into account the number of common members, the sizes of the groups, and the time decay between the groups. Then, two groups are strung together as being in the same community if their similarity is above a certain threshold. The framework not only strings groups into communities but also detects splitting and merging of communities by a separate set of threshold parameters.

Tantipathananandh et al. [215] proposed the first framework which rigorously formulates the problem of tracking communities as an optimization problem. Although the appealing aspect of this framework is the social costs model which has its roots in the social sciences view of group dynamics [174], the framework has a strong assumption that all time steps must have the same length. Tantipathananandh et al. [214] further introduced an improved framework which can handle data with time steps of variable length.

2.2.2 Analyzing Community Evolution in Networks

Slightly different from the task of community tracking is the study of the evolution of communities over time. This problem attracts a lot of research interest due to its enormous applications in real-world scenario. For example, in a blog network we might wish to detect which communities of blogs are relatively stable in size over a period of time [137]. In a mobile phone network, changes in community size over a timeframe can reveal calling patterns and customer churns [84, 168]. In other contexts such as scientific collaboration networks, communities of researchers that span many years suggest long-term research collaboration [168]. Such communities can be further investigated to identify researchers in particular fields who are consistently productive over a period of time [137]. Previous work along this line analyzed community changes using a life-cycle model comprising events such as birth, death, expand, contract, merge, and split [84,168]. Asur et al. [14] emphasized the life-cycle of nodes, an emphasis that is impractical in networks with millions of nodes and irrelevant when an overview of how communities evolve is required. Palla et al. [168] used the above events to quantify the evolution of a phone call network and a coauthorship network, whereas Greene et al. [84] used the events to investigate community evolution in a phone call network. Except for [168], little attention has been paid to modeling an event as a function of time. Recently, Lužar et al. [143] studied interdisciplinarity of research communities detected in the coauthorship network of Slovenian scientists over time.

2.2.3 Community Structure in Link Prediction

The information of community of nodes can also be leveraged in the task of link prediction. Clauset et al. [49] proposed a method to determine the hierarchical structure of a network by using MCMC sampling to create a binary dendrogram that joins nodes into groups. Since this method got introduced, a variety of similar methods and models have been proposed. Valverde-Rebaza and de Andrade Lopes [220] described experiments to analyze the viability of applying the within and inter cluster (WIC) measure for predicting the existence of a future link on a large-scale online social network. They further proposed three measures for the link prediction task which take into account all different communities that users belong to [221]. Sachan and Ichise [196] proposed to build a link predictor in a co-authorship

network, and showed that the knowledge of a pair of researchers lying in the same dense community can be used to improve the accuracy of our predictor further. Recently, Fenhua et al. [136] proposed a link prediction method based on clustering and global information.

2.2.4 Community Structure in Information Diffusion

Communities are vehicles for efficiently disseminating news, rumors, and opinions in human social networks. Several approaches studied this phenomenon using the community structure of the network. Belak et al. [22] studied information diffusion across communities and showed that one can achieve high community-based spreading using an efficient targeting strategy. Nematzadeh et al. [156] used the linear threshold model to systematically study how community structure affects global information diffusion. Kimura et al. [115] used community analysis to find influential nodes for information diffusion on a social network under the independent cascade model. Weng et al. [233] focused on understanding interactions between community structures and information diffusion, and developed predictive models of information diffusion based on community structure. Chen et al. [43] employed the network to investigate the impact of overlapping community structure on susceptible-infected-susceptible (SIS) epidemic spreading process. Similarly, Xiangwei et al. [48] studied epidemic spreading in weighted scale-free networks with community structure. Recently, Shang et al. [200] classified vertices into overlapping and non-overlapping ones, and investigated in detail how they affect epidemic spreading.

2.2.5 Community Structure in Recommendation Systems

Community detection algorithms and clustering functions constitute a powerful tool in the development of network based recommendation system. Zhuhadar et al. [249] used the community detection method to design a visual recommender system to recommend learning resources to cyberlearners within the same community. Lisboa et al. [138] proposed a method to improve recommendation systems by taking into consideration changes in the behavior of users over time. For that, communities are first detected using a network analysis method and recommendations are made for each community using Naïve Bayes

modeling. Kamahara et al. [109] proposed a recommendation method in which a user can find new interests that are partially similar to the user's taste, where partial similarity is an aspect of the user's preference which is projected by the community in which the user belongs. Musto et al. [154] particularly studied user community behavior in OSN and developed *STaR* to suggest a set of relevant keywords for the resources to be annotated. Fatemi and Tokarchuk [69] proposed novel community based social recommender system, *CBSRS* which utilizes the social data to provide personalized recommendations based on communities constructed from the users' social interaction history with the items in the target domain.

Chapter 3

Analyzing structural properties of temporal networks

This chapter is devoted to out first objective - understanding and predicting structural properties of temporal networks.

3.1 Introduction

Understanding the structural properties of temporal network is of primary importance. To this end the research community was initially inclined towards aggregating the nodes and edges over all time steps and then analyzing the behavior of the aggregated network. This strategy however was found to hide the time ordering of the nodes and the edges which plays a significant role in understanding the true nature of such temporal networks. This led researchers to come up with various growth Recently, new network applications have cropped up where an estimate of the network properties are helpful even though the network structure is itself unavailable. For instance, in order to launch a targeted attack on the network one might not require the full knowledge of the network structure. Instead, an approximate estimate of some of the properties might be useful in finding the order in which the nodes and the edges may be removed.

In this work, we propose a simple strategy to represent a temporal network as time series. Essentially, we consider a temporal network as a set of static snapshots collected at consecutive time intervals and represent each of them in terms of the properties of the network. In specific, we consider eight properties namely number of active nodes, average degree, clustering coefficient, number of active edges, betweenness centrality, closeness centrality, modularity and edge-emergence [210].

We then use the known analytical tools for time series predictions to predict the network properties at a future time instance. Note that the time series framework can be particularly effective as it is impossible to define a unified network evolution/growth model for temporal networks simply because the rules of temporality are varied across systems. Hence the feasible alternative could be to learn the evolution pattern (which we do through time-series analysis) and then predict the later time steps.

Due to various irregularities in the time series, predictions at certain points are erroneous. Therefore we further refine our prediction framework using spectrogram analysis by identifying beforehand the cases where the prediction error is high i.e., unsuitable for prediction. In fact we observe that the accuracy of the framework is enhanced further by 7.96 (for error level $\leq 20\%$) on an average across all datasets if we remove the cases which are deemed unsuitable for prediction by spectrogram analysis.

As an application we also propose a strategy to launch targeted attacks based on our prediction framework and show that this scheme beats the state-of-the-art ranking method used for such attacks. We believe that our framework could be used in designing ranking schemes for nodes in temporal networks at a future time step albeit the network structure at that time step itself is unknown.

We perform our experiments on five different human face-to-face communication networks and observe that the above properties could be segregated based on time domain and frequency domain (spectrogram) characteristics. In general this method allows us to make predictions with very low errors. Importantly, the frequency domain analysis also nicely separates out those properties that can be predicted with low errors from those for which it is not possible.

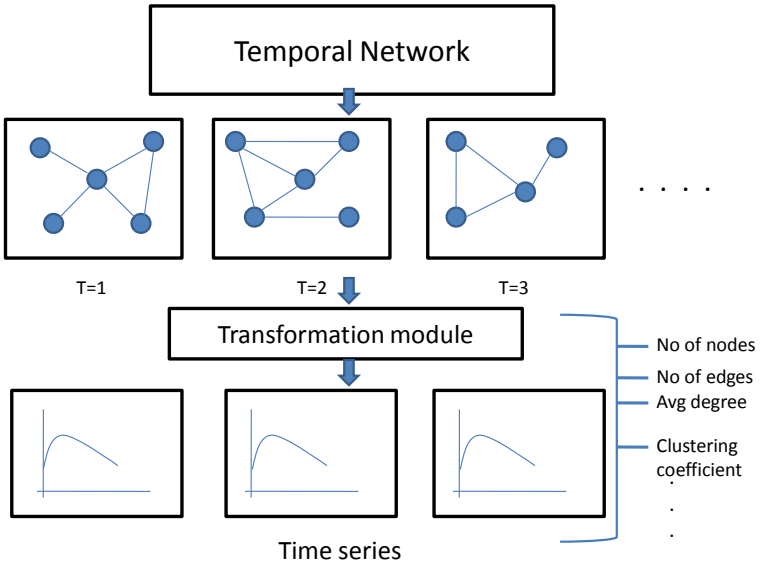


Figure 3.1: Converting Temporal network to Time series

3.2 Mapping Temporal network to time series

We consider a temporal network as a set of static snapshots collected at consecutive time intervals. Each snapshot is then represented in terms of eight (mainly structural) properties of the underlying graph. Consequently we obtain a set of points ordered in time or equivalently a discrete time series (see figure 6.1). The eight properties we use to represent the temporal network as a time series are:

- (1). **Number of active nodes**: This is the count of the number of nodes in the system at a given time step. We consider active nodes to be those which have non-zero degree in a time step. We represent the number of active nodes in the system at time step t by N_t. In a similar way we define (2). **number of active edges** and (3). **average degree** and represent the values of these properties at time t by E_t and Avg_deg_t respectively.
- (4). **Edge emergence**: Edge emergence [210] is a measure that estimates structural similarity. For measuring the edge emergence at time t we consider each edge of the network at time t and for each of its two endpoints we calculate the number

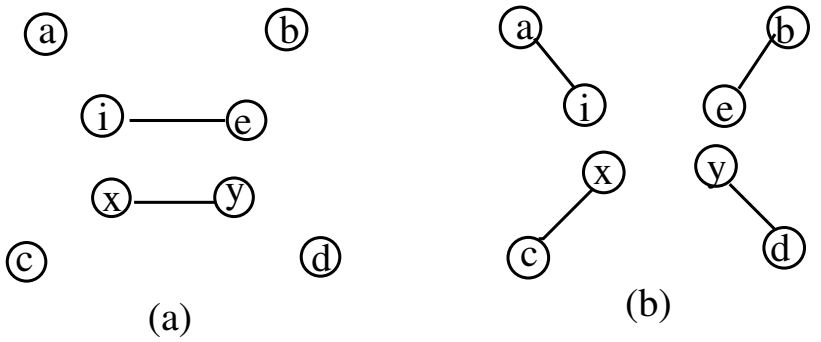


Figure 3.2: (a) and (b) denote the status of the network at time t and $t+1$ respectively. For the edge (i, e) in t the corresponding edges emanating from i and e are (i, a) and (e, b). For the edge (x, y) they are (x, c) and (y, d). So the $Edge\_emer_t = \frac{2+2}{2} = \frac{4}{2}$

of edges emerging in the next time step $t+1$. We represent edge-emergence at time t by $Edge\_emg_t$. If E_t denotes the set of edges present in the network at time t and A_{t+1} denotes the set of edges at time $t+1$ which are adjacent to E_t then $Edge\_emg_t = \frac{|A_{t+1}|}{|E_t|}$. Figure 6.14 shows how we calculate this measure for a temporal network at any time instance.

- (5). **Modularity**: We decompose each snapshot into communities using the technique specified in [28] and measure the goodness of this division using modularity [160]. We represent modularity of the system at a given time step t by Mod_t.

We also consider (6). **betweenness centrality**, (7). **closeness centrality** and (8). **clustering coefficient** of the graph (values computed for each node and then summed over all nodes) and their values at time step t are represented by Bet_cen_t, $Clos\_cen_t$ and $Clus\_coeff_t$ respectively.

3.3 Description of the dataset

We perform our experiments on five human face-to-face network datasets: INFOCOM 2006 dataset [199], SIGCOMM 2009 dataset [177][1], High school datasets (2011,

[1] http://crawdad.org/

2012) [75] and Hospital dataset [223][2].

- **INFOCOM 2006:** This is a human face-to-face communication network and was collected at the IEEE INFOCOM 2006 conference at Barcelona. 78 researchers and students participated in the experiment. They were equipped with imotes and apart from them 20 stationary imotes were deployed as location anchors. The stationary imotes had more powerful battery and had a radio range of about 100 meters. The dynamic imotes had a radio range of 30 meters. If two imotes came in each others' range and stayed for at least 20 seconds then an edge was recorded between the two imotes. The edges were recorded at every 20 seconds. Therefore, this is the lowest resolution at which the experiments can be potentially conducted. However, we observe that at this resolution the network is extremely sparse which makes it difficult to conduct meaningful data comparison and prediction. We observe experimentally that the lowest interval that allows for appropriate comparison and prediction is 5 minutes and therefore we set this value as our resolution for all further analysis.

- **SIGCOMM 2009:** This is also a human face-to-face communication network and was collected at the SIGCOMM 2009 conference at Barcelona, Spain. The dataset contains data collected by an opportunistic mobile social application, MobiClique. The application was used by 76 persons during SIGCOMM 2009 conference in Barcelona, Spain. The trace records all the nearby Bluetooth devices reported by the periodic Bluetooth device discoveries. Each device performed a periodic Bluetooth device discovery every 120+/-10.24 seconds for nearby Bluetooth devices. A link was added with a device on discovering it. We remove the contacts with external Bluetooth devices and a network snapshot is an aggregate of data obtained for 5 minutes.

- **High school datasets:** These are two datasets containing the temporal network of contacts between students in a high school in Marseilles taken during December 2011 and November 2012 respectively. Contacts were recorded at intervals of 20 seconds. We consider a network snapshot as an aggregate of data obtained for 5 minutes.

- **Hospital dataset:** This dataset consists of the temporal network of contacts between patients and health care workers in a hospital ward in Lyon, france. Data was col-

[2]http://www.sociopatterns.org/

Dataset	# unique nodes	# unique edges	edge type	Time span of the dataset	Time steps for prediction
INFOCOM 2006	98	4414	undirected	1120	200 - 800
SIGCOMM 2009	76	2082	do	1068	300 - 900
Highschool 2011	126	5758	do	1215	200 - 900
Highschool 2012	180	8384	do	1512	200 - 1000
Hospital	75	5704	do	1158	100 - 900

Table 3.1: Properties of the dataset used.

lected at every 20 second intervals. Due to sparseness of the network of 20 seconds, we consider each network snapshot as an aggregated network of 5 minutes.

In table 1 we provide the details of the datasets.

3.4 Analysis of time series

We now present the plots of the time series and analyze their properties based on both time domain and frequency domain characteristics.

3.4.1 Time domain characteristics

For the time domain analysis of the properties we look into the time series plots for the datasets represented in figures 3.3(A), (C), (E) and 3.4(A), (C). From the time series plots we observe the presence of periodicity in almost all the datasets. A stretch of high values is followed by a stretch of low values and so on. However, they are of varying lengths. This indicates the presence of correlation in case of human face-to-face communication network. We quantify this structural correlation later in this paper. We also check whether these time series are stationary. On performing KPSS (KwiatkowskiâĂŞPhillip-sâĂŞSchmidtâĂŞShin) [121] and ADF (Augmented Dickey Fuller) test [57] on the data we conclude that the data is non-stationary. Overall, the presence of correlation in case of

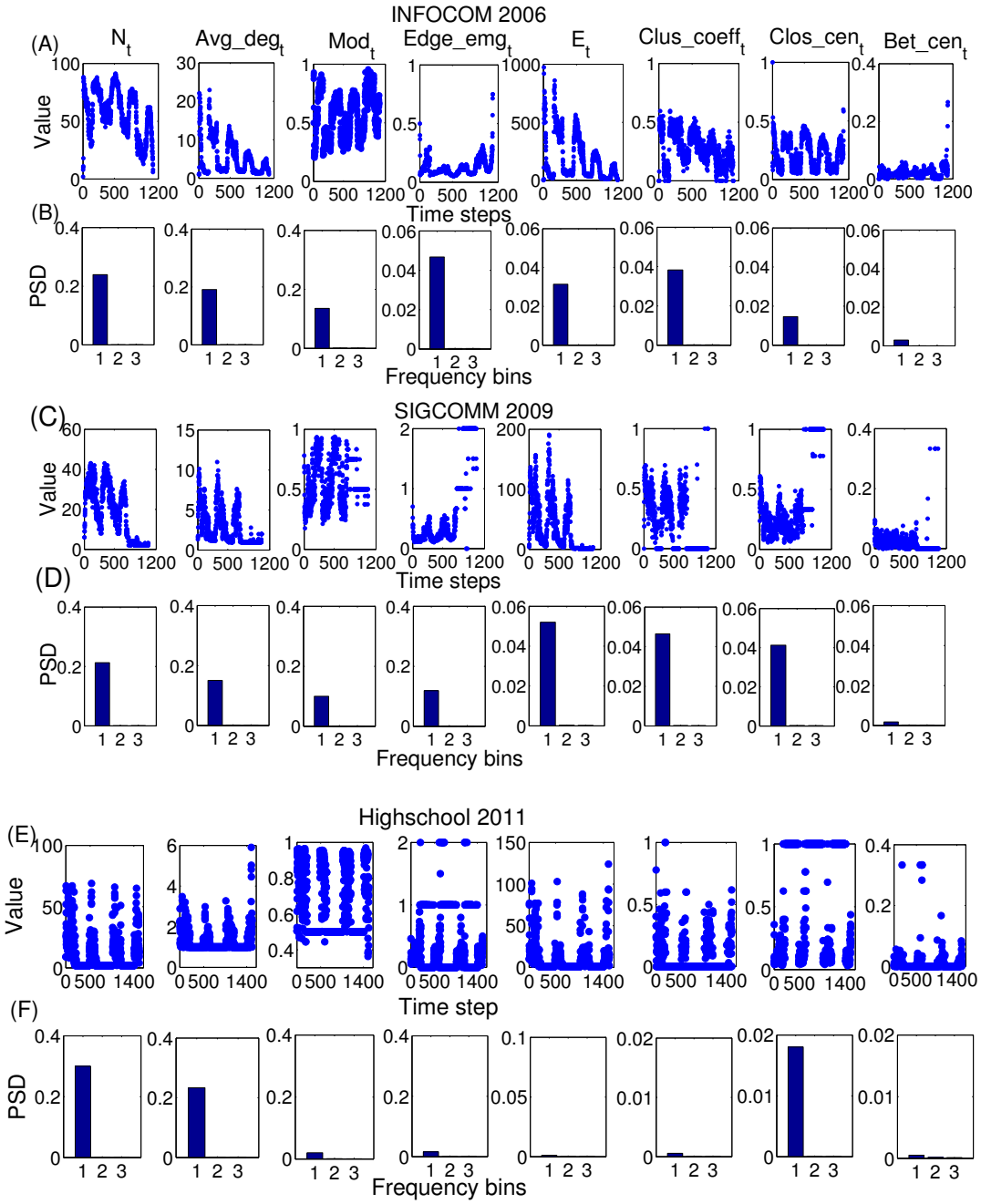


Figure 3.3: (A), (C) and (E) represent the time series plots for INFOCOM 2006, SIGCOMM 2009 and High-school 2011 respectively. (B), (D) and (F) represent the power spectral density (PSD) corresponding to the frequency bins for INFOCOM 2006, SIGCOMM 2009 and High-school 2011 dataset respectively. Bins 1, 2 and 3 corresponds to frequencies $<$5, 5-15 and $>$15(Hz) respectively.

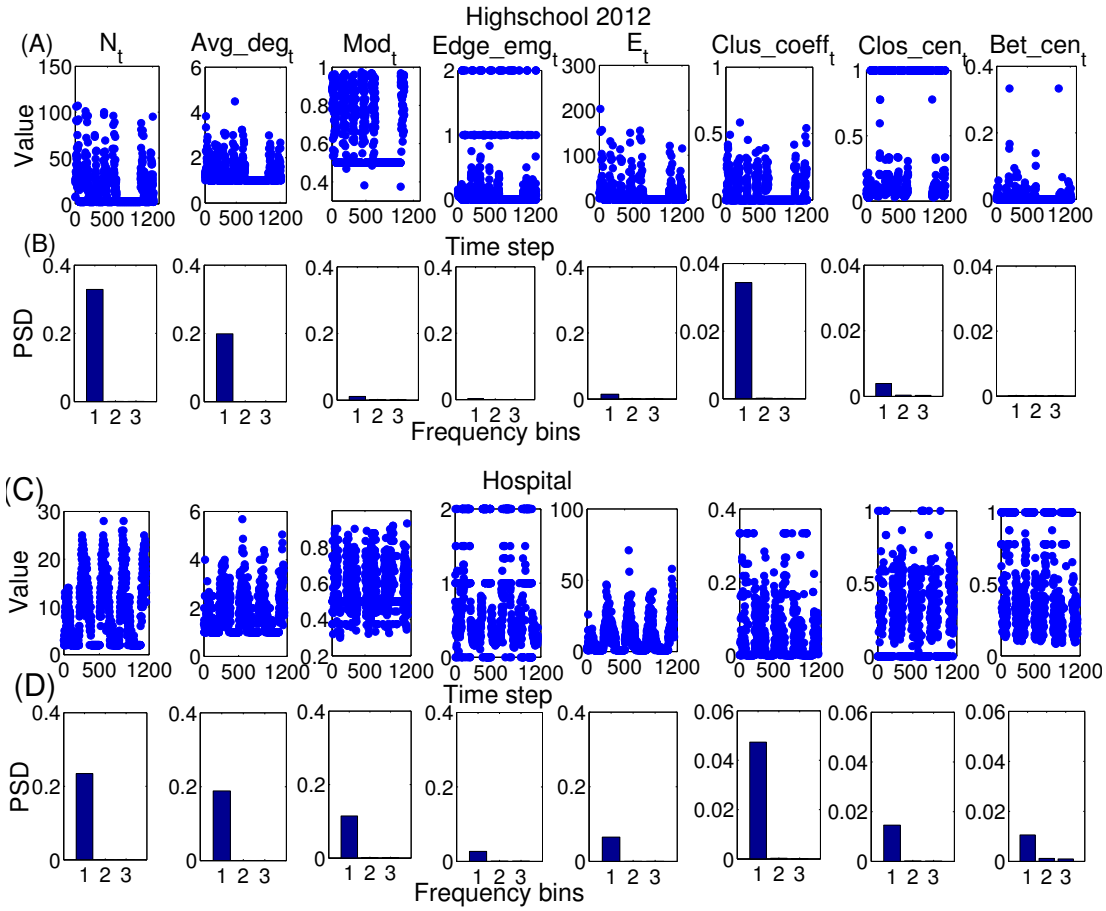


Figure 3.4: (A) and (C) represent the time series plots for High-school 2012, Hospital respectively. (B) and (D) represent the power spectral density (PSD) corresponding to the frequency bins for High-school 2012 and Hospital dataset respectively. Bins 1, 2 and 3 corresponds to frequencies $<$5, 5-15 and $>$15(Hz) respectively.

human face-to-face network indicates that it is a stochastic process with memory i.e., the contacts a node makes in the current time step is influenced by its contact history.

3.4.2 Frequency domain analysis

We perform the frequency domain analysis of the time series extracted from the temporal network by conducting a spectrogram analysis of the data. Spectrogram analysis is a short time fourier transform where we divide the whole time series into several equal sized windows and apply discrete fourier transform on this widowed data. The main advantages

of using spectrogram analysis are (a). we do not lose the time information, (b). we are able to obtain a view of the local frequency spectrum. Also note that the spectrogram analysis allows us to identify as well as quantify the fluctuations in the data which is difficult to identify from the corresponding time series. A high concentration of low frequency components would indicate lower fluctuations in the data; in contrast no such concentration of low frequency components would indicate higher fluctuations and irregularities in the data.

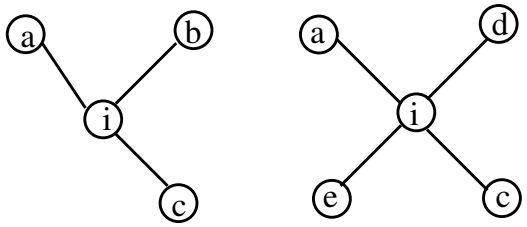
We construct the spectrogram and segregate the power spectral density (PSD measured in Watts/Hz) based on the frequency into three bins. In bin 1 we calculate the mean PSD corresponding to the frequencies < 5 Hz, in bin 2 we calculate the PSD corresponding to frequencies between 5 and 15 Hz and bin 3 consists of the mean PSD value corresponding to frequencies > 15 Hz. We call them LPSD, MPSD and HPSD respectively. So a higher value of mean PSD corresponding to bin 1 (LPSD) would indicate lower fluctuations in data. In figures 3.3(B), (D), (E) and 3.4(B), (D) we plot the PSD corresponding to the three bins across all the properties for all the datasets. We observe that the lower frequencies dominate to a higher extent in case of the properties like number of active nodes, number of active edges, modularity but to a much lower extent in case of betweenness centrality, closeness centrality and clustering coefficient. In fact the prediction accuracy of a property can be enhanced through spectrogram analysis.

3.5 Prediction framework

In this section, we employ the time series to forecast the different structural properties of the temporal networks. Elementary models of time series forecasting could be categorized into Auto-regressive(AR) and Moving average(MA) models [41]. In case of an auto-regressive model of order p, AR(p), the value of the time series at time step t is given as -

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + e_t + c$$

where α_is are parameters, e_t is the white noise error term and c is a constant. Similarly, in case of Moving average model of order q, MA(q), the value of the time series at time step t is given as -



(a) (b)

Figure 3.5: (a) and (b) denote the status of node i at time t and $t + k$ respectively. $NBR(i)_t = \{a, b, c\}$ and $NBR(i)_{t+k} = \{a, d, c, e\}$ $Correlation(i)_k = \frac{NBR(i)_t \bigcap NBR(i)_{t+k}}{NBR(i)_t \bigcup NBR(i)_{t+k}} = \frac{2}{5}$ where $NBR(i)_t \rightarrow$ the set of neighbors of i at time t

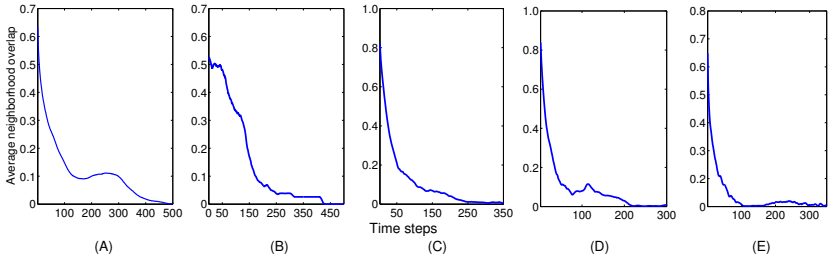


Figure 3.6: The average neighborhood-overlap value at different lags for (A)INFOCOM 2006, (B)SIGCOMM 2009, (C)Highschool 2012, (D)Highschool 2011 and (E)Hospital datasets.

$$y_t = \beta_1 e_{t-1} + \cdots + \beta_q e_{t-q} + \mu + e_t + c$$

where β_is are parameters, $e_t, e_{t-1}, ...$ are white noise error terms and μ is the expectation of y_t. These two models could be combined into Auto-regressive-moving-average (ARMA(p,q)) [41] where the value of the time series at time step t is given as

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \beta_1 e_{t-1} + \cdots + \beta_q e_{t-q} + e_t + c$$

However, in our case the time series show evidences of non-stationarity and short term dependencies and these models are insufficient and hence we use ARIMA model [32] for forecasting. The initial differncing step in ARIMA model is used to reduce the non-stationarity. On fitting an ARIMA(p,d,q) model to a time series we obtain an auto-regressive equation of the form-

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \beta_1 e_{t-1} + \cdots + \beta_q e_{t-q} + c$$

Hence we can take a time series corresponding to a network property and fit an ARIMA model to it. Thus, we obtain an auto-regressive equation for that series which can be used in forecasting. In order to predict a value at a future time point, we divide the data in smaller parts and perform our predictions on these smaller stretches. In the next subsection we discuss how we perform this division.

3.5.1 Selecting a window

In order to identify the right length of a stretch (i.e., a window size) we need to identify how the network at any time point is influenced by the network at the previous time points. The basic idea is that the time points to which this influence extends should all get included into a single window. To quantify this influence we define a new metric called **neighborhood-overlap** which measures the structural correlation between network snapshots at two time steps. We define the difference between these two time steps as the lag. To measure the neighborhood-overlap of the network snapshots at time t and $t + k$, we calculate for each

active node at time t the overlap in its neighborhood between two time points. To measure this overlap we use the standard Jaccard similarity as has been pointed out in [212]. Note that this is one of the most standard and interpretable ways to measure structural similarity as has been identified in the literature with applications ranging from measuring keyword similarity [167] to similarity search in locality-sensitive-hashing (LSH) [21]. It has also been extensively used in link prediction [141, 142] as well as community detection [171]. Figure 3.5 shows how we formulate this measure using the Jaccard similarity index. We represent the neighborhood overlap at lag k as the mean value across all the active nodes in time step t. To measure the extent of similarity we measure neighborhood-overlap for each snapshot at different lags and take the average of them. This essentially shows, given a time specific snapshot how the similarity changes as we increase the lag. Figures 3.6(A) - (E) show how this similarity changes with time as we increase lag for the five different datasets. As we increase the lag the similarity decreases almost exponentially and hence considering snapshots at larger lag where the similarity value is very low could introduce error in learning the auto-regressive equation. Also for a higher similarity value the corresponding lag would increasingly introduce more error in the fit due to lesser number of data points on which the ARIMA model gets trained to learn the fit function (see figure 3.7). In fact we observed that the error in prediction increases if we consider a lag too small (high similarity value) or too large (low similarity value) (see figure 3.7). Hence we consider the similarity value of 0.2 as the threshold for calculating the lag. For our prediction framework the corresponding value of the lag acts as the window for fitting the ARIMA model.

Let the size of the window be w and we want to predict the value of the time series at time t. To our aim we consider the time series of the previous w time steps consisting of the values between time steps $t-1-w$ to $t-1$ and fit the ARIMA model to it and obtain its value at time step t. We repeat this procedure for forecasting at every value of t. Thus, the time step t is the test point and the series of points $t-w-1$ to $t-1$ form the training set. One can imagine this process as a sliding window of size w which is used for learning the auto-regressive equation and the point that falls immediately outside the window is the unknown that is to be predicted.

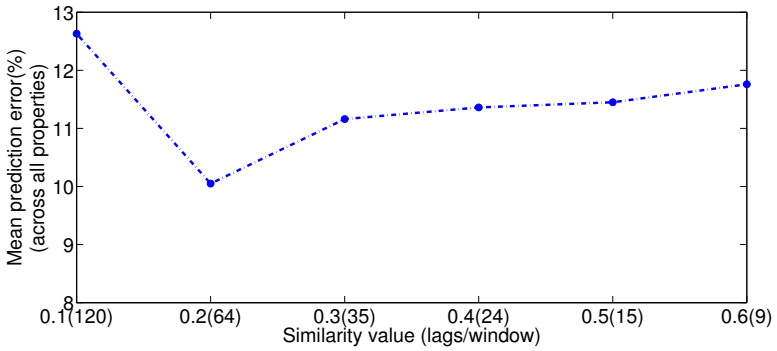


Figure 3.7: Mean prediction error (%) across different properties for INFOCOM 2006 dataset for different similarity values. The lags corresponding to the similarity value are also provided.

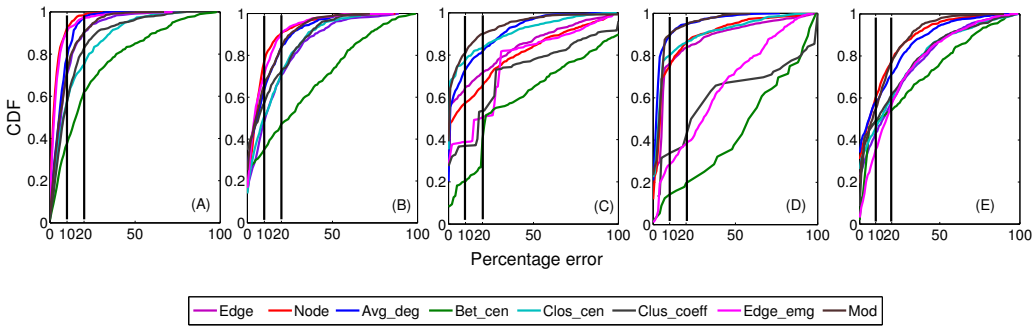


Figure 3.8: The percentage error distribution of all the properties (time series) for (A) INFOCOM 2006 dataset, (B) SIGCOMM 2009 dataset, (C) High-school 2011. (D) High-school 2012 and (E) Hospital. X-axis represents percentage error and Y-axis represents probability.

3.6 Prediction results

In this section, we provide detailed results of the our prediction framework on the datasets discussed earlier. To determine the accuracy of our prediction strategy we use the cross validation technique. For each time step in this range we use our framework to obtain a prediction at that time step. Since we already know the original value, we can obtain a percentage error for the prediction. Let $predict_t$ represent the prediction value at time t and $original_t$ represent the original value. We obtain percentage error ($error_t$) using the formula:

$$error_t = \frac{|original_t - predict_t|}{original_t} * 100$$

First we try to find the suitable window for predicting the value of a time series at a time step. For this we refer to figure 3.6 where we quantify structural correlation and show how the similarity value decreases with increasing lag. We observe that the value of the structural correlation decreases as we increase the lag. For INFOCOM 2006 dataset (figure 3.6(A)) the correlation drops to less than 0.2 at lag around 70. Therefore we select a window of size 64. We could have selected any other value between 60 and 70, but we select 64 as it is in the power of 2 and it helps in the spectrogram analysis. Similarly we find the suitable window size to be around 128, 64, 64, 32 (closest power of 2) for the SIGCOMM 2009, Highschool 2011, Highschool 2012 and Hospital datasets respectively (refer to figure 3.6).

For the INFOCOM 2006 dataset we consider the time steps 200-800. Note that selection of these is just representative and one is free to take any time step given there is a window of appropriate length available. For SIGCOMM 2009, High School 2012, High School 2011 and Hospital datasets we consider our test time steps to be 300-900, 200-1000, 200-1000 and 100-900 respectively (refer to table 1).

To check how efficient our predictions are we plot the cumulative probability distribution of percentage error for all the datasets in figure 3.8. In table 3.2, we compare the prediction results across different datasets and different metrics for cases where the prediction error $\leq 20\%$. Note that this error level is representative and ideally a table can be recovered for each such error level from figure 3.8. We make the following observations from the results:

	Prediction error $\leq 20\%$				
Datasets	INFOCOM 2006	SIGCOMM 2009	Highschool 2012	Highschool 2011	Hospital
# Active nodes	**0.984**, (0.988)	**0.907**, (0.91)	0.68, (0.765)	**0.861**, (0.882)	0.782, (*0.859*)
Average degree	**0.975**, (0.968)	**0.84**, (0.834)	**0.816**, (0.81)	**0.91**, (0.908)	0.714, (0.724)
Modularity	**0.905**, (0.921)	**0.838**, (0.85)	**0.90**, (0.91)	**0.92**, (0.917)	0.78, (0.812)
Edge emergence	**0.971**, (0.983)	**0.906**, (0.91)	0.56, (*0.71*)	0.42, (*0.512*)	0.57, (*0.652*)
# Active edges	**0.901**, (0.91)	0.71, (*0.81*)	0.72, (*0.78*)	**0.836**, (0.86)	0.734, (*0.796*)
Clustering coefficient	**0.829**, (0.858)	0.725, (0.75)	0.54, (*0.623*)	0.5, (*0.682*)	0.71, (*0.751*)
Closeness centrality	0.751, (*0.887*)	0.71, (*0.83*)	**0.83**, (0.843)	**0.821**, (0.853)	0.74, (*0.786*)
Betweenness centrality	0.621, (*0.818*)	0.472, (*0.61*)	0.51, (0.63)	0.22, (*0.418*)	0.542, (*0.689*)
Average	0.867, (*0.916*)	0.763, (*0.813*)	0.694, (*0.768*)	0.686, (*0.754*)	0.69, (*0.74*)

Table 3.2: Network property and the fraction of predictions with percentage error $\leq 20\%$ without (with) spectrogram analysis. The cases where more than 80% of the points have prediction error $\leq 20\%$ have been highlighted in bold font and the cases where on using spectrogram analysis the improvement is more than 5% have been underlined.

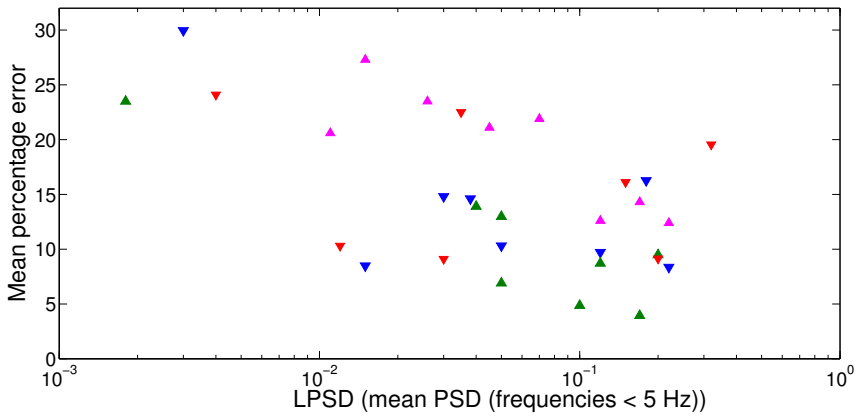


Figure 3.9: (A) LPSD versus mean percentage error for all the properties across all the datasets.

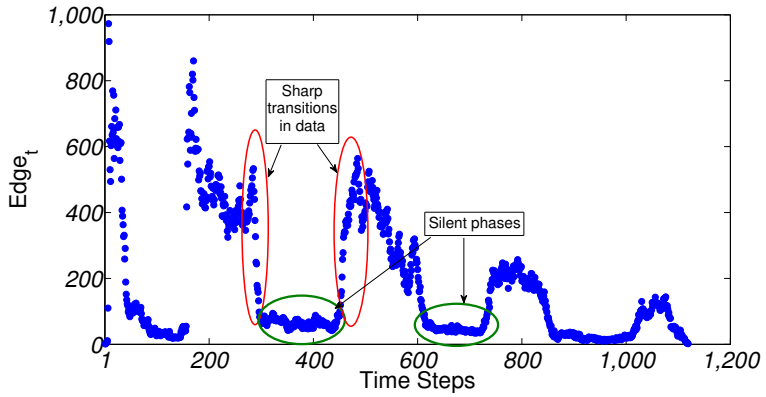


Figure 3.10: The time series plot for number of active edges. The red and the green ellipses identify two transition and two silent phases respectively

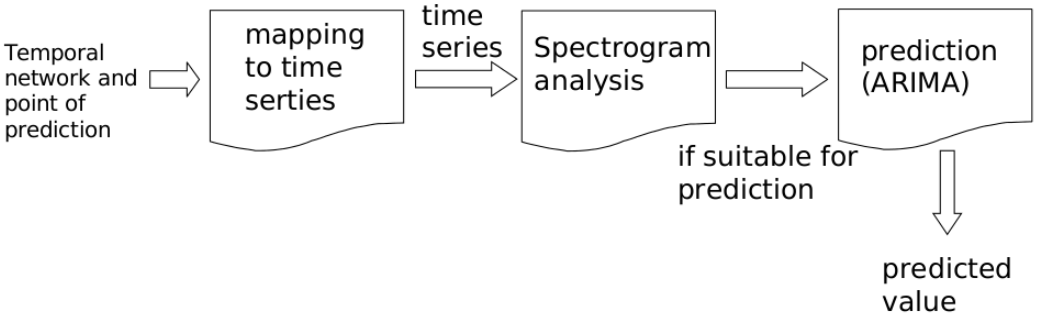


Figure 3.11: The prediction framework.

- Our framework is able to predict the values for active nodes, average degree and modularity with high accuracy across all datasets.
- For active edges, edge emergence, clustering coefficient and closeness centrality our framework is able to predict the values with moderate accuracy although the prediction accuracy for these properties is reasonably high for some datasets (INFOCOM 2006, SIGCOMM 2009).
- The prediction accuracy is poor across all datasets for betweenness centrality and in some cases for clustering coefficient and closeness centrality.

An important observation is that the spectrogram analysis (introduced in section 3.4) is able to distinguish between these properties based on their predictability. On ranking the properties based on the PSD value at bin 1 (refer to figures 3.3(B), (D), (F) and 3.4(B), (D)), we observe that the higher ranked properties are the ones for which the prediction error is low while the lower ranked ones have higher prediction error. Following this observation we further plot the mean percentage error for all the properties across all the datasets versus LPSD in figure 3.9. The plot clearly shows that the higher the value of LPSD, lower is the mean percentage of error and vice versa.

On further investigating into the cases where the prediction error is high, we observed that these points are mostly located either in places where a sharp transition occurred or in silent phases where there was limited interaction among the nodes. Figure 3.10 identifies some of the transition and silent phases in the time series of number of active edges in INFOCOM 2006 dataset. Similar phases are also present in the other datasets as well.

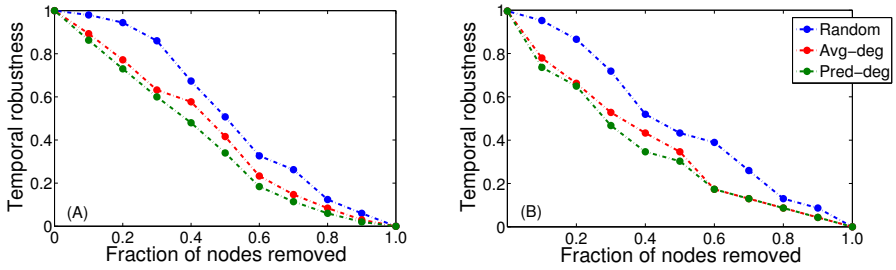


Figure 3.12: Temporal robustness as a function of the fraction of removed nodes for (a) INFOCOM 2006 and (b) SIGCOMM 2009 datasets.

3.6.1 Enhancing the prediction scheme through spectrogram

Since spectrogram analysis of the time series could determine the predictability of the corresponding property, an immediate extension would be to check whether it could be leveraged to identify beforehand the cases where the prediction error is high (unsuitable for prediction). To that aim we extend the spectrogram analysis to the single point case whereby while predicting a property at a give time step, we find that the spectrogram of the window

(w) and use the LPSD value as an indicator for potential prediction accuracy. The cases identified by spectrogram analysis to be unsuitable for prediction can then be filtered out to improve the overall accuracy of the prediction framework. A schematic diagram of this enhanced prediction framework is provided in figure 6.8. We now consider all the datasets and the corresponding time steps for prediction (which we considered earlier in this section, refer to table 1) and instead of directly using our prediction framework we perform spectrogram analysis (single point method) on these points to separate out those which are unsuitable for prediction. We predict only the points which the spectrogram analysis identified as suitable for prediction. In table 3.2 we compare the fraction of predictions with error $\leq 20\%$ between both the cases where we do not use spectrogram and where we use spectrogram. We observe that the fraction of prediction with error $\leq 20\%$ is enhanced for all the properties albeit only marginally in some cases where the accuracy was already high. More importantly for (ill-predicted) properties like betweenness centrality, closeness centrality, the prediction accuracy increases substantially. Note that prediction error 20% is again representative and similar results could be obtained for other values of prediction error as well.

3.7 An attack strategy using prediction framework

In this section we show how our prediction can be used in order to launch targeted attack on temporal networks. The strategy proposed is a modification over the average node degree attack presented in [218]. In case of average node degree attack the temporal degree[3] [218] of the nodes are calculated and the node with highest temporal degree is removed in the subsequent steps (i.e., "the node is attacked"). We observe that for every node its degree over a given time interval forms a time series. Using our prediction framework we calculate the degree of the node at a future time step based on the previous w time steps (window size for the corresponding dataset, refer to section 6.10) and remove a node with the highest degree as predicted by our proposed framework. We compare our strategy (Pred-deg) with average node degree based attack (Avg-deg) and the random case (nodes are selected at random and removed). The effectiveness of an attack strategy is measured using temporal robustness [218] which is estimated by the relative change

[3]Given a time interval $[t_1, t_n]$ temporal degree of node i is the average degree of i over the time interval

in efficiency [218] after a structural damage D. Temporal efficiency of a network G in a given time interval $[t_1, t_n]$, $E_G(t_1, t_2)$ is defined as the averaged sum of the inverse temporal distances over all pairs of nodes in that time interval.

$$E_G(t_1, t_2) = \frac{1}{N(N-1)} \Sigma_{i,j:i \neq j} \frac{1}{d_{ij}(t_1, t_2)}$$

Here N is the number of nodes in the network and $d_{ij}(t_1, t_2)$ is the temporal distance which is the smallest temporal length paths among all the temporal paths between i and j in the time interval $[t_1, t_2]$. Hence, temporal robustness is defined as $R_G(D) = \frac{E_{GD}}{E_G}$. In figure 3.12 we plot temporal robustness as a function of the fraction of nodes (P) removed for (a) INFOCOM 2006 and (b) SIGCOMM 2009 datasets. We observe that our strategy does better than both the random and average node degree based strategy.

3.8 Summary of this chapter

Our contributions in this chapter can be summarized as below:

- we provide a general framework to map temporal network of human contacts consisting of a series of graphlets equispaced in time into time series and provide a detailed time domain and frequency domain analysis.
- we re-establish the presence of structural correlation in a temporal network of human face-to-face contact using a new metric which we call neighborhood-overlap.
- we further quantify the extent of this correlation using neighborhood-overlap and use to identify the correct window used in our prediction framework.
- we also provide an approach for predicting the properties of future network instances using time series as a proxy and show that even though the precise network structure is not known at time step, one can estimate its properties.
- finally we provide a frequency domain analysis of temporal network and show how it can be useful in enhancing the prediction accuracy.

- as an application we show how our framework can be used in devising better strategies for targeted network attacks.

In its current state our framework can predict the values of the network properties at a future time step but is unable to offer the exact network structure at that time step. But our framework can have genuine contributions toward link prediction in temporal networks. Since we show that structural correlation exists in these networks and we can also predict the network properties at these time steps as well, we can re-frame the link prediction problem as a network at a time step which is obtained from the network at a previous time step with minimal changes made depending on the values of the properties.

Chapter 4

Sampling temporal graphs

In the last chapter we developed a framework that could predict structural properties of a temporal network albeit we could not make any prediction regarding the structure of the network. It is in fact difficult to predict the structure of the network at a future time instant. We in this chapter look into a related problem - given a streaming graph, we obtain a sample of the graph which is most representative of its underlying structure. In specific we obtain a sample that is able to preserve the underlying community structure.

4.1 Introduction

Nowadays graphs are so large that their analysis in entirety can be intractable and impractical. How then one should proceed in analyzing and mining these graphs? Traditional approaches include designing more efficient algorithms or leveraging computing power through parallelization or distributed computing. Unfortunately, these existing methods are not always easily available as an option. Another approach that has received recent attention is the technique of *sampling* [134].

Although data sampling is exhaustively studied in statistics [61, 93], sampling from graphs has got only limited attention [80, 146, 184, 190]. Moreover, when it comes to the case of sampling from *streaming graphs* (where edges arrive in discrete time intervals) there

is hardly any work beside [3, 5]. Existing graph sampling methods are mostly designed for *static graphs* and aim at preserving general structural properties (such as degree distribution, clustering coefficient etc.) of the original graph in the sample. However we posit that it is impossible for any sampling method to produce a universal representation that can preserve *all* sorts of graph properties of the original graph; rather graph sampling should be *application specific*. For instance, sampling method designed for information diffusion should preserve the hubs (high-degree nodes) in the sample; whereas sampling for outbreak detection (such as disease outbreak) should preserve the nodes with high local clustering coefficient.

In this work, we propose a novel sampling algorithm that preserves the original *community structure*[1]. A community in a graph is a cluster of nodes more densely connected among themselves than to others [228]. Identifying community structure is important, as they represent the *mesoscopic view* of the graph and often correspond to real social groups, functional groups, or demographic similarity etc. [79]. The ability to easily construct a sample consisting of members from diverse communities has several important applications. For instance, in marketing, surveys often seek to construct stratified samples that collectively represent the diversity of the population [117]. Many popular community detection algorithms considered to be accurate are also computationally expensive [55, 71]. Representative graph sampling, then, provides a potential solution for inferring and approximating global, latent properties such as these in large graphs. By sampling a representative subgraph, analysis can be performed on the sample instead of the larger graph. Results could, then, be generalized to the larger population, which, in this case, is the original graph. However, if one still intends to run an accurate and computationally-expensive community detection algorithm on large graphs and is confused which one algorithm to choose, she can run several potential algorithms on the sample graph and choose one that performs best. Then that algorithm can be used to detect communities from the original large graph.

Our contributions: We propose `ComPAS`, a novel sampling algorithm on streaming graph (most realistic graph representation [3,5]) that is capable of representing and inferring community structure in the original graph. `ComPAS` systematically interweaves graph sampling and community detection so that one gets benefits from the other to produce a more repre-

[1] In this work, we consider disjoint community structure.

sentative sample. In particular, our contributions are three-fold:

- To our knowledge ComPAS is the first *community-preserving* sampling method for *streaming graphs*. Along with the sample, ComPAS also produces the community structure of the sample.
- Empirical evidences on synthetic graph and different real-world graph demonstrate that the sample generated by ComPAS is not only the most representative to preserve the community structure, but also is quite competitive in reproducing the other general graph properties. The average performance of our algorithm reaches 85.5% of the most informed algorithm (GA [217]) on static graphs.
- We show additional benefits of ComPAS through two applications – (i) selection of right community detection algorithm for large graphs, and (ii) selection of (limited) training set for online learning. We obtain a performance that is within 95.6% and 90.5% of the most informed algorithm (i.e., GA) available for static graphs for first and second applications respectively.

4.2 Problem Definition

We consider a graph stream S represented by a set of edges $e_1, e_2, ...$ with each edge e_i arriving at i^{th} (discrete) time step. A graph G at time t is the aggregate of all the edges arriving till time t. V represents the set of unique nodes present in the graph G. The community structure of the graph G is represented by C. We consider G to be both unweighted and undirected (see Table 4.1 for the notations).

Definition 1. *Given a streaming graph G and sample size n in terms of the number of nodes, our objective is to obtain a sample graph G_s such that C, the underlying community structure of G is highly preserved in G_s (i.e., $C \sim C_s$ where C_s is the community structure of G_s).*

Algorithm 1: ComPAS: A **Com**munity **P**reserving Sampling **A**lgorithm for **S**treaming Graph

Data: S: Graph stream, n: Sample size, α: Initial fraction of nodes inserted, β: Edge density threshold, n_d: size of the buffer, $Algo$: a community detection algorithm

Result: Sampled Subgraph $G_s(V_s, E_s)$, C_s

Initialize G_s: $V_s = \phi$, $E_s = \phi$
Crate an empty buffer $\mathcal{H}$ of size n_d
Initialize buffer $\mathcal{H}$: $\mathcal{H}_c = \phi$, $\mathcal{H}_p = \phi$
$flag = 1, t = 0$
for e_t *in the graph stream* S **do**
    $e_t = \{u, v\}$
    if $\frac{|V_s|}{n} < \alpha \wedge e_t \notin E_t$ **then**
        $V_s = V_s \cup u \cup v$
        $E_s = E_s \cup e_t$
        Continue;
    else if *flag==1* **then**
        Run $Algo$ on G_s and detect community structure C_s
        $flag$=0
    else if $u, v \in V_s$ **then**
        $V_s, E_s, C_s = BothinSample(u, v, e_t, V_s, E_s, C_s)$
    else if $u \in V_s \wedge v \notin V_s \wedge v \notin \mathcal{H}$ **then**
        $V_s, E_s, C_s, \mathcal{H} = OneinSampleOneNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$
    else if $u \in V_s \wedge v \notin V_s \wedge v \in \mathcal{H}$ **then**
        $V_s, E_s, C_s, \mathcal{H} = OneinSampleOneinBuffer(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$
    else if $u \notin V_s \wedge u \in \mathcal{H} \wedge v \notin V_s \wedge v \notin \mathcal{H}$ **then**
        $V_s, E_s, C_s, \mathcal{H} = OneinBufferOneNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$
    else if $u, v \notin V_s \wedge u, v \notin \mathcal{H}$ **then**
        $V_s, E_s, C_s, \mathcal{H} = BothNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$
    else if $u, v \notin V_s \wedge u, v \in \mathcal{H}$ **then**
        $\mathcal{H} = BothinBuffer(u, v, \mathcal{H})$
    $t = t + 1$
return G_s, C_s

Table 4.1: Important notations used in this paper.

Notation	Description
S	Graph stream $\{e_1, e_2, \dots\}$
n	Required sample size (in terms of nodes)
G_s	$G_s = (V_s, E_s)$, final graph sample
C_s	Community structure of G_s
α	Initial fraction of nodes inserted
$Algo$	Algorithm used to detect initial community structure
$N(x)$	Neighbor of x
$P(x)$	Parent of x
$\mathcal{H}$	Buffer consisting of $\mathcal{H}_c$ and $\mathcal{H}_p$
$\mathcal{H}_c$	Dictionary tracking number of times each node is encountered
$\mathcal{H}_p$	Dictionary storing the recent parent of each node
n_d	Size of $\mathcal{H}$
D_s	Sum of degree of all nodes inside set s
$C(v)$	Community of node v

4.3 Proposed Algorithm: COMPAS

Here we propose `ComPAS`, a **Com**munity **P**reserving sampling **A**lgorithm for **S**treaming graphs. `ComPAS` aims at sampling a streaming graph in such a way that its underlying community structure is preserved in the sample (a pseudo-code and a toy example are presented in Algorithm 1 and Figure 4.1 respectively). To start with, `ComPAS` keeps on adding streaming edges (nodes) into the sample G_s as long as a certain fraction of nodes α is inserted (lines 1-1). This in turn provides an initial knowledge about the graph structure. Once the threshold is reached, a pre-selected community detection algorithm $Algo$ is run on G_s to detect the initial community structure (line 1). After that, it interweaves both graph sampling and community detection in such a way that each task gets benefits from the other. Once an edge e_t is taken from the stream, `ComPAS` judiciously inserts e_t into G_s with the help of a buffer $\mathcal{H}$ which is composed of $\mathcal{H}_c$ and $\mathcal{H}_p$. $\mathcal{H}_c$ counts the "number of hits" of a node (i.e., number of times a node is encountered till that time)[2], and $\mathcal{H}_p$ keeps track of the current parent of a node (i.e., node with which it arrived last). A streaming

[2]In streaming graph, an edge might appear multiple times.

edge $e_t = \{u, v\}$ is first inserted into $\mathcal{H}$, and depending upon the current position of u and v (whether in the buffer or in the sample), their counts in $\mathcal{H}_c$, their current parents in $\mathcal{H}_p$, and the current community structure C_s of G_s, a decision that which node/edge is to be inserted into G_s is taken. One of the six different submodules is invoked to systematically handle this decision. Throughout the iterations, `ComPAS` maximizes *modularity* [158], a well-studied objective function for community detection defined below:

$$Q(G(V, E), C) = \sum_{c \in C} (\frac{m_c}{M} - \frac{D_c^2}{4M^2}) \tag{4.1}$$

where C is the community structure of G, m_c is the total number of edges inside c, D_c is the sum of degree of all the nodes inside a community $c \in C$, and $M = |E|$ is the total number of edges G. In this section, we individually present each submodule of `ComPAS` in details.

Once the initial threshold α is reached, a community finding algorithm $Algo$ is used to detect the initial community structure of G_s (denoted as C_s) (line 12). Then in every occurrence of a new edge $e_t = \{u, v\}$, it is not allowed to enter into the sample immediately; instead depending upon the current position of u and v different subroutines are invoked to place the edge (i.e., two end nodes) into the buffer. This in turn may remove an existing node out of the buffer and connect it with its parent in G_s. C_s is adjusted accordingly. In this section, we describe different subroutines in details.

(i) Both u and v are present in the sample: When both u, v are in V_s, $BothinSample()$ (see Function 1) is called from line 15 of Algorithm 1. We further divide this case into two subcases: e_t is an intra-community edge (totally inside a single community) or an inter-community edge (connecting two communities $C(u)$ and $C(v)$). In the former case (edge $\{b, d\}$ in Figure 4.1), addition of e_t will strengthen the internal community structure according to Proposition 1[3]. We also know from Proposition 2 that adding an intra-community edge should not split the current community. Therefore we leave C_s in its current form without any modification.

In case of e_t connecting two different communities (edge $\{b, f\}$ in Figure 4.1), several possibilities may arise. u (or v) may leave its current community and join in other community. Additionally, if the community membership of u (or v) is changed, it can also pull out its neighbors to join with it, and some of the neighbors might eventually

[3]Detailed proofs of all the propositions can be found in [1].

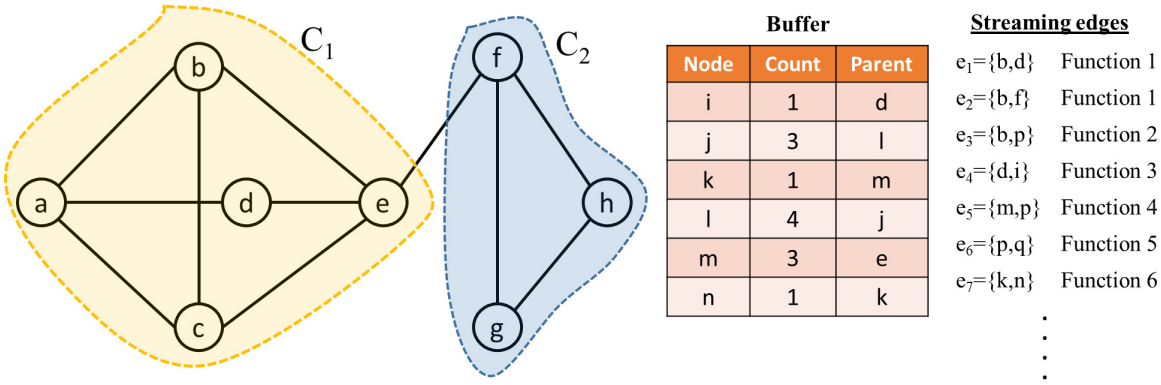


Figure 4.1: Toy example depicting various conditions handled by ComPAS when a streaming edge arrives.

want to change their memberships as well. According to Proposition 3, we know that if u (or v) ever changes its community membership, $C(v)$ (or $C(u)$) would be the best new community for it. But how do we quickly decide it? Here we provide a criteria to check the change in membership for u and v in Proposition 4. If both $\Delta Q(u, C(u), C(v))$ and $\Delta Q(v, C(v), C(u))$ (where $\Delta Q(u, C(u), C(v))$ indicates the change in modularity after assigning u from $C(u)$ to $C(v)$) fail to satisfy the criteria (see Corollary 1), we can retain the current community structure. Otherwise, we move u (or v) to $C(v)$ (or $C(u)$) and consequently we let its neighbors decide their best move in the similar way.

Proposition 1. *For a community $c \in C$, if $D_c \leq M - 1$ (where $M = |E|$) then addition of an edge within c will increase its modularity.*

Proof. From Equation 4.1, we see the contribution of individual community $c \in C$ in modularity as: $Q_c = \frac{m_c}{M} - \frac{D_c^2}{4M^2}$.

Addition of a new edge within c, the c's contribution of modularity becomes:

$$Q'_c = \frac{m_c + 1}{M + 1} - \frac{(D_c + 2)^2}{4(M + 1)^2}$$

So the increase in modularity is $\Delta Q_c = Q'_c - Q_c$,

$$\begin{aligned}\Delta Q_c =& \frac{4M^2 - 4m_cM^2 - 4D_cM^2 - 4m_cM + 2D_c^2M + D_c^2}{4(M+1)^2M^2} \\ &\geq \frac{4M^2 - 6D_cM^2 - 2D_cM + 2D_c^2M + D_c^2}{4(M+1)^2M^2} \\ &\geq \frac{(2M^2 - 2D_cM - D_c)(2M - D_c)}{4(M+1)^2M^2} \\ &\geq 0\end{aligned}$$

The equality holds if $D_c \leq M - 1$. This thus implies $(2M^2 - 2D_cM - D_c) \geq 0$. This proves the proposition. □

Proposition 2. *For a community $c \in C$, addition of any intra-community edge into c should not split it into smaller communities.*

Proof. We will prove this proposition by contradiction. Assume that once a new intra-community edge is added into c, it gets split into k small modules, namely X_1, X_2, $\cdot$,X_k. Let D_{X_i} and e_{ij} be the total degree of nodes inside X_i and number of edges connecting X_i and X_j respectively.

Recall that the contribution of X_i in the modularity value is $Q_{X_i} = \frac{m_{X_i}}{M} - \frac{D_{X_i}^2}{4M^2}$. Before adding the edge, we have $Q_c \geq \sum_{i=1}^{k} Q_{X_i}$ (where Q_c is the total modularity of community c), because otherwise all X_is can be split earlier, which is not in this case. This implies that: $\frac{m_c}{M} - \frac{D_c^2}{4M^2} > \sum_{i=1}^{k}(\frac{m_{X_i}}{M} - \frac{D_{X_i}^2}{4M^2})$. Since $X_1, X_2, \cdot, X_k$ are all disjoint modules of c, $D_c = \sum_{i=1}^{k} D_{X_i}$ and $m_c = \sum_{i=1}^{k} m_{X_i} + \sum_{i<j} e_{ij}$. This further implies that:

$$\frac{m_c}{M} - \sum_{i=1}^{k} \frac{m_{X_i}}{M} > \frac{D_c^2}{4M^2} - \sum_{i=1}^{k} \frac{D_{X_i}^2}{4M^2}$$

or, $\sum_{i<j} e_{ij} > \frac{\sum_{i<j} D_{X_i} D_{X_j}}{2M}$.

Without loss of generality, let us assume that the new edge is added inside X_1. Since we assume that after adding the new edge into c, it gets split into k small modules, the

modularity value should increase because of the split. Therefore, This implies that

$$\begin{aligned}
&Q'_c < \sum_{i=1}^{k} Q_{X_i} \\
&\Leftrightarrow \frac{\sum_{i=1}^{k} m_{X_i} + \sum_{i<j} e_{ij} + 1}{M+1} - \frac{(\sum_{i=1}^{k} D_{X_i+2})^2}{4(M+1)^2} \\
&< \frac{\sum_{i=1}^{k} m_{X_i} + 1}{M+1} - \frac{(D_{X_1}+2)^2}{4(M+1)^2} - \sum_{i=2}^{k} \frac{D_{X_i}^2}{4(M+1)^2} \\
&\Leftrightarrow \sum_{i<i} e_{ij} < \frac{\sum_{i=1}^{k} D_{X_i} - 2D_{X_1} + \sum_{i<j} D_{X_i} D_{X_j}}{2(M+1)}
\end{aligned}$$

Since $\sum_{i=1}^{k} D_{X_i} - 2D_{X_1} < 2M$, this implies that

$$\begin{aligned}
\frac{\sum_{i<j} D_{X_i} D_{X_j}}{2M} < \sum_{i<j} e_{ij} &< \frac{\sum_{i=1}^{k} D_{X_i} - 2D_{X_1} + \sum_{i\neq j} D_{X_i} D_{X_j}}{2(M+1)} \\
&< \frac{\sum_{i<j} D_{X_i} D_{X_j}}{2M} + 1
\end{aligned}$$

Therefore, the proposition holds. □

Proposition 3. *If a new edge (u, v) connecting two communities $C(u)$ and $C(v)$ is introduced, $C(u)$ (or $C(v)$) is the best candidate for v (or u) if it should ever change its membership.*

Proof. The method is inspired by [242] that a vertex u is influenced by two factors: $F_{in}^{c}(u)$ the force that keeps u stay in its own community c, and $F_{out}^{c}(u)$, the force that a community S imposes to u in order to bring u to S as follows:

$$F_{in}^{c}(u) = e_{c}^{u} - \frac{d_u(D_c - d_u)}{2M}$$

and

$$F_{out}^{S}(u) = max_{S \in NC(u)}\{e_{S}^{u} - \frac{d_u D_{outS}}{2M}\}$$

where $NC(u)$ is the set of neighboring communities of u, and D_{outS} is the total degree of vertices outside S.

Now we will show that the presence of new edge (u, v) will strengthen $F_{out}^{C(v)}(u)$ and

weaken $F_{out}^{S}(u)$. In other words, we will show that $F_{out}^{C(v)}(u)$ increases while $F_{out}^{S}(u)$ decreases for all $S \in C \wedge S \notin \{C(u), C(v)\}$.

$$
\begin{aligned}
& F_{out}^{C(v)}(u)|_{new} - F_{out}^{C(v)}(u)|_{old} \\
&= (e_u^{C(v)} + 1 - \frac{(d_u+1)(d_{outC(v)}+1)}{2(M+1)} - (e_u^{C(v)} - \frac{d_u d_{outC(v)}}{2M}) \\
&= \frac{2M + d_u d_{outC(v)}}{2(M+1)} - \frac{d_u d_{outC(v)} + d_{outD(v)} + d_u + 1}{2(M+1)} \\
&\geq \frac{2M + d_u d_{outC(v)}}{2(M+1)} - \frac{d_u d_{outC(v)} + d_{outC(v)} + d_u + 1}{2(M+1)} \\
&> 0
\end{aligned}
$$

Therefore $F_{out}^{C(v)}(u)$ is strengthened when a new edge (u, v) is introduced. Further, for any community $S \in C \wedge S \notin \{C(u), C(v)\}$

$$
\begin{aligned}
& F_{out}^{S}(u)|_{new} - F_{out}^{S}(u)|_{old} \\
&= (e_u^{S} - \frac{(d_u+1)d_{outS}}{2(M+1)}) - (e_u^{S} - \frac{d_u d_{outS}}{2M}) \\
&= d_{outS}(\frac{d_u}{2M} - \frac{d_u+1}{2(M+1)}) < 0
\end{aligned}
$$

This implies that $F_{out}^{S}(u)$ is weakened when (u, v) is added. Therefore, the proposition holds. □

Proposition 4. *If a new edge (u, v) is added into the graph, then joining u to v's community $C(v)$ will increase the modularity value if $\Delta Q(u, C(u), C(v)) \equiv 4(M+1)(e_{C(v)}^{u} + 1 - e_{C(u)}^{u}) + e_{C(v)}^{u}(2D_{C(v)} - 2d_u - e_{C(u)}^{u}) - 2(d_u+1)(d_u + 1 + d_{C(v)} - d_{C(u)}) > 0$.*

Proof. Vertex u will leave its current community $C(u)$ and join v's community $C(v)$ if

$$
\begin{aligned}
&Q_{C(v)+u} + Q_{C(u)-u} > Q_{c(u)} + Q_{C(v)} \\
&\Leftrightarrow \frac{m_{C(v)} + e_{C(v)} + 1}{M+1} - \frac{(d_{C(v)} + d_u + 2)^2}{4(M+1)^2} + \\
&\frac{m_{C(u)} - e_{C(u)}}{M+1} - \frac{(d_{C(u)} - d_u - e_{C(u)})}{4(M+1)^2} \\
&> \frac{m_{C(v)}}{M+1} - \frac{(d_{C(v)} + d_u + 2)^2}{4(M+1)^2} + \frac{m_{C(u)}}{M+1} - \frac{(d_{C(u)} + 1)^2}{4(M+1)^2} \\
&\Leftrightarrow 4(M+1)(e_{C(v)} + 1 - e_{C(u)}) + e_{C(u)}(2d_{C(v)} - 2d_{C(u)} - e_{C(u)}) \\
&- 2(d_{C(u)} + 1)(d_{C(u)} + 1 + d_{C(v)} - d_{C(u)}) > 0
\end{aligned}
$$

□

Corollary 1. *If the condition in Proposition 4 is not satisfied, then neither u nor its neighbors should be assigned to $C(v)$.*

Function 1: $BothinSample(u, v, e_t, V_s, E_s, C_s)$

```
1  if C_s(u) == C_s(v) then
2      E_s = E_s ∪ e_t
3  else
4      if ΔQ(u, C_s(u), C_s(v)) < 0 ∧ ΔQ(v, C_s(v), C_s(u)) < 0 then
5          return V_s, E_s, C_s
6      else
7          w ← argmax{ΔQ(u, C_s(u), C_s(v)), ΔQ(v, C_s(v), C_s(u))}
8          Move w to a new community and update C_S
9          for t ∈ N(w) do
10             Let t decide its own community
11             Update C_s
12 return V_s, E_s, C_s
```

(ii) u **is in sample and** v **is new:** When a new node v connecting node $u \in G_s$ appears, $OneinSampleOneNew()$ (see Function 2) is called from line 17 of Algorithm 1 (edge

$\{b, p\}$ in Figure 4.1). In this case, we do not add $\{u, v\}$ into the sample immediately. Instead, we first insert v into $\mathcal{H}$ if $\mathcal{H}$ is not full. If $\mathcal{H}$ is full, we pick one node x from $\mathcal{H}$ preferentially based on $\mathcal{H}_c[x]$ with an additional constraint that $\mathcal{P}(x)$, the parent of x should be in G_s[4] (for example, in Figure 4.1 if the buffer is full although node l has highest count we do not pick l from the buffer because its parent node i is not in G_s; instead we pick m which satisfies both the constraints). x is then added to G_s through the edge $\{\mathcal{P}(x), x\}$ (line 10 in Function 2) and is assigned the community of $\mathcal{P}(x)$ (line 11 in Function 2). We also check if including x in G_s violates the sample size constraint through the function $CheckResizeSample()$ (Function 7). If G_s is already full (i.e., $V_s = n$), one existing node which has lowest degree is removed from G_s and the community structure is adjusted accordingly using $CommunityAfterNodeRemoval()$ (both Functions 7 and 8 are discussed later).

Function 2: $OneinSampleOneNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$

1 **if** $\mathcal{H}$ *is not full* **then**
2 Insert v to $\mathcal{H}$
3 **else**
4 Choose a node x with $\mathcal{P}(x) \in V_s$ from $\mathcal{H}$ preferentially based on $\mathcal{H}_c$
5 Remove x from $\mathcal{H}$
6 $\mathcal{H}_c[x] = 0$
7 $\mathcal{H}_p[x] = \phi$
8 $V_s, E_s, C_s = CheckResizeSample(V_s, C_s, n, 1)$
9 $V_s = V_s \cup x$
10 $E_s = E_s \cup \{P(x), x\}$
11 $C_s(x) = C_s(P(x))$
12 Update C_s
13 Insert v to $\mathcal{H}$
14 $\mathcal{H}_c[v] = 1$
15 $\mathcal{H}_p[v] = u$
16 **return** $V_s, E_s, C_s, \mathcal{H}$

(iii) **u is in sample and v is in buffer:** In this case (edge $\{d, i\}$ in Figure 4.1),

[4]The additional constraint allows to avoid a disconnected component being created in the sample.

$OneinSampleOneinBuffer()$ (see Function 3) is called from line 19 of Algorithm 1. We first check the sample size constraint (line 4 in Function 3) and accordingly add v (and edge $\{u, v\}$) into G_s. v is further assigned to $C(u)$.

Function 3: $OneinSampleOneinBuffer(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$

1 Remove v from $\mathcal{H}$
2 $\mathcal{H}_c[v] = 0$
3 $\mathcal{H}_p[v] = \phi$
4 $V_s, E_s, C_s = CheckResizeSample(V_s, C_s, n, 1)$
5 $V_s = V_s \cup v$
6 $E_s = E_s \cup \{v, \mathcal{P}(v)\}$
7 $C_s(v) = C_s(u)$
8 Update C_s
9 **return** $V_s, E_s, C_s, \mathcal{H}$

(iv) u **is in buffer and** v **is new:** This case (edge $\{m, p\}$ in Figure 4.1, see Function 4) is similar to Function 2. We first increment the counter corresponding to u in the buffer and add v into the buffer in the same way as mentioned in Function 2.

Function 4: $OneinBufferOneNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$

1 $\mathcal{H}_c[u] = \mathcal{H}_c[u] + 1$
2 $V_s, E_s, C_s, \mathcal{H} = OneinSampleOneNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$
3 **return** $V_s, E_s, C_s, \mathcal{H}$

(v) Both u **and** v **are new:** When both u and v are new (edge $\{p, q\}$ in Figure 4.1), $BothNew()$ (see Function 5) is called from line 23 of Algorithm 1. We initially check whether buffer $\mathcal{H}$ is full or not. In case it is not full, we insert u in $\mathcal{H}$ and then call Function 2. Otherwise, we preferentially remove x and y from $\mathcal{H}$ based on the counts in $\mathcal{H}_c$ with the additional constraint that both $\mathcal{P}(x)$ and $\mathcal{P}(y)$ are in G_s, and add nodes x, y and edges $\{\mathcal{P}(x),x\}$, $\{\mathcal{P}(y),y\}$ into G_s. x and y are also assigned to the community of the their respective parents. Finally, u and v are inserted into $\mathcal{H}$. If during the insertion into G_s the sample size is violated, the required number of nodes along with their adjacent edges are deleted from G_s using $CheckResizeSample()$.

Function 5: $BothNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$

if $\mathcal{H}$ *is not full* **then**
  Insert u to $\mathcal{H}$
  $\mathcal{H}_p[u] = v$
  $\mathcal{H}_c[u] = 1$
  $V_s, E_s, C_s, \mathcal{H} = OneinSampleOneNew(u, v, e_t, V_s, E_s, \mathcal{H}, C_s)$
else
  Choose nodes x, y with $\mathcal{P}(x), \mathcal{P}(y) \in V_s$ from $\mathcal{H}$ preferentially based on $\mathcal{H}_c$
  Remove x, y from $\mathcal{H}$
  $\mathcal{H}_c[x] = 0$, $\mathcal{H}_c[y] = 0$
  $\mathcal{H}_p[x] = \phi$, $\mathcal{H}_p[y] = \phi$
  $V_s, E_s, C_s = CheckResizeSample(V_s, C_s, n, 2)$
  $V_s = V_s \cup x \cup y$
  $E_s = E_s \cup \{x, \mathcal{P}(x)\} \cup \{y, \mathcal{P}(y)\}$
  $C_s(x) = C_s(\mathcal{P}(x))$
  $C_s(y) = C_s(\mathcal{P}(y))$
  Update C_s
  Insert u, v to $\mathcal{H}$
  $\mathcal{H}_p[u] = v$, $\mathcal{H}_p[v] = u$
  $\mathcal{H}_c[u] = 1$, $\mathcal{H}_c[v] = 1$
return $V_s, E_s, C_s, \mathcal{H}$

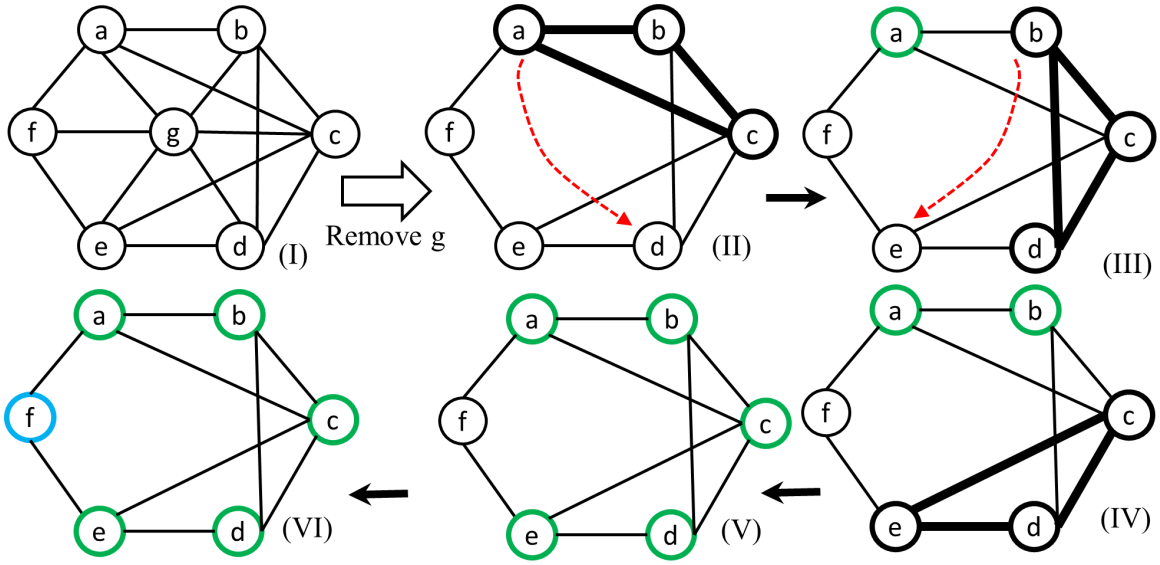


Figure 4.2: Illustrative example of 3-clique percolation. Once node g is removed, a 3-clique is placed on node a. The clique percolates and accumulates all the nodes except node f which forms a singleton community along with $\{a, b, c, d, e\}$.

(vi) Both u and v are in buffer: In this case (edge $\{k, n\}$ in Figure 4.1), $BothinBuffer()$ (see Function 6) is called from line 24 of Algorithm 1 whereby, only the buffer $\mathcal{H}$ is modified by increasing $\mathcal{H}_c$ entries of u and v by 1.

Function 6: $BothinBuffer(u, v, \mathcal{H})$

1 $\mathcal{H}_c[u] = \mathcal{H}_c[u] + 1$
2 $\mathcal{H}_c[v] = \mathcal{H}_c[v] + 1$
3 **return** $\mathcal{H}$

• Adjust communities after removing a node: When G_s is full, we remove m nodes (and their adjacent edges) from the sample using $CheckResizeSample()$ (Function 7). Since each such node u is already a part of its community $C(u)$, its deletion might keep the previous community structure unchanged, or break the community into smaller parts, or merge several communities together. The community structure C_s is adjusted using $CommunityAfterNodeRemoval()$ (Function 8) incrementally. Let us consider two extreme cases – a node with degree 1 is removed, and a node with highest degree is removed. Removal of a single degree node will keep the community unchanged. However, removal of a highest degree node can render the community disconnected or broken into smaller parts which might further merge to the other existing communities. Here we utilize

the clique percolation method [169] to handle this situation efficiently. In particular, when a vertex v is removed from a community C, we place a 3-clique to one of its neighbors and let the clique percolate until no vertices in C are discovered. Nodes discovered in each such clique percolation will form a community. We repeat this clique percolation from each of v's neighbors until each member in C is assigned to a community. For example, in Figure 4.2 when node g is removed, we place a 3-clique on its neighbor a. Once the 3-clique starts percolating, it accumulates all nodes except f. Therefore, two new communities $\{a, b, c, d, e\}$ and $\{f\}$ emerge due to the deletion of g. In this way, we let the remaining communities of C choose their best communities to merge in.

Function 7: $CheckResizeSample(V_s, C_s, n, m)$

```
if V_s == n then
    Remove m nodes
    say, u_1, u_2, ··· , u_m (and all their adjacent edges) from G_s having lowest degree
    for u ∈ {u_1, u_2, ··· , u_m} do
        C_s ← CommunityAfterNodeRemoval(u, C_s)
return V_s, E_s, C_s
```

Function 8: $CommunityAfterNodeRemoval(u, C_s)$

```
Assume node u and its adjacent edges are removed from G_s
i = 1
while N(u) ≠ φ do
    b_i=Nodes found by a 3-clique percolation on v ∈ N(u)
    if b_i == φ then
        b_i = {v}
    C_s = C_s ∪ b_i
    N(u) = N(u) \ b_i
    i = i + 1
Update C_i
return C_s
```

4.4 Experimental Setup

In this section, we describe the baseline sampling algorithms and the datasets used in our experiments.

4.4.1 Sampling algorithms

We compare our method with five existing sampling methods: (i) Streaming Node (SN) [5], (ii) Streaming Edge (SE) [5], (iii) Streaming BFS (SBFS) [5], (iv) PIES [4], and (v) Green algorithm (GA) [217]. The first four algorithms are exclusively designed for streaming graphs while the last one is designed for static graphs. Note that unlike ours, none of the existing algorithms explicitly produce a community structure as a bi-product of the sampling[5], and thus one needs to execute community detection algorithm separately on the sample to obtain the community structure. Therefore to evaluate the competing algorithms with respect to how the underlying community structure in the sample resembles with that of the original graph, for SN, SE, SBFS and PIES we run Louvain algorithm [28] on each individual sample and detect the communities. In case of GA, we consider the aggregated graph and run GA to obtain the sample, and further run Louvain on the sample to detect the community structure. Note that although by considering the aggregated graph GA acquires far more information of the entire graph structure, we use it as a strict baseline in this study to show that the community structure obtained from `ComPAS` is quite competitive to that obtained by running Louvain on GA's sample graph.

4.4.2 Datasets

We perform our experiments on five graphs mentioned below. While first two are streaming graphs, last three graphs are static graphs.
(i) **Facebook**[6]: This is an undirected graph with nodes representing users, and an edge

[5]Although GA claims that its sample graph preserves the underlying community structure, it does not explicitly produce the community structure.

[6]konect.uni-koblenz.de/networks/facebook-wosn-links

exists between two users if they are friends. Each edge is time stamped by the occurrence of the friendship. The graph consists of $63,731$ nodes and $817,035$ edges.
(ii) **arxiv hep-th**[7]: This dataset represents the collaboration graph of the authors in arXiv's High Energy Physics papers. Each node represents an author and an edge exists between two authors if they have co-authored a paper. The edges are time stamped by publication date of the co-authored paper. The graph consists of 22,908 nodes and 2,673,133 edges.
(iii) **Youtube**[8]: This dataset represents the Youtube social network with nodes representing users and edges representing friendship. The graph consists of 1,134,890 nodes and 2,987,624 edges.
(iv) **Dblp**[9]: This dataset consists of authors indexed in DBLP. The graph is same as arxiv hep-th. There are 317,080 nodes and 1,049,866 edges in the graph.
(v) **LFR**: This is a synthetic graph [127] with underlying community structure implanted into it. We construct the graph with 25,000 nodes, 254,402 edges and 18,34 communities.

Since last three graphs are static, we consider that each edge arrives in a pre-decided order (which is set randomly) i.e., each edge has a (discrete) time of arrival. We later show that edge ordering does not influence the inferences drawn from the results (Section 4.5.3).

Moreover, since first four graphs do not have any underlying ground-truth community structure, we run Louvain algorithm [28] on the aggregated graph and obtain the disjoint community structure. This community structure is the best possible output that we can expect from our incremental modularity maximization method, and therefore serves as the ground-truth.

4.5 Evaluation

We design a two-fold experimental setup. First, we show how competing sampling algorithms detect the original community structure, and second, we measure how good individual samples are w.r.t. the structural properties of the original graph. Although the

[7] konect.uni-koblenz.de/networks/ca-cit-HepTh
[8] snap.stanford.edu/data/com-Youtube.html
[9] snap.stanford.edu/data/com-DBLP.html

Table 4.2: Summary of the D-statistics (the lower, the better) values of the topological measures for all the datasets. For Youtube we present all the results, while for the rest we provide the average D-statistics and standard deviation (SD). ComPAS truns out to the second best algorithm after GA (the most informed static graph sampling algorithm for which the sample is obtained from the aggregated graph and Louvain is run on the sample, thus serving as the strict baseline). Top two values for each average result is highlighted.

Algorithm	Youtube														Facebook	Com-dblp	LFR	hep-th
	ID	EI	AD	FOMD	TPR	EX	CR	CON	NC	AODF	MODF	FODF	MOD	Avg,SD	Avg,SD	Avg,SD	Avg,SD	Avg,SD
ComPAS	0.063	0.051	0.078	0.057	0.227	0.082	0.054	0.091	0.260	0.073	0.201	0.121	0.052	**0.10,0.07**	**0.17,0.09**	**0.16,0.10**	**0.18,0.06**	**0.10,0.03**
SN	0.164	0.171	0.471	0.061	0.542	0.581	0.112	0.265	0.064	0.157	0.182	0.092	0.216	0.23,0.17	0.33,0.17	0.29,0.20	0.27,0.07	0.26,0.04
SE	0.257	0.244	0.241	0.501	0.281	0.098	0.287	0.087	0.151	0.097	0.246	0.093	0.198	0.21,0.11	0.27,0.11	0.25,0.14	0.32,0.08	0.29,0.06
SBFS	0.126	0.131	0.172	0.106	0.454	0.145	0.056	0.165	0.045	0.257	0.108	0.076	0.181	0.15,0.10	0.26,0.09	0.24,0.10	0.25,0.09	0.26,0.04
PIES	0.234	0.241	0.252	0.190	0.409	0.042	0.051	0.049	0.061	0.157	0.042	0.053	0.121	0.14,0.10	0.29,0.06	0.24,0.07	0.26.0.05	0.21,0.05
GA	0.156	0.055	0.065	0.053	0.267	0.066	0.076	0.053	0.085	0.150	0.075	0.069	0.102	**0.09,0.06**	**0.12,0.04**	**0.12,0.06**	**0.14,0.06**	**0.08,0.04**

primary focus of ComPAS is to quality better in the first evaluation, we also show that ComPAS is quite competitive to retain the graph structure in the second evaluation.

4.5.1 Community-centric evaluation

In this section, we start by explaining the metrics used to evaluate the goodness of the community structure, followed by a detailed comparison of the sampling algorithms.

4.5.1.1 Evaluation criteria

To measure how sampling algorithms capture the underlying community structure, we evaluate them in two ways. First we measure the quality of the obtained community structure based on the **topological measures** defined by Yang et. al. [241]. In particular, we look into four classes of quality scores - (i) *based on internal connectivity*: internal density (ID), edge inside (EI), average degree (AD), fraction over mean degree (FOMD), triangle participation ratio (TPR); (ii) *based on external connectivity*: expansion (EX), cut ratio (CR); (iii) *combination of internal and external connectivity*: conductance (CON), normalized cut (NC), maximum out-degree fraction (MODF), average out-degree fraction (AODF), flake out-degree fraction (FODF); and (iv) *based on graph model*: modularity (MOD)[10]. Note for every individual community we obtain a score, and therefore a distribution of scores (i.e., distribution of ID, distribution of EI etc.) is obtained for all

[10]See [241] for the detailed definitions of all these metrics.

Table 4.3: NMI between the ground-truth and community structure obtained from individual sampling algorithms for all datasets.

Dataset	ComPAS	SN	SE	SBFS	PIES	GA
Facebook	0.52	0.34	0.28	0.41	0.48	0.61
hep-th	0.51	0.32	0.21	0.36	0.39	0.68
Youtube	0.72	0.49	0.33	0.58	0.51	0.77
Dblp	0.65	0.28	0.21	0.57	0.39	0.69
LFR	0.69	0.29	0.32	0.38	0.31	0.72
Average	**0.61**	0.34	0.27	0.46	0.41	**0.69**

the communities of a graph. We measure how similar (in terms of Kolgomorov-Smirnov D-statistics[11]) these distributions are with those of the ground-truth communities. *The less the value of D-statistics, the better the match between two distributions*.

As a second level of evaluation, we use the **community validation metrics** – Purity [147], Normalized Mutual Information (NMI) [55] and Adjusted Rand Index (ARI) [102] to measure the similarity between the ground-truth and the obtained community structures. *The more the value of these metrics, the more the similarity.*

4.5.1.2 Parameter estimation

As reported in Section 4.3, our algorithm consists of two parameters (i) α (initial fraction of nodes inserted), (ii) n_d (length of the buffer). For α, we observe that D-statistics is initially high and reduces as we increase α (Figures 4.3(a)). This is because for low α values the community structure obtained initially by running Louvain algorithm (Step 12 in Algorithm 1) is coarse. For large values of α even though initial community structure obtained is good, it is not allowed to evolve much. Similarly in Figure 4.3(b), given a small buffer size several nodes mostly arriving once would be added to the sample leading to formation of pendant vertices. As we increase the buffer size ComPAS performs better till a certain point, after which the improvement is negligible. Since we are interested in using minimum space we fix n_d at $0.001 \cdot n$. Thus, for the rest of the experiments we set α to 0.5 and n_d to $0.001 \cdot n$ unless otherwise specified. Further we set n to $0.4|V|$ as default (see Section 4.5.3 for different values of n).

[11] It is defined as $D = max_x\{|f(x) - f^{'}(x)|\}$ where x is over the range of the random variable, and f and $f^{'}$ are the two empirical cumulative distribution functions of the data.

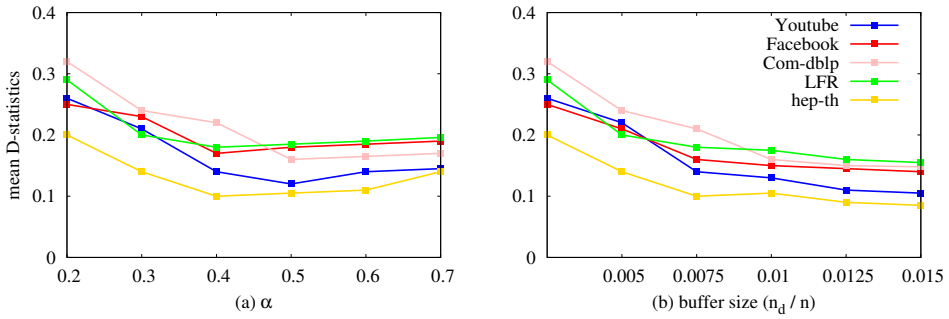


Figure 4.3: Average D-statistics value across all the topological measures.

4.5.1.3 Comparison of sampling algorithms

We start by measuring the similarity between the obtained and the ground-truth community structures using topological measures. In Table 4.2 we summarize the D-statistics values of all the scoring functions for Youtube dataset; for the other graphs we only present the average value (and standard deviation) across the D-statistics among different topological measures (see [1] for detailed results). Clearly `ComPAS` outperforms all the streaming algorithms across different datasets. GA performs better than `ComPAS` since it has in its consideration the whole graph to obtain the sample. However, we stress that even with *minimal community information to start with* and *no subsequent community detection in the later steps*, we are able to reach very close to GA as well as to the ground-truth.

As a second level of evaluation, we further calculate three validation metrics – purity, NMI and ARI between the ground-truth and the obtained community structure for all the algorithms (see Table 4.3 for NMI, details in [1]). Once again we observe that `ComPAS` is the second ranked algorithm after GA with an average (over all datasets) purity, NMI and ARI of 0.74, 0.61 and 0.53 respectively.

4.5.2 Graph-centric evaluation

Although our primary objective is to obtain a community-preserved graph sample, we further evaluate our sample in terms of three graph properties mentioned in [5]: (i) degree distribution (Degree), (ii) clustering coefficient distribution (CC), (iii) top 100 eigen value

distribution (EV). In Table 4.4 we report the D-statistics values between the distributions of the original graph and those obtained from the sample (see more results in [1]). For all the datasets, we observe that ComPAS stands within top three ranks in terms of low D-statistics, in many cases, also beating the strict baseline GA. This essentially indicates that ComPAS, apart from preserving the community structure, also preserves general graphs properties in the sample.

Table 4.4: Summary of D-statistics for different graph properties. For Youtube we present all the results, while for the rest we provide only average D-statistics (top three results in each average case are highlighted).

Algorithm	Youtube				Facebook	Com-dblp	LFR	hep-th
	Degree	CC	EV	Average	Average	Average	Average	Avearge
ComPAS	0.083	0.105	0.42	**0.20**	**0.17**	**0.16**	**0.28**	**0.18**
SN	0.076	0.108	0.54	0.24	0.36	0.24	0.34	0.23
SE	0.114	0.195	0.39	0.23	0.48	0.28	0.39	0.27
SBFS	0.105	0.172	0.32	**0.19**	**0.28**	**0.13**	**0.26**	**0.19**
PIES	0.046	0.129	0.38	**0.18**	**0.16**	0.21	**0.28**	**0.16**
GA	0.218	0.063	0.46	0.24	0.35	**0.12**	0.32	0.20

4.5.3 Effect of edge ordering and sample size

As mentioned in Section 4.4.2, we artificially imposed an edge ordering for two static graphs – Youtube and LFR. In this section, we show that most of our inferences are valid irrespective of any edge ordering. To do so, we randomly pick one pair of edges and swap their arrival time. We repeat it for 20% of edges present in each static graph. The entire experiment is repeated 20 times and the average value is reported. Figure 4.4(a) shows that for Youtube graph edge ordering does not affect much the final sample obtained (the pattern is same for LFR graph, see [1]).

Lastly, we present the effect of sample size (n) on the obtained community structure. We plot average D-statistics values across all the topological measures for all the algorithms on Youtube dataset (see others in [1]) as a function of n (Figure 4.4(b)). As expected, with the increase of n we obtain better results. Interestingly, for ComPAS and GA, the pattern

remains consistent compared to others.

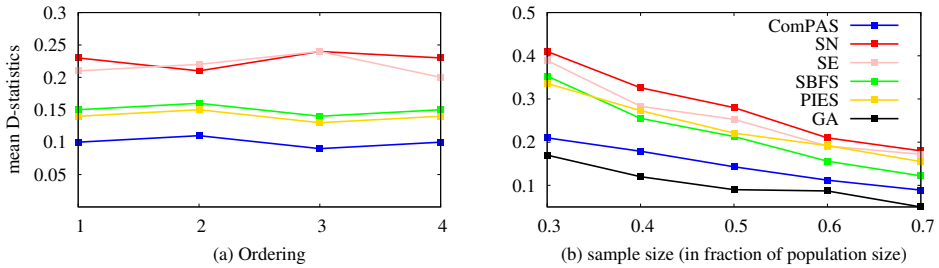


Figure 4.4: Average D-statistics across all the topological measures for (a) different edge ordering and (b) sample size (n) of Youtube graph.

4.6 Applications of compas

Here we present two additional applications of `ComPAS` – ranking community detection algorithms, and selecting appropriate training set for (restricted) online learning.

4.6.1 Ranking community detection algorithms

As pointed out in [146], there is always a trade-off between quality and running time for discovering the community structure from a graph. While some algorithms discover excellent community structure with larger running time, others compromise quality by achieving lesser running time. Further, it is not possible to identify beforehand which community detection algorithm is more useful given a large graph. In this case, one is forced to run several algorithms on the large graph and converge on the best performing one. This may lead to a large computation cost and is especially detrimental in resource-scarce situations (absence of parallel infrastructures, distributed systems etc.).

In case of community detection on large streaming graphs, we posit that one may first execute `ComPAS` and obtain a community-preserved sample. Then different community detection algorithms can be tested on the sample (which is much smaller in size compared to the original graph) and the best performing algorithm can be used for the original large

Table 4.5: Rank correlation of community detection algorithms based on the performance on the sample (generated from individual sampling methods) and the original graph. (b) Performance of SVM using the training set obtained from sampling methods.

(a)

Algo	Facebook	hep-th	Youtube	DBLP	LFR
ComPAS	**0.8**	**0.8**	**0.8**	**1.0**	**1.0**
SN	0.4	0.6	0.4	0.6	0.6
SE	0.4	0.4	0.4	0.6	0.4
SBFS	0.6	0.7	0.6	0.8	0.6
PIES	0.6	0.6	0.8	0.7	0.8
GA	**1.0**	**0.8**	**0.8**	**1.0**	**1.0**

(b)

AUC	F-Score
0.48	**0.61**
0.31	0.35
0.25	0.28
0.28	0.31
0.36	0.43
0.53	**0.64**

graph. To this end, we execute five community detection (CD) algorithms (CNM [50], Fast-Greedy [159], Walktrap [180], Infomod [193], Infomop [195]) separately on each sample obtained from the sampling algorithms. Then we rank the CD algorithms based on their performance in terms of modularity [158]. We further rank the CD algorithms in the same way considering the *entire graph*. We report the Kendall's τ rank correlation in Table 4.5(a) between these two ranks for individual sampling algorithms across all the datasets. We find GA and ComPAS performing the best. Therefore, ComPAS can be used for selecting a CD algorithm quickly before running on a large streaming graph.

4.6.2 Selecting training set for online learning

In online learning, sometimes memory is limited and it is required to train the model on limited number of instances. In such situations it is important to choose a training set in such a way that it consists of members from most or all parts of the original dataset.

We hypothesize that more diverse the chosen set, better would be the performance. ComPAS is useful in such cases since it tries to sample from several communities, hence improving the diversity of the training set. To this end, we consider Wiki-Rfa[12] [234], a streaming signed graph in which nodes represent Wikipedia members and edges (with time-stamp) represent votes. Each vote is typically accompanied by a short comment. The task is to predict the vote (+1, 0, -1) of an incoming edge based on the textual features – (i) word count, (ii) sentiment value, and (iii) LIWC features of the statement corresponding to the edge. We allow training instances to be included till a certain time period t (first 75%

[12]https://snap.stanford.edu/data/wiki-RfA.html

of the edges are allowed to enter) and run the sampling algorithms in parallel. However not all instances can be considered for training due to the memory constraint. We assume n, the sample size as the allowed training size and obtain sampled training set from individual sampling algorithms. We train SVM with linear kernel (see [1] for other classifiers) on each sampled training set, and predict the labels (votes) of those instances coming after t. Table 4.5(b) shows that GA and ComPAS perform the best in terms of AUC and F-Score. This once again emphasizes the fact that ComPAS selects most representative training instances for (restricted) online learning.

4.7 Summary of the chapter

The contributions of this chapter can be summarized as follows -

- we proposed ComPAS, a novel sampling algorithm for streaming graphs which is able to retain the community structure of the original graph.
- through rigorous experimentation on four real-world and one synthetic graphs we showed that our algorithm performs better than five state-of-the-art graph sampling algorithms.
- we further showed the effectiveness of ComPAS through two real-world applications

As an immediate future work we plan to analytically estimate the relation between sample size and the accuracy of our algorithm. It would also be interesting to investigate in detail the effectiveness of ComPAS in other real applications.

Chapter 5

Diffusion in temporal networks

Previous two chapters dealt with analyzing structural properties of temporal networks. This chapter is devoted towards the functional aspect of time-varying networks. In specific, we are interested in studying diffusion dynamics in these networks. This chapter is divided into two parts. To start with we propose a threshold based diffusion model and systematically analyze the diffusion rate for different underlying graph topologies. We then leverage these results to obtain a message broadcast algorithm in dynamic networks.

5.1 Introduction

Information diffusion is one of the most common phenomena that occurs on a network and the most elementary model of this process is SI (Susceptible-Infected) [9] and its different variations [11, 59, 119, 205, 213, 230]. Epidemic/adoption models have recently found a renewed interest with the formulation of temporal networks and realization that most of the real-world networks are temporal in nature (network structure changes with time).

The fundamental difference between static and temporal epidemic (SI) models is that in temporal models every agent within a population is *not equally susceptible* to a disease or equally amenable to a rumor - the one which has been exposed more number of times (in recent past) are more amenable. This difference however is not well formulated and

hence not well modeled - the primary contribution of this letter is to succinctly define the problem in terms of a simple model and then theoretically calculate the rate of spread of the epidemics. We consider a spreading model in the lines of [211] where each susceptible node needs to communicate with the infected nodes *multiple times* to contract infection. More importantly, unlike memoryless systems, we assume that each node comprises a memory which keeps track of the number of contacts it makes with infected ones. Note that in our system memory is a property which allows each node to remember the number of contacts it has already made with the infected ones. While Karimi et. al. studied the effect of burstiness in the diffusion process, we are more interested in estimating both empirically and analytically the diffusion time and rate. Note that this model is completely different from threshold models [83, 150] where a node gets infected when majority of its neighbors are infected or probabilistic SI-models where an infected node, on coming in contact, infects a susceptible node with a probability p, as those are memoryless systems and the transition depends only on the activity of present time step. Our diffusion model also differs from the Neighborhood Exchange (NE) model proposed in [225]. NE assumes at any given time, an individual will be in contact with an individual-specific number of neighbors with whom disease transmission is possible while our model considers that a node can be in contact with only a single individual. Also NE does not consider the fact that multiple contacts are required to contract infection.

The analytical results are obtained considering simple yet important topologies like the complete graph and the infinite d-regular trees which accounts for two extreme variants of network topology in terms of edge density. We demonstrate that the total diffusion time required for complemete graph topology (with n nodes) scales as $n^{\frac{k-1}{k}}$ where $k > 1$ is the number of contacts required to infect a susceptible individual. This is in sharp contrast to the case with $k = 1$, where it has been shown that diffusion time scales as $\log(n)$ [47, 133]. Another important inference we draw from the theoretical analysis is that irrespective of the topology the diffusion process could be divided into two phases: (i) an initial phase where the diffusion rate is very slow and (ii) a residual phase where the diffusion becomes very fast. This inference is also a crucial contribution of this work which can help in containing the spread of infectious diseases with minimum overhead if acted upon in the initial phase which we further prove through a detailed empirical study.

Broadcast algorithm based on diffusion model: The study of broadcast over unstructured and mobile networks always assumes that the size of the message is small enough to be transfered from one node to other on the short durations of contacts between the nodes. Contrary to this, we here explore the idea of "segmented messages" where we assume that the duration of a contact between the nodes is not always sufficient for the transfer and therefore the message might need to be segmented/divided into sub-parts and sent individually. At the ethernet level such techniques of segmented broadcast are often termed as pipelined broadcast [173, 231]. Note that this is equivalent to diffusion model we proposed earlier with number of contacts (k) replaced by number of packets in a message. We systematically study the effect of the size and partition structure of the message on the broadcast time.

In specific, we investigate in details the effect of message segmentation as well as the message transfer protocols on the overall broadcast delay and message wastage. We assume that a big file is split into k (> 1) packets and at the beginning, there is only one sender node in the network that has all the packets. Further, a node can transfer only one packet in a single contact opportunity, and it can do so only when it has received all the packets constituting at least one segment of the message. When all the nodes present in the network have eventually received all the packets, broadcasting is assumed to be complete.

Initially, we investigate the push transfer protocol whereby messages are 'pushed' by the node holding a message to the node not having the message [56, 139]. We attempt to study the effect in different types of topologies e.g., complete graph, d-regular tree, d-regular graph, random graph (with average degree d). A remarkable observation is that in topologies like d regular tree, d regular graph and random graph, for even the simple push transfer protocol, one can find an optimal value of $d(d > 1)$, for which the broadcast delay and wastage is minimum. (section 5.7). As a corollary, through simulations on real traces, we identify that for two networks with the same number of nodes, broadcast time required is far smaller for the one with lesser edge density. This finding, we believe, indicates a very crucial point – a sparse communication network per se is not disadvantageous.

However, we observe that push transfer protocol results in a large number of useless contacts. An unsuccessful/useless contact here refers to a case where a sender node attempts to send a packet to another node who already has the packet. In order to reduce both broad-

cast time and wastage we propose a combined strategy whereby the nodes in the system initially push and then switch over to pull after a certain percentage (say x) of the nodes have received the full message. We observe that if x is carefully chosen, both gain in broadcast time and reduction in wastage is achieved. However, to determine that $x\%$ of the nodes have indeed received the message, the system needs to maintain a global information which is not feasible in a distributed setting like this. In order to circumvent this problem, we introduce a distributed version of the previous technique along with "give-up" mechanism whereby nodes attempt to discover their neighborhood by maintaining a history of all previous contacts and stops participating in the broadcast after a certain number of unique unsuccessful contacts. This algorithm when tested over Gnutella topologies is found to yield lesser broadcast delay without significant increase in wastage. We believe, our algorithm with minor modifications is applicable to a wide range of dynamic networks [65, 108, 197].

5.2 Diffusion model

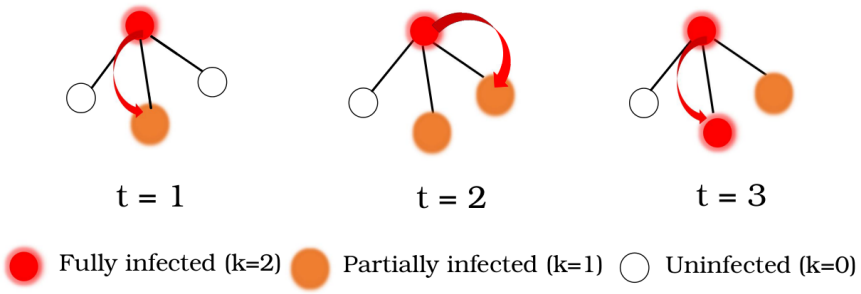


Figure 5.1: Proposed diffusion model for $k = 2$.

Formally, we consider a network topology $G = (V, E)$ where V represents the set of nodes in the network and E denotes the set of edges between any pair of nodes in V. We initially start with a single infected node in the system. We further assume that for a susceptible node to get infected, k encounters with infected nodes are required. To put it in a simple way we consider that a message M needs to be spread over a network and message M consists of k identical tokens. At each communication instance one token gets transmitted from an infected node to the susceptible node. Therefore, number of tokens (k) in a

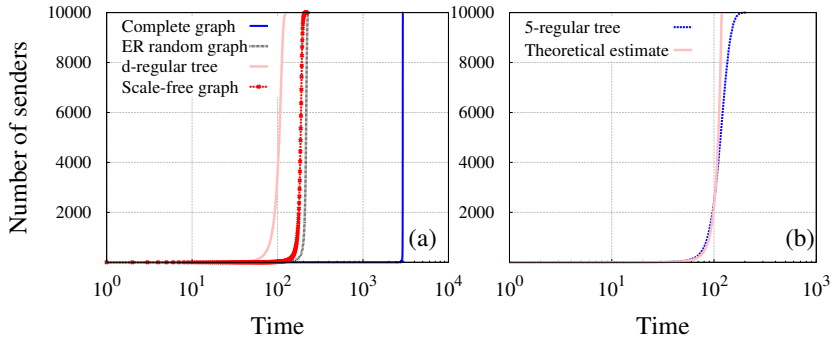


Figure 5.2: (a) The number of senders versus time steps for different network topologies (For ER random graph p is 0.005, for d-regular graph $d = 5$ and for BA scale-free graph m i.e., number of edges to attach from a new node to existing nodes is 5) and (b) The theoretical estimate and the simulated result for $d-$regular trees. The theoretical eastimate is obtained from equation 17.

message corresponds to the number of contacts required for a susceptible node to get infected. Note that for the rest of the paper we will present our diffusion model in terms of messages and tokens.

We assume that at time 0, there is only one sender (infected) node present in the system and it acts as the *initiator* of the diffusion process. At each discrete time step a sender node randomly selects one of its neighbors and there is a transfer of a token from the sender to the non-sender. A non-sender node becomes a sender only after it receives exactly k tokens (refer to figure 5.1). The analytical estimation of the diffusion time based on the underlying topology requires a case-by-case examination. We formulate both analytical and empirical results for two extreme variants (in terms of edge density) of networks (a) complete graphs (dense) and (b) infinite regular trees (sparse) while for others we provide empirical results with intuitive justifications.

5.3 On Complete graph

For **complete graph** we assume the number of nodes in the system to be n. To determine how the number of senders in the system changes with time, we plot the number of senders against time for complete graph in Fig. 6.1. We observe that the diffusion is initially

slow which is then followed by a ramp-up after which the diffusion rate becomes almost exponentially fast. To better analyze the process we divide the process into two phases i) the initial phase and ii) the residual phase.

Initial phase: In the spreading process we define the initial phase to be the time between the initiation and the point at which the first sender is created. We define this as t_1; we will show later that $\hat{E}(t_1)$ (expected value of t_1) is indeed an indicator for $\hat{E}(T^*)$ (T^* - total diffusion time) in case of a complete graph.

Note that analytically deriving $\hat{E}(t_1)$ assuming a discrete (i.e., for a node the delay between two successive contacts is 1 unit) diffusion model becomes severely complex and hence we adopt a continuous variant of the model. In fact, the calculation of the $\Pr\{t_1 = t\}$ can be treated as an expected time of filling the first urn with k balls in the experiment where we have initially d empty urns (degree of the node, for complete graph $d \sim n$) and at each single time step we add a single ball to one urn chosen randomly. For any k-parts message, by [110] we describe our problem as a unit-time Poisson process. Note that for a poisson process the inter-arrival time follows exponential distribution (λ) and expected number of arrivals in time t is λt. Let $X_j(t)$ be a random value describing the number of balls in j^{th} urn up to time t. More precisely $X_j(t) = \sum_{i=1}^{N} 1$ where N is a random variable with Poisson distribution $\mathcal{P}(\frac{t}{d})$. Essentially N represents number of draws of the j^{th} urn up to time t if d urns exist in the system. Hence $\{X_j(t)\}$s are i.i.ds and $X_j(t) \sim \mathcal{P}(\frac{t}{d})$. We formulate the analytical result for complete graph topology through the following theorems 1-5. The various notations used in the paper are further summarized in table 5.1

Theorem 1. *For a message with k tokens, the expected value of t_1, $\hat{E}(t_1) = \int_0^\infty (1 - P(t_1 \leq t))dt = \int_0^\infty Q(k, \frac{t}{d})^d dt$ where $Q(k, u)$ is a regularized incomplete gamma function and d is the degree.*

Proof.

$$\begin{aligned} \hat{E}(t_1) &= \int_0^\infty (1 - P(t_1 \leq t)dt = \int_0^\infty (1 - (1 - P(t_1 > t))dt \\ &= \int_0^\infty P(X_j(t) < k)^d dt = \int_0^\infty Q(k, \frac{t}{d})^d dt \end{aligned} \tag{5.1}$$

Table 5.1: Summary of the notations used.

Symbol	Definition
t_1	Time between initiation and creation of first sender
T^*	Total diffusion time
$\hat{E}(t_1)$	Expected t_1 (obtained analytically)
$\hat{E}(T^*)$	Expected T^*(obtained analytically)
$Av(t_1)$	Expected t_1 (obtained empirically)
$Av(T^*)$	Expected T^* (obtained empirically)
s_1	Limiting random variable of $t_1 d^{-\frac{k-1}{k}}$ as $d \to \infty$
τ_i	Scaled time span between creation of $(i-1)^{th}$ sender and i^{th} sender
s^*	Limiting random variable for $T^* d^{-\frac{k-1}{k}}$

where $Q(k,u)$ is a regularized incomplete gamma function i.e.

$$Q(k,u) = \frac{\Gamma(k,u)}{\Gamma(k)} = e^{-u} \sum_{l=0}^{k-1} \frac{u^l}{l!} \tag{5.2}$$

is valid for any natural k and non-negative u.

□

Note that t_1 for a Poisson process, is a continuous random variable.

Residual phase: We next proceed to establish the relation between $\hat{E}(t_1)$ and $\hat{E}(T^*)$. Apart from assuming the continuous model, we compute scaled t_1 and T^* (by $d^{\frac{k-1}{k}}$, (this specific scaling function was initially calculated for $k = 2$ and then generalized for higher values)) instead of their explicit values to further aid our analysis. To summarize, we start by computing expected values of scaled t_1 and T^* considering a continuous (Poisson) model with $d \to \infty$ and show that the results hold for finite d. Finally we show that the results for the continuous model extend to the discrete model. We begin by showing that t_1 (time to create the first sender apart from the initiator) is an indicator for the diffusion delay T^* through the following two theorems.

Theorem 2. *For a message with k tokens the random variable $t_1 d^{-\frac{k-1}{k}}$ converges as $d \to \infty$ to a limiting random variable s_1 with density $\frac{x^{k-1}}{(k-1)!} e^{-\frac{x^k}{k!}}$ and expectation $\hat{E}(s_1) = (k!)^{\frac{1}{k}} \Gamma\left(1 + \frac{1}{k}\right)$.*

Proof. The proof is based on Poisson clock approximation approach introduced previously. We start from the calculation of the CDF for the random variable $t_1 d^{-\frac{k-1}{k}}$:

$$
\begin{aligned}
F_{t_1 d^{-\frac{k-1}{k}}}(x) &= P(t_1 d^{-\frac{k-1}{k}} \leq x) \\
&= 1 - Q(k, \frac{x d^{\frac{k-1}{k}}}{d})^d \\
&= 1 - \left(\sum_{i=0}^{k-1} e^{-x d^{-\frac{1}{k}}} (x d^{-\frac{1}{k}})^i / i! \right)^d
\end{aligned}
\tag{5.3}
$$

where in the last line we used the common simple approximation of Poisson cumulative distribution in long tail.

Since we are interested in limits as $d \to \infty$, for $exp(-x d^{-\frac{1}{k}}) \to 1$ and we can compute limits of $F_{t_1 d^{-\frac{k-1}{k}}}$ as follows:

$$
\begin{aligned}
F_{s_1}(x) &= \lim_{d \to \infty} F_{t_1 d^{-\frac{k-1}{k}}}(x) \\
&= \lim_{d \to \infty} 1 - \left(\sum_{i=0}^{k-1} e^{-x d^{-\frac{1}{k}}} (x d^{-\frac{1}{k}})^i / i! \right)^d \\
&= 1 - e^{(-x^k/k!)}
\end{aligned}
\tag{5.4}
$$

Now, the density function of τ_1 can be calculated as -

$$
f_{s_1}(x) = \frac{dF_{s_1}}{dx}(x) = \frac{x^{k-1}}{(k-1)!} exp\left(-\frac{x^k}{k!} \right) \tag{5.5}
$$

Further the expectation of s_1 is -

$$
\begin{aligned}
\hat{E}(s_1) &= \int_0^\infty x f_{s_1}(x) dx = \int_0^\infty (uk!)^{\frac{1}{k}} e^{-u} du \\
&= (k!)^{\frac{1}{k}} \int_0^\infty u^{\frac{1}{k}} e^{-u} du = (k!)^{\frac{1}{k}} \Gamma\left(1 + \frac{1}{k}\right)
\end{aligned}
\tag{5.6}
$$

□

We next compute the expectation of the (scaled) time T^* till all nodes become senders. We consider $T_i = t_i d^{-\frac{k-1}{k}}$. We further assume τ_i as the scaled time span between the creation of $(i-1)^{\text{th}}$ new sender and that of i^{th} new sender. Correspondingly τ_i^* represents the scaled

time for the original process where every node with at least k tokens acts as a sender node. We have $\hat{E}(\tau_i^*) = \frac{1}{i}\hat{E}(\tau_i) = \frac{1}{i}\left(\hat{E}(T_i) - \hat{E}(T_{i-1})\right)$. Note that τ_1 is equal to T_1 as T_0 is 0 and hence $\hat{E}(s_1)$ equals $\hat{E}(\tau_1)$.

Theorem 3. *$T^* d^{-\frac{k-1}{k}}$ converges to a limiting random variable s^* with*

$$\hat{E}(s^*) = \sum_{i=1}^{\infty} \hat{E}(\tau_i^*) = \hat{E}(\tau_1)\frac{k}{k-1}$$

Proof. $G_i^{(d)}(z) = 1 - F_i^{(d)}(z) = \Pr\left\{t_i > zd^{\frac{k-1}{k}}\right\} = \Pr\{T_i > z\}$(complementary cdf of T_i). Since $\frac{zd^{\frac{k-1}{k}}}{d} = zd^{-\frac{1}{k}}$ we have

$$\begin{aligned} G_i^{(d)}(z) =& \sum_{j=0}^{i-1}\binom{d}{j}\left(\sum_{l=0}^{k-1}\frac{1}{l!}\left(zd^{-\frac{1}{k}}\right)^l e^{-zd^{-\frac{1}{k}}}\right)^{d-j} \\ & \left(1 - \sum_{l=0}^{k-1}\frac{1}{l!}\left(zd^{-\frac{1}{k}}\right)^l e^{-zd^{-\frac{1}{k}}}\right)^{j} \\ =& \sum_{j=0}^{i-1}\frac{1}{j!}\left(\frac{z^k}{k!}\right)^j e^{-\frac{z^k}{k!}}\left(1 + o_d(1)\right) \end{aligned} \tag{5.7}$$

taking limits $d \to \infty$ and using the abbreviation $a = \frac{z^k}{k!}$ we obtain for $G_i(z) = \lim_{d\to\infty} G_i^{(d)}$ and

$$G_i(z) = \sum_{j=0}^{i-1}\frac{a^j}{j!}e^{-a} \tag{5.8}$$

Subsequently, we get for $\hat{E}(\tau_i) = \int (G_i(z) - G_{i-1}(z))\,dz$:

$$\hat{E}(\tau_i) = \int_0^{\infty}\frac{1}{(i-1)!}\left(\frac{z^k}{k!}\right)^{i-1} e^{-\frac{z^k}{k!}}dz, \tag{5.9}$$

$$\hat{E}(\tau_i^*) = \int_0^{\infty}\frac{1}{i}\frac{1}{(i-1)!}\left(\frac{z^k}{k!}\right)^{i-1} e^{-\frac{z^k}{k!}}dz. \tag{5.10}$$

For computing $\sum_{i=1}^{N}\hat{E}(\tau_i^*)$ we can exchange integration and summation. Hence we first estimate

$$\lim_{N\to\infty}\sum_{i=1}^{N}\frac{1}{i!}a^{(i-1)}e^{-a} = \frac{1}{a}\left(1 - e^{-a}\right) \tag{5.11}$$

Transforming variables in the integral as $y = \frac{z^k}{k!}$ we finally get

$$\hat{E}\left(s^*\right) = \frac{(k!)^{\frac{1}{k}}}{k-1}\Gamma\left(\frac{1}{k}\right) = \hat{E}(\tau_1)\frac{k}{k-1} \tag{5.12}$$

□

We observe from the above result that the expectation of scaled T^* converges to a value which depends only on k which is constant for a given setting. Hence we conclude that the expectation of T^* **is proportional to** $d^{\frac{k-1}{k}}$ and similarly for t_1.

The above results are based on the assumption that i is fixed as $d \to \infty$. We now proceed to show that the computations hold for finite d for i varying with d. Note that the above formulas hold true for $i \leq f\left(d\right)$ as long as $f\left(d\right) = o\left(d^{\frac{1}{k}}\right)$.

Theorem 4. *Considering that the range of i (number of senders) varies with d (degree) if $f\left(d\right) = d^{\frac{1}{k}-\epsilon}$ for some $\epsilon > 0$, $\sum_{i>f(d)}^{d} \tau_i^*\left(d\right) = o_d\left(\sum_{i\geq 1}^{f(d)} \tau_i^*\left(d\right)\right)$*

Proof. We compute first $\mathbb{E}\left(\tau_i\right) = \int_0^\infty \frac{1}{(i-1)!}\left(\frac{z^k}{k!}\right)^{i-1} e^{-\frac{z^k}{k!}} dz$ using again the transformation of variables $y = \frac{z^k}{k!}$

$$\begin{aligned}
\hat{E}\left(\tau_i\right) &= \frac{1}{(i-1)!}\frac{(k!)^{1/k}}{k}\int_0^\infty y^{i-2+\frac{1}{k}} e^{-y} dy & (5.13)\\
&= \frac{1}{(i-1)!}\frac{(k!)^{1/k}}{k}\Gamma\left(i-1+1/k\right) & (5.14)
\end{aligned}$$

Using Stirlings approximation we obtain -

$$\frac{\Gamma\left(i-1+1/k\right)}{(i-1)!} \simeq e^{-2+\frac{2}{k}}\frac{1}{(i-1)^{1-1/k}} \tag{5.15}$$

The further argumentation is independent of the involved constant coefficients since we only need leading orders.

We have -

$$\begin{aligned} T_L &= \sum_1^L \hat{E}(\tau_i) = O(1) \cdot \sum_1^L \frac{\Gamma(i-1+1/k)}{(i-1)!} \\ &= O(1) \int_1^L \frac{1}{x^{1-1/k}} dx = O(1) L^{1/k}. \end{aligned} \tag{5.16}$$

Note once more that all the computations up to now are in scaled time units. Hence in real time we have $t_L \sim L^{1/k} d^{1-\frac{1}{k}}$. Setting $L \sim d^{\frac{1}{k}-\epsilon}$ we get $t_L = d^{1-\frac{1}{k}+1/k^2-\frac{\epsilon}{k}}$. Taking t_L as unit and taking into account that the total time t^* (in the scaled process) is according to the results in Erdös and Kaplan [110] $t^* = (1+o(1)) d \log d$ we have $t^* = O(1) \cdot t_L \cdot d^{\frac{1}{k}-1/k^2+\frac{\epsilon}{k}} \log d$. But since the acceleration at this point is $d^{1/k-\epsilon}$ we have for the remaining time (that is the time after the $d^{\frac{1}{k}-\epsilon}$-th event) in the accelerated process a contribution of at most $\tilde{t}_L d^{-1/k^2+\frac{\epsilon}{k}+\epsilon} \cdot \log d = \tilde{t}_L \cdot o_d(1)$ since ϵ can be chosen arbitrary small - here $\tilde{t}_L$ denotes the time till the L^{th} event in the accelerated process. This shows that $\sum_{i>f(d)}^d \tau_i^*(d) = o_d\left(\sum_{i\geq 1}^{f(d)} \tau_i^*(d)\right)$.

□

The above results show that previous computations (considering $d \to \infty$) give correct limiting values. This indicates that our analysis is able to correctly estimate the diffusion time for a complete graph of finite size.

It now remains to show that the asymptotic estimations for the model with Poisson clock carry over to the discrete time model (which we use for our simulations) defined at the beginning. Note that the discrete time model is actually the Poisson model when looked at in event time steps, where events here are the times when a token is sent.

Theorem 5. *If t_i is the time between the creation of the $(i-1)^{th}$ and i^{th} sender and $\hat{t}_i$ is the corresponding time in the discrete model, then $\hat{E}(\hat{t}_i) = \hat{E}(t_i)(1+o_d(1))$*

Proof. Since we have i independent senders all acting with Poisson clocks of intensity 1 we have the time between two tokens sent - denoted in the following by a random variable x - to be an exponential distribution $Exp(i)$. We index the events by l and observe that $t_i = \sum_{l=1}^{K_i} x_l$ where K_i is the random stop time when the i^{th} sender is created. In the discrete

model i messages are sent simultaneously hence i successive events in the Poisson model correspond to one time step in the discrete model. Hence $\hat{t}_i = \left\lfloor \frac{1}{i} \cdot K_i \right\rfloor = \frac{1}{i} \cdot K_i \cdot (1 + o_d(1))$. Since the $\{x_l\}$s are i.i.ds we can apply Wald's theorem [227] and get

$$\hat{E}(t_i) = \hat{E}(K_i)\,\hat{E}(x) = \tfrac{1}{i}\hat{E}(K_i)$$

hence $\hat{E}\left(\hat{t}_i\right) = \hat{E}(t_i)(1 + o_d(1))$ and the analytical results for the poisson model hold for the discrete case as well. □

We further simulated our diffusion model on complete graphs to verify our analytical results. For this purpose, we plot in figure 5.3 the values of average diffusion time ($Av(T^*)$) and average time to create the first sender ($Av(t_1)$) respectively as we vary the size of the network. We further report the values of $Av(T^*)$ and $Av(t_1)$ for different values of k with network size fixed at 1000. Note that the two quantities $Av(T^*)$ and $Av(t_1)$ (the results were averaged over 1000 simulations) exhibit a very similar profile irrespective of the chosen value of k. In the same figure we also plot the function $d^{\frac{k-1}{k}}$ (represented by $\hat{E}(t_1)$) obtained from theorem 2, suitably scaled by a constant to show how the theoretical results closely follow the numerical simulations.

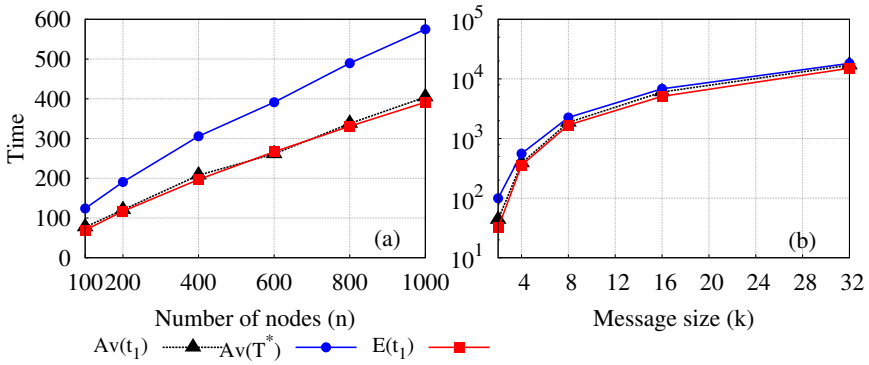


Figure 5.3: $Av(T^*)$ and $Av(t_1)$ versus (a) the number of nodes with message size $k = 4$ and (b) k for fixed $d = 1000$. For both the plots $\hat{E}(t_1) = C * d^{\frac{k-1}{k}}$ where $C = (k!)^{\frac{1}{k}}\Gamma\left(1 + \frac{1}{k}\right)$ (refer to theorem 2).

E-R random graph: We further look into **Erdos-Renyi random graphs** [64] and observe that for sufficiently dense graphs (i.e., having high edge probability) the analysis on

the complete graph case holds. In this regard we first plot $Av(T^*)$ (obtained through simulations) and $\hat{E}(T^*)$ ($n^{\frac{k-1}{k}}$ scaled by a constant, n is the number of nodes) for different values of k (message size) (refer to figure 5.4(a)). Clearly the theoretical estimate closely follows the simulated result. As we increase the value of edge probabilities(p) (i.e., make the network more dense) the closer it gets to the theoretical estimate. We further plot $Av(T^*)$ (scaled by $\hat{E}(t_1)$) for varying p in figure 5.4(b). The value gets close to 2 with edge probability 1 but remains close to 2 even for lower values of p. All the results are averaged over 1000 simulations.

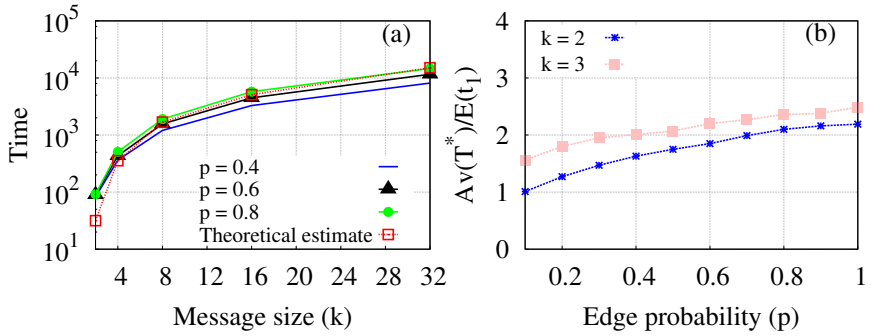


Figure 5.4: (a) $Av(T^*)$ and $\hat{E}(T^*)$ (suitably scaled) for different values of k (b)$Av(T^*)$ versus edge probability in Erdos-Renyi random graph for $k = 2$ and $k = 3$. In both cases $Av(T^*)$ is normalized by $\hat{E}(t_1)$ which is $\sqrt{n}$ and $n^{\frac{2}{3}}$ for $k = 2$ and $k = 3$ respectively.

5.4 On d-regular tree:

We next consider the case of d-**regular trees** for $d \geq 3$ (at least 2 children apart from 1 parent) with a distinguished root index 0 which acts as the initial sender. For simplicity we give the root an out-degree of $(d-1)$ by attaching a virtual "mother vertex" to the root which is also a sender but has only one offspring and is not counted in the estimation of sender nodes (this helps us avoid handling the initial steps (i.e., when only the root is having the message) differently from the later steps). Let $A_l(t)$, $0 \leq l < k$, be the number of nodes on the tree which have exactly l packets at time t and have a direct communication link to one of the sender nodes at time t. Note that each of the so defined

nodes has exactly one connection to a sender node due to the tree structure and the initial condition of having just one sender at the beginning. We get the following exact linear recursion for the expectation $a_l(t+1) := \hat{E}(A_l(t+1))$ at time $t+1$:

$$\begin{aligned} a_l(t+1) &= \frac{d-1}{d} a_l(t) + \frac{1}{d} a_{l-1}(t), 1 \leq l \leq k-1 \\ a_0(t+1) &= \frac{d-1}{d} a_{k-1}(t) + \frac{d-1}{d} a_0(t) \end{aligned}$$

Note that for the expected number of sender nodes s_t at time t, we have

$$s_t = \sum_{t'<t} \frac{1}{d} a_{k-1}(t') \tag{5.17}$$

The asymptotic rate of growth of the variables $\{a_i(t)\}$ as well as s_t is entirely determined by the value of the largest eigenvalue of the associated transition matrix. The maximal eigenvalue of the associated characteristic polynomial is given by

$$\lambda_{\max} = \frac{d-1}{d} + \left(\frac{d-1}{d} \left(\frac{1}{d} \right)^{k-1} \right)^{\frac{1}{k}} = \frac{d-1+(d-1)^{1/k}}{d}$$

In figure 6.1(b) we draw the diffusion dynamics for a 5-regular tree and in the same figure we show that the analytical estimate (obtained from equation 17) of diffusion rate closely resemble the empirical observation.

5.5 Innoculation strategy

We have defined the epidemic setting aptly fitting the temporal network system whereby agents remember interactions from the previous time-steps and only adopt an idea after encountering it multiple times. We find that such information diffusion process undergoes two phases with a slow initial phase followed by a very fast residual phase. This behavior is observed irrespective of the underlying topology (refer to figure 6.1, simulations done for scale-free graphs [17] and ER-random graphs). The reason behind such behavior is that during the process when the first few nodes get fully infected a large fraction of nodes

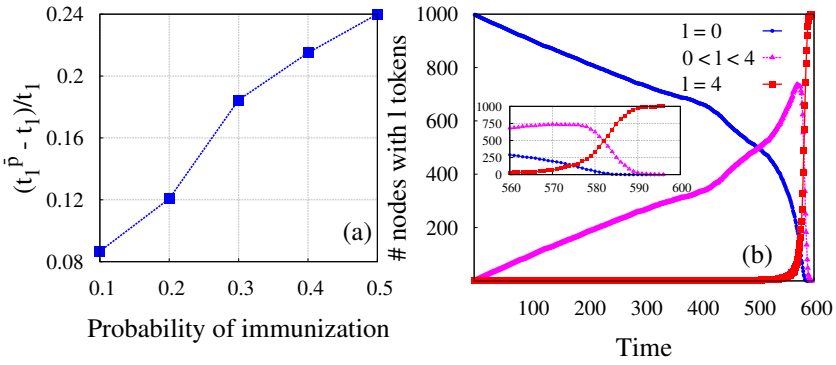


Figure 5.5: (a) $\frac{t_1^{\bar{p}} - t_1}{t_1}$ versus $\bar{p}$ for a complete graph with 1000 nodes and $k = 4$. (b) Number of nodes at each stage of infection versus time for complete graph of 1000 nodes with $k = 4$. Although the creation of infected nodes is slow initially, the number of partially infected nodes ($0 < l < 4$) increases rapidly. (inset) Magnified version of the same figure.

also simulataneously get partly infected as represented in figure 5.5(b). Once the first set of infected nodes are created these partially infected nodes also get quickly infected and this results in a sudden ramp-up in the diffusion rate.

The above observations indicate that for such systems spreading could be controlled/contained while the system is still in the slow initial phase. Accordingly, we perform an empirical study, whereby, we reduce the level of infection of the agents at one particular time step (say t) and then estimate the time required to obtain the first sender vis-a-vis the time t_1 when such action is not initiated. Reduction of infection means removing one packet from the chosen (say with probability $\bar{p}$) agent, for example, if an agent has acquired j packets at time t, we reduce it to j - 1. In specific we considered a complete graph with 1000 nodes and at $t = 0.5 * t_1$ we reduce the level of infection in each node with a probability ($\bar{p}$) and measure the corresponding time (say $t_1^{\bar{p}}$). In figure 5.5(a) we plot $\frac{t_1^{\bar{p}} - t_1}{t_1}$ for different values of $\bar{p}$. We observe that creation of the first sender (other than initiator) i.e., beginning of the residual phase (where the rate of spread is increased manifold) could be delayed by almost 20% with a probability of 0.3. We show the experiment by inoculating at one particular time step, however a continuous low-grade innoculation strategy can be initiated and we believe a threshold can be derived whereby the first sender creation can be pushed to infinity. However that can be an interesting research direction to be pursued in the future.

5.6 Broadcast algorithm based on the diffusion model

We now proceed to propose a broadcast algorithm based on the information diffusion model proposed earlier. Note that contrary to the general assumption that size of a message is small enough to be transfered from one node to other on the short durations of contacts between the nodes, we consider a more stringent case that is, the duration of a contact between the nodes is not always sufficient for the transfer and therefore the message might need to be segmented/divided into sub-parts and sent individually. Further note that our proposed diffusion model is a natural fit (which we elaborate on next) as the number of contacts to contract infection can be replaced by the number of sub-parts in the message.

5.6.1 Agent configuration and network setup

We consider a network topology $G =< V, E >$ where each node in V represents an agent of the network and any link in E represents a contact opportunity between a pair of nodes (agents) in the whole time span through which the network is active. So for any node (agent) n_i in this network, its one hop neighbors are the nodes (agents) which are within the connection proximity of n_i and at each time step n_i at random can connect to any one of them.

5.6.2 Message configuration

We consider that a message $\mathcal{M}$ is divided into a set of m packets, i.e., $|\mathcal{M}| = m$. Packets in $\mathcal{M}$ are grouped into a set $\mathcal{S}$ of s segments, where each segment consists of $k = m/s$ packets (i.e., s can take only those values for which $m \bmod s = 0$). For instance, if $\mathcal{M}$ constitutes of 4 packets and 2 segments then each of the segment is composed of 2 packets. Note that in this paper we mostly consider that there is only a single segment in the message ($m = k$) unless specified otherwise.

5.6.3 Transfer protocol

In this framework, transfer of a message during a contact refers to the transfer of one single packet of any segment of a message. Transfer of a packet from u_i to v_j during a contact can take place only when u_i qualifies as a *sender* by having all the packets of at least one of the message segments. The two basic modes of message transfer that we consider are the push and the (restricted) pull epidemic.

Push technique:

- *Step 1:* At any time step, u_i (already a sender) establishes a communication link with v_j, from its neighborhood and finds an exclusive set of packets that u_i has but v_j does not have in its buffer.
- *Step 2:* If u_i can find such a (non-empty) set, then it transfers only one packet from this set to v_j.

Pull technique:

- *Step 1:* At any time step between v_j and u_i, v_j establishes a communication link with u_i and requests for a packet that it has not received yet.
- *Step 2:* Given that u_i has already become a sender it first finds such a packet from its own buffer and then transfers a copy of it to v_j.

Note that in the traditional pull technique, u_i, being a sender, may transfer more than one packet if it gets multiple simultaneous requests from more than one agent at any particular time step. However, unlike the traditional case, here u_i can serve only one request in case there are multiple pull requests. Such a restriction actually allows for conservation of both battery power and network bandwidth in resource constrained scenarios like DTN. Also note that we consider the duration of each communication link is long enough for the transfer of at least a single packet.

5.6.4 Metrics of interest

We are interested to evaluate the performance of a broadcast protocol in terms of two different metrics. The first metric concerns broadcast delay. The second metric centers around a complementary issue of power and bandwidth consumption.

- **Broadcast delay** T^* - this is the time from the point when the message source starts sending the first packet to the point when all the agents in the network have received the entire message. $E(T^*)$ denotes the expected broadcast time. In addition, we are also interested in the time T_i which is the minimum time at which there are i senders (except the source) in the network, and especially in T_1 since, as we shall see, that this is the prime determinant of the entire broadcast time.
- **Broadcast wastage** C_m^* - let C_l and C_p be the total number of communication links that get established in the network and the total number of successful packet transfers respectively. We define the broadcast wastage as $C_m^* = (C_l - C_p)/C_l$. This metric essentially measures the number of useless contacts in the network during which no packet could be transferred and is therefore a direct cause of energy wastage.

5.6.5 Broadcast algorithms

We consider following four algorithms for broadcasting messages. While the first one (Blind Push) is exactly similar to the diffusion model proposed earlier, the others builds on it.

5.6.5.1 Blind push (B-P)

An initiator node is the one which has the full message in the beginning. At each time step all the nodes in the system having the full message communicate with a node in their proximity and try to $push$. At the end of each time step all the nodes which have received all the packets qualify as sender in the next time step. The algorithm terminates when all the nodes in the system have the full message.

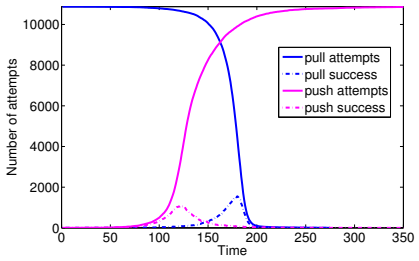


Figure 5.6: Pull attempts, successful pulls, push attempts, successful pushes versus time for gnutella1 network

5.6.5.2 Blind pull

At each time step all the nodes in the system not having the full message, communicate with a node in their proximity and try to $pull$. At the end of each time step the nodes which have received the full message stops pulling from the next time step. The algorithm terminates when all the nodes in the system have the full message.

5.6.5.3 Strategy-x%, automatic switch from push to pull

In this Strategy (X-P-P), the nodes in the system follow $Blind - push$ initially and then switch to $Blind - Pull$ once x% of the nodes in the system have the full message. The algorithm terminates when all the nodes in the system have the full message.

We consider the $Blind - push$ and $Blind - pull$ algorithm on a network and check how efficiently they perform over a broadcast time window. In figure 5.6 we plot the number of attempts and the number of successful ones for the above two cases on gnutella1 network (described later in this section). This clearly shows pull mechanism performs poorly in the beginning but picks up after a certain percentage of nodes have become senders. We observe just the opposite behavior for push mechanism. Our X-P-P strategy is based on this idea to obtain the best out of both the strategies.

The X-P-P strategy cannot be implemented in a practical setting because the nodes in the system need to maintain a global information of the percentage of nodes in the system having the full message which is difficult in a distributed setting like this.

5.6.5.4 A distributed version of X-P-P strategy

Here, we introduce a new strategy $Push - pull - with - giveup$ (P-P-G) which approximately mimics the X-P-P strategy in a distributed setting and it functions in the following way- Initially there is a single node in the system which has the full message. At each time step the sender nodes in the system communicate with one of the nodes in their proximity and try to $push$. Among all the other non-sender nodes in the system those which have at least a single packet (i.e., nodes which have participated in a message transfer at least once and hence are aware of the broadcast) try to pull. Each node also keeps a local history regarding the number of unsuccessful communications it has participated in and once this exceeds a threshold, it 'gives-up' and no longer participates in the broadcast. Once all the nodes have 'given-up', the broadcast terminates.

5.6.6 Dynamic topology

We performed our experiments on gnutella snapshots and on synthetic topologies like complete graph, regular tree, regular graph and random graph. A topology specifies the potential neighborhood of a node - a node at each time step connects randomly to one of these nodes. A complete graph topology would indicate that the node can connect to any other node in the network while for other sparser topology it would connect only to a subset of them.

5.7 Experiment on different network topologies

In this section we systematically study the effect of topology of the underlying contact network on the broadcast time and wastage and come up with some suggestions which we feel will be helpful while designing networks. We first analyze B-P algorithm on different topologies like regular graph, regular tree and random graph. In particular, we wish to check whether the average degree (d) of the underlying contact network influences the performance metrics. In the later part of this section we make a comparative study

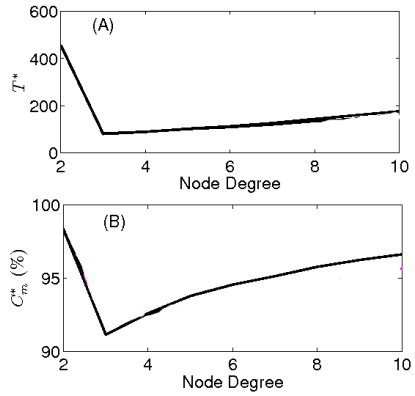


Figure 5.7: (A) Broadcast time and (B) broadcast wastage versus different values of d for B-P. The parameters values are $n = 200, m = 4, k = 2$.

of different broadcast strategies (discussed in section 5.6) and also reinspect into their sparser variants to identify the effect of lowering the value of d through removal of edges without hampering network connectivity.

5.7.1 Blind push on different topologies

The results of B-P on complete graph are similar to the ones provided in section 5.3, hence we concentrate on the sparser topologies. In specific, we empirically analyze the B-P algorithm on regular-tree, regular-graph and random-graph. For each topology we consider $n = 200$, $m = 4$ and $k = 2$. Note that here we consider that the message has 2 segments and each segment has $k = 2$ packets. We then vary the average degree d for each of these networks and check how broadcast delay and wastage depend on it. Remarkably, for each of these topologies - regular tree (figure 5.7), regular graph (figure 5.8) and random graph (figure 5.9), one can observe that there is a critical value of d for which we obtain minimum broadcast delay and wastage.

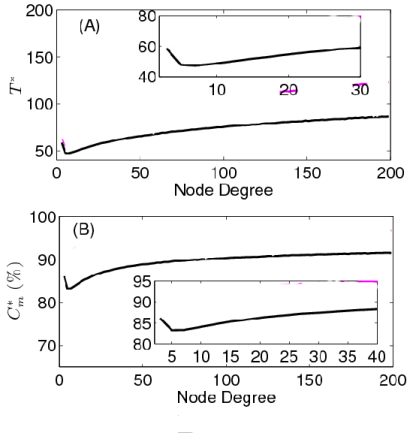


Figure 5.8: (A) Broadcast time and (B) broadcast wastage versus different values of d for B-P technique. The parameters values are $n = 200, m = 4, k = 2$. The inset in both the figures show the metrics of interest for the first few values of d to indicate the critical d more appropriately.

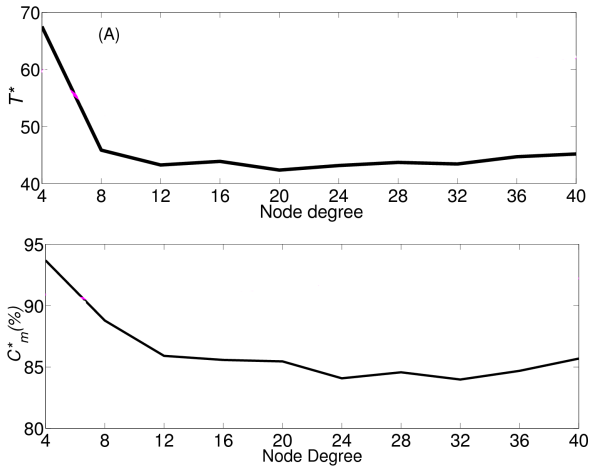


Figure 5.9: (A) Broadcast time and (B) broadcast wastage versus average degree for B-P . The parameters values are $n = 200, m = 4, k = 2$.

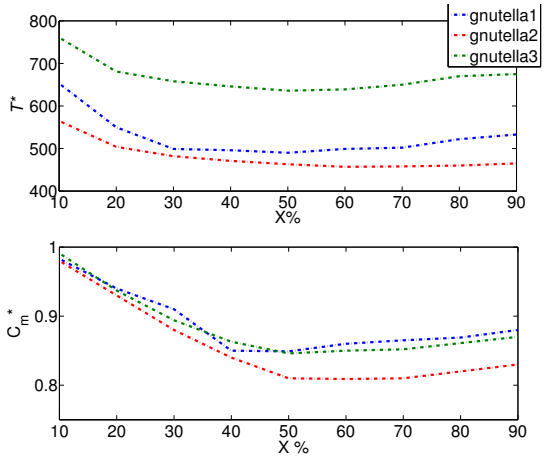


Figure 5.10: Average broadcast time and wastage versus x for gnutella1,gnutella2 and gnutella3

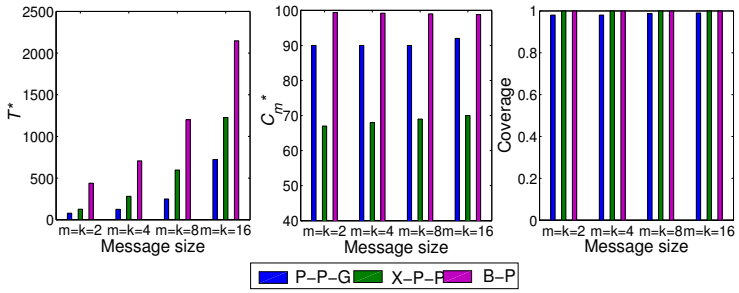


Figure 5.11: (A) Broadcast time versus message size (B) Wastage versus message size (C) Coverage versus message size for gnutella3

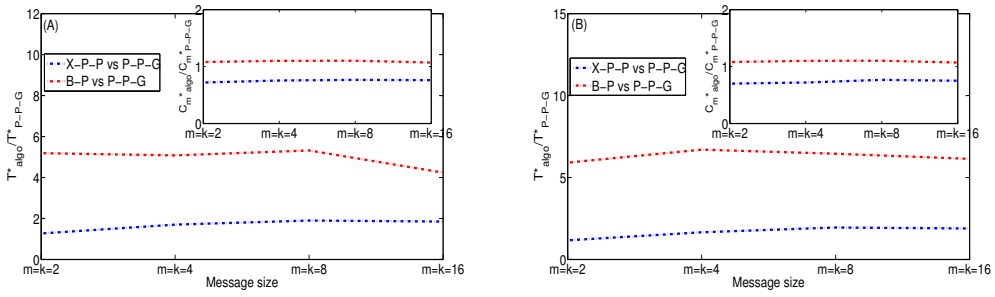


Figure 5.12: Gain in broadcast time of P-P-G over X-P-P and B-P [Inset shows gain in wastage] for (A)gnutella1 and (B)gnutella2 networks. Note: algo = B-P/X-P-P

5.7.2 Comparison of different broadcast strategies on Gnutella Topology

We measure the performance of the algorithms (B-P, X-P-P and P-P-G) on three real network traces based on broadcast time, wastage and coverage. The data sets are Gnutella network snapshots namely p2p-Gnutella04 (gnutella1), p2p-Gnutella06 (gnutella2) and p2p-Gnutella25 (gnutella3) [135, 191] taken on dates August 4, August 6 and August 25, 2002 respectively. The gnutella1 network has 10876 nodes and 39994 edges in its largest connected component. Corresponding number of nodes and edges in the largest connected component in gnutella2 and gnutella3 are respectively 8717, 31525 and 22663, 54693. We simulate these algorithms for varying message sizes. We perform our simulations on peer-to-peer systems specifically as our study can find a major application in such systems.

For simulating the X-P-P algorithm in particular we first need to obtain the best value of x for each network and then run the simulations for varying message sizes. In figure 5.10, we show how the broadcast time and wastage varies with x for networks gnutella1, gnutella2 and gnutella3. We observe that the best value of x lies around 50% for the gnutella1 network. Similarly the obtained value of x are found it to be around 50% and 60% for gnutella2 and gnutella3 respectively. In figure 5.11 we plot the broadcast time, wastage and coverage for gnutella3 network. We observe that with P-P-G we gain in both broadcast time and wastage with respect to B-P. With respect to X-P-P, P-P-G offers better broadcast time but with a higher wastage. For gnutella1 and gnutella2 networks we plot (figure 5.12) the ratio of broadcast time and wastage of X-P-P and B-P over P-P-G for different message sizes. We observe that across different message sizes, on an average the gain in broadcast time of X-P-P over P-P-G and B-P over P-P-G are 5.45 and 1.5 respectively for gnutella1 network while for gnutella2 network the corresponding values are 5.9 and 1.6 respectively. Corresponding values for wastage are 0.75 and 1.10 respectively for gnutella1 network and 0.72 and 1.06 respectively for gnutella2 network. So with P-P-G we gain in both broadcast time and wastage with respect to B-P while with respect to X-P-P we gain in broadcast time without significant increase in wastage. Actually, X-P-P provides the best optimization between broadcast time and wastage but it would be hard to implement in a distributed setting. Our proposed algorithm (P-P-G) presents the best possible trade-off of delay and wastage guaranteeing almost 100% coverage and can be implemented in a

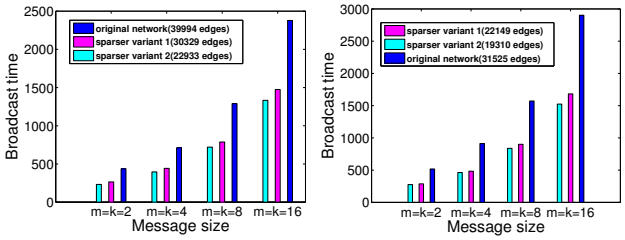


Figure 5.13: Broadcast time for the gnutella snapshots and their sparser variants versus different values of message sizes for (a). gnutella1 and (b). gnutella2
distributed fashion with almost negligible computational overhead.

5.7.2.1 Effect of Degree on Broadcast time

In earlier part of this section we observed that irrespective of the topology one is able to a obtain critical value of d for which the broadcast time is minimum. So we performed simulations on sparser variants of these gnutella networks and observed that broadcast time reduces even for the B-P. To obtain sparser variants, we considered each gnutella snapshot and randomly removed some of the edges without hampering the network connectivity. From figure 5.13 we observe that the broadcast time reduces significantly in case of the sparser variants in comparison to the original network. Hence, while designing a network it is advisable to keep the network sparse rather than creating unnecessary connections between the nodes. This, as our results indicate, should lead to a faster dissemination of messages.

5.8 Summary of the chapter

Our contributions in this chapter can be summarized as -

- We proposed a threshold based diffusion model whereby a susceptible individual contracts infection only after multiple contacts with infected individuals motivated from the idea that an individual adopts an idea after encountering it multiple times.
- Irrespective of the topology, the diffusion seems to progress in two phases - (i) initial

phase where the diffusion rate is slow followed by (ii) residual phase where the diffusion rate increases manifold.

- Through detailed empirical evaluations we showed that for these systems spreading could be controlled if acted upon during the initial phase. In fact we believe that our findings could open up paths to a number of future studies towards regulating contagion processes in systems with memory
- Based on our diffusion model we could further propose a set of message broadcast algorithms in dynamic networks. In fact we could come up with a practical strategy which can optimize the dual (conflicting) objective of speed and wastage.
- We explored the impact of the problem in different topologies and surprisingly noticed that mere dense topology is not of much help in a dynamic setting.

Chapter 6

Application of temporal network analysis in peer-review network

leverage temporal network analysis techniques to improve peer-review system.

6.1 Introduction

Peer-review system has been relied upon by the scientific community for determining the correctness and the quality of the findings presented in a research article. The authenticity and, hence, the need for this process has long been debated since in many cases flawed research has got into the literature even though the peer-review process was rigorous [30]. Similarly, there have been cases where excellent research was misjudged by the peer-review process and therefore rejected [33]. The publishing house makes significant investments into ensuring the quality of editing and reviewing of the received submissions and, therefore, identifying the necessity of this entire system is of prime importance.

Debates on the scientific peer-review: The effectiveness of peer-review have been studied to a large extent in the domain of medical sciences where peer-review is heavily relied upon for judging the quality of a research article [105, 111, 189]. The effect of blinding on the quality of peer review has also been studied in detail [106, 149]. It was observed

that blinding improves the quality of reviews. Several limitations of the review process have also been pointed out [99]. In [51] the authors show that there is a high degree of disagreement within the population of eligible reviewers. [33] also shows that there are a significant number of papers that receive more citations after rejection. All these together point to limitations of the review process and have resulted in the scientific community questioning the requirement of this process.

A massive peer-review dataset: In this paper, we investigate the effectiveness of the peer-review system through a rigorous and large-scale analysis of the scientific review data. In particular, we consider a set of around $29k$ papers along with roughly $70k$ unique review reports containing $12m$ lines of review text submitted to the Journal of High Energy Physics (JHEP) between 1997 and 2015. We would like to point out here that this dataset is unique as well as very rich and we do not know of any other work that presents such a large-scale analytics of an equivalent dataset. Informed with the details of the number of reviews per paper, the content of the review reports and the citation counts we perform, for the first time, a series of systematic measurements to determine whether the peer-review process is indeed able to correctly differentiate between high impact contributions and the rest.

Citation impact of accepted papers: Assuming that citation count of a paper is representative of its overall quality, we observe that on average those papers which were accepted at JHEP after passing through the peer-review process, are cited more often compared to those which got rejected at JHEP and eventually got accepted at a different venue. While this is true for the majority, there are a few exception cases where either a rejected paper is found to receive high citations or an accepted paper is found to receive (almost) no citation.

Reviewer-reviewer interaction network: One of the central contributions of this work is the introduction of a novel reviewer-reviewer interaction network built as the one-mode projection of the editor-reviewer bipartite network. The reviewer-reviewer interaction network has nodes as the reviewers and two reviewers are connected by an edge if they have been assigned by the same editor. Surprisingly, the network related structural features such as the degree, the clustering coefficient and the centrality values (closeness, betweenness etc.) of the reviewer nodes in the reviewer-reviewer network strongly correlate with the long-term citations received by the papers these reviewers refereed.

Supporting features: Another unique contribution of this paper is that we also build a set of supporting features based on the various characteristics of the papers submitted as well as the authors and the referees of the submitted papers. *Papers*: The highly cited papers tend to undergo lesser rounds of review and there also exists an optimal team size (number of contributing authors) for which the accrued citation is maximum. *Review Reports*: For the accepted papers, the length of the review reports seem to be indicator of the long-term citation. Moreover there exists an optimal length for which the citation obtained by the corresponding paper is maximum. On performing sentiment analysis, we observe the review reports to be mostly neutral as the referees hardly use highly polar words in their reports. However, we observe several linguistic quality indicators which can be extracted from the review text that determines whether a paper is going to be cited well in the future. *Authors*: From the author specific analysis we observe that for authors who have a higher acceptance to submission ratio tend to receive more citations than others who have a lower acceptance to submission ratio. In addition, the reviews received by authors having higher acceptance to submission ratio tend to contain more positive sentiments on average. *Reviewers*: In a previous work [203] the authors showed that the reviewers who tend to accept or reject most of the papers assigned to them fail to correctly judge the quality of the papers. We include such history based features of the reviewers in the set of supporting features.

Two further interesting observations from the analysis of the supporting features are – the low cited accepted papers got in due to the higher accept history of the authors and the lenience of the referees; the high cited rejected papers could not make a place because of lower accept history of the authors and the strictness of the referees.

Before the contributions of a paper are brought to the notice of the research community, it has to usually pass through a peer-review process, whereby, the correctness and the novelty of the paper is judged by a set of knowledgeable peers. The primary intent of which is to prevent flawed research from getting into mainstream literature [111].

Debates on peer-review system: The effectiveness of this system has been put to question in numerous cases ([104, 187, 204]) with flawed research being added to literature while significantly novel contributions being rejected. That the reviewers often fail to reach consensus ([51]) and that rejected papers are often cited more in the long run ([33]), have already been pointed out. Although there have been several proposals to make it more

effective ([36, 82, 149]), the research community is coming to a conclusion that although peer-review system is indispensable it is nonetheless flawed [15].

Entities in the peer-review system: The effectiveness of the peer-review system is dependent directly on the knowledge and training of the editors and reviewers. The editor is responsible for identifying the correct set of referees who can give expert comments on the submission and also for taking the final decision whether a particular paper should be accepted or rejected. The assisting reviewers send their views on the paper in the form of a report. This report is an important part of the whole process as it not only forms the basis of the acceptance/rejection decision but is also sent to the authors for further improvement of the paper.

Anomalous behavior: Ideally impactful papers should be accepted for publication while flawed works should be rejected. We quantify the impact of a paper by the citations it garnered. Thus, a paper getting accepted but managing to garner very less or no citation should be attributed to the anomaly of the system; similarly, a paper getting rejected by the peer-review-system but garnering large number of citations in the long run is also an anomaly. We in this paper investigate the reasons behind the anomalous behaviors ([40]) of the reviewers and editors as they are the most important entities of the peer-review system. Note that although the number of such anomalous editors or referees might be small compared to the number of normal editors or reviewers (as is usually the case with any anomalous set), the damage they can cause to the peer-review system could be irreparable and therefore a thorough investigation of this set is extremely necessary.

Characterizing anomalous editors and reviewers: A thorough investigation of the behavior of the **editors** shows that those editors who (i) are assigned papers more frequently, (ii) select reviewers from a very small set, (iii) assign themselves as reviewers more often (rather than assigning other reviewers) are often under-performers and hence anomalous. Similarly, for **reviewers** we observe the following behaviors to be anomalous - (i) frequent assignments, (ii) very small or very large delay in sending reports, (iii) reviewing papers in very specific topics, (iv) assignments from a very small set of editors or in some cases a single editor, (v) very high or very low proportion of acceptance, (vi) large delay in informing the editor about inability to review and (vii) often declining to review. Papers accepted by reviewers with such behaviors are often low cited while those rejected by

them are often highly cited.

Identifying anomalous editors and reviewers: All the above observations lead us to believe that anomalous editors and reviewers can be differentiated from the genuine contributors. To this aim we use these observations as features and by leveraging anomaly detection techniques we are indeed able to filter out the anomalous editors and reviewers. In specific we use k-means clustering [90] to classify normal and anomalous editors and reviewers. We find 26.8% of the editors and 14.5% of the reviewers to be anomalous. We further observe that the papers accepted by these anomalous reviewers are on average cited less while those rejected by them are cited more.

The scientific community relies heavily on the peer-review system to judge the quality of a new research contribution. In this process a set of peers or reviewers are handed the responsibility of judging whether the work is flawed and should be discarded or is relevant enough to be brought to the notice of the research community. Since the reviewers are the most important entities of the entire peer-review process, their knowledge and training are highly critical to the proper functioning of the review process. In fact in may cases when more than one referees are involved in reviewing a paper, lack of consensus among them might make it difficult for the editor to judge the true quality of the work, which then might lead to a severe mistake.

Single or multiple reviewers: So a natural query arises: whether peer-reviewing comprising multiple referees should be preferred over a single referee system? To answer this question, we in this paper for the first time, analyze the peer-review information of all the papers that were submitted to two leading physics journals together consisting of approximately $36k$ papers with about $19m$ lines of review texts. An exploratory analysis of citation information of the papers reveals that papers reviewed by multiple referees, on acceptance tend to be cited less (on average) while on rejection tend to be cited more (on average) compared to the papers which get reviewed by a single referee. However, papers reviewed by multiple reviewers constitute the majority of the most cited (top 25%) papers. In fact, the observations are consistent across both these datasets. The dichotomy in this observations raises a natural question that why multiple refereeing does not work well on average. We hypothesize that this is mainly due to lack of consensus among the referees in the multi-reviewer system.

Lack of consensus among reviewers: In [51] the authors demonstrated that in multi-refereed papers the referees often fail to reach consensus. As an immediate next step following the previous observations, we investigate the review reports of the multi-refereed papers. Leveraging several natural language processing (NLP) tools we establish the lack of consensus among reviewers for such papers. In fact, we observe that in terms of report length, sentiment and content, the referees differ in almost 30% of the cases on average across the two datasets.

Analyzing multi-referee behavior: Further analysis of the peer-review system reveals that the performance of a reviewer can be quantified by his/her (a). frequency of assignment and (b). tendency to be too critical (tends to reject most of the papers assigned to them) or too liberal (tends to accept majority of assignments). The discordance also occurs when such reviewers are grouped together, which perhaps leads to acceptance of paper without due diligence. In contrast, we find that even when under-performing reviewers are grouped with well-performing reviewers, the overall quality of acceptance improves. Remarkably, we also observe, that the most under-performing groups have the highest lack of consensus, which further corroborates our hypothesis.

Recommending reviewer groups: From the above observations we hypothesize that multi-referee systems mostly fail due to lack of proper selection and the assignment of the referees. We in this paper propose a systematic scheme for recommending reviewer groups to the editor. We argue that the problem of assigning multiple referees to a paper is similar to the problem of forming compatible groups from a population, which has already been studied in great detail in the context of collaborative learning. In fact, genetic algorithm (GA) based frameworks have been shown to be very effective in such a setting [10, 153]. We hence propose a GA based framework which, given a paper, its topic and a reviewer pool with past information, is able to recommend a set of groups of compatible referees to assist the editor in assignment of referees. We observe that in cases where the reviews led to acceptance and the paper garnered a large number of citations, our algorithm is able to correctly identify almost 78% of the group of referees involved on average across the two datasets.

Importance of editor intervention: The above results might give a false impression that given a set of topics and reviewer history, our system can recommend reviewer groups

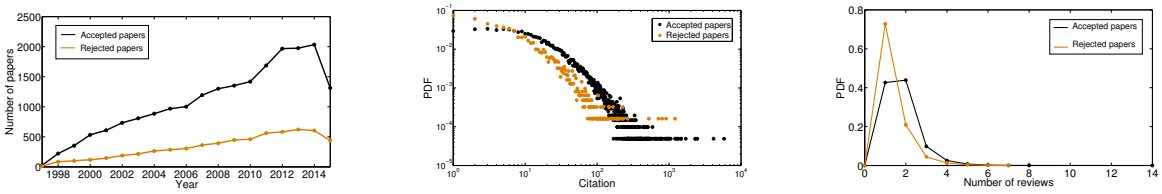


Figure 6.1: (Left) Number of accepted and rejected papers per year from 1997 to 2015. **(Middle)** Citation distribution of both the accepted and the rejected papers. **(Right)** Distribution of number of reviews for accepted and rejected papers.

without the expert intervention of the editor. Using a carefully designed simulation setup, we show that intervention of the editor in selecting a reviewer group from the set of recommended groups is highly critical to the proper functioning of the system in long term. In fact we show that the ability of correctly identifying the reviewer groups reduces to almost 45% **(from** 78%**)** if the editor is not involved in the peer-review process.

Determining the fate of the paper: Based on the network features built above, we propose a supervised model which quite accurately predicts (R^2 = **0.79**, $RMSE$ = **0.496**) the long-term citation of a paper. In addition, if we also include the supporting features into the model we obtain further gains (R^2 = **0.81**, $RMSE$ = **0.46**). Analysis of the importance of the features shows that the network features are the strongest predictors for this task. We believe that our system would be of immense help in assisting the editor in deciding acceptance or rejection of the paper specifically in cases when the review reports are contradictory. Note that while our work is a case study of JHEP, the formulations that we report are very general and can be extended to any other available dataset.

6.2 Dataset

The main aim of this work is to understand the importance of the review process and especially the contribution of the referees in this process. Such an analysis requires detailed information of reviews like the number of rounds of review, final decisions and the text of the review report in each round and, lastly, the number of citations for each paper.

Obtained data: As stated earlier, for our analysis, we consider the dataset of the *Journal of High Energy Physics* (JHEP). It is one of the leading journals in its field and publishes

theoretical, experimental and phenomenological papers. Among JHEP's most direct competitors there are Physical Review D, Nuclear Physics, EPJC, Physics Letters B and Physical Review Letters. This dataset consists of **28871** papers that were submitted between 1997 (year of inception) and 2015 of which **20384** were accepted and **7073** were rejected. The rest of the papers were either withdrawn by the authors or the final decisions were not available. The number of distinct review reports is **70k** containing **12m** lines of review text. For each paper we have the title, the abstract, the authors, the date of publication (in case it was accepted) and the number of citations for the accepted papers. The dataset further contains for each paper the number of rounds of reviews it received before it was accepted (rejected) as well as the detailed text of the report submitted by the assigned reviewer and the editor.

Pre-processing: To obtain the necessary information for the rejected papers we queried the "Inspire" search engine[1] by their corresponding arXiv[2] ids. We could obtain for each paper the citation information, the abstract, the title, the authors and also the publishing journal (if at all it got published). Note that all through our analysis we refer to number of citations as the cumulative number of citations that a paper/author obtained at the end of 2015. We further had to disambiguate the names of the authors and assign each of them a unique id.

Some basic facts about the dataset: In fig. 6.1**(Left)** we plot the year-wise distribution of the accepted and the rejected papers from 1997 to 2015. We observe an increasing trend in the number of submissions except for the year 2015 for which the data is incomplete.

Table 6.1: General information of the dataset.

Number of papers	28871
Number of papers (accepted)	20384
Number of papers (rejected)	7073
Average number of reviews (accepted papers)	1.76
Average number of reviews (rejected papers)	1.35
Average number of citations (accepted papers)	31.89
Average number of citations (rejected papers)	9.45

In fig. 6.1**(Middle)** we plot the citation distribution of the accepted and the rejected papers. Both the distributions seem to follow a power-law behavior. We further plot the distribution

[1] `https://inspirehep.net`
[2] `http://arxiv.org/`

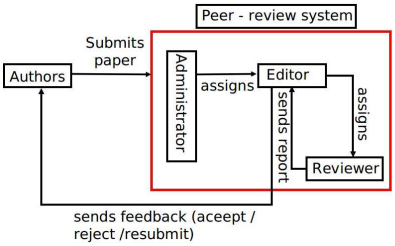


Figure 6.2: Peer-review system in JHEP

of the number of reviews for the accepted and the rejected papers in fig. 6.1**(Right)**. An important observation is that for JHEP, majority of the papers undergo one or two rounds of reviews after which they are either accepted or rejected. We note certain general information related to the dataset in table 6.1. Each submitted paper in the dataset also consists of the list of authors. There are **15127** unique authors in the dataset with at least one submission to JHEP and **12434** authors with at least one accepted paper. The average number of submissions per author is **5.18** while the number of authors per paper is **2.87**.

Peer-review process in JHEP: For every submitted paper the administrator assigns an editor for it. The editor selects a single or a small set of reviewers for judging the quality of the contributions in the paper. The reviewer sends back his views in the form of a report. Based on this report the editor decides whether to accept or reject the paper. The editor may also ask the authors to reshape the paper based on the feedback of the reviewer(s) and in which case they have to resubmit before a decision on its acceptance could be taken. In fig. 6.2, we present a schematic showing the peer-review process in JHEP.

6.3 Reviewer-reviewer interaction network

In this section, we construct a reviewer-reviewer interaction network and show that its properties are linked to the future scientific impact of a paper (measured in terms of the cumulative citation count). In specific, we find that the position of the assigned reviewer in the network (measured in terms of degree, centrality, clustering coefficient and PageRank) could be used to predict the long term citation of the paper.

The reviewer-reviewer interaction network is created with each node representing a

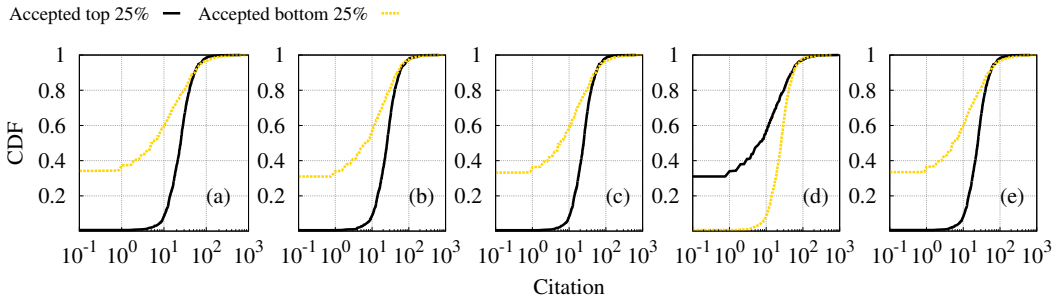


Figure 6.3: Cumulative distribution function (CDF) of citations received by the papers (accepted) reviewed by referees in top 25% and bottom 25% reviewers ranked according to (a) degree, (b) betweenness centrality, (c) closeness centrality (d) clustering coefficient values and (e) PageRank in the reviewer-reviewer interaction network.

reviewer and an edge exists between two reviewers if they have been assigned by at least one common editor. We devote the rest of the section in demonstrating the importance of the various structural properties of the network in determining the long term citation of the paper. Note that there are 4035 unique reviewers in the system each of which form a node in this network.

6.3.1 Degree (Deg)

Degree of a node v is the number of other nodes it is connected to in the network. A node with a higher degree in the reviewer-reviewer interaction network would indicate (i) assignment from multiple editors, (ii) assignment from a reputed editor (with large number of assignments) which in turn would indicate the reputation of the reviewer. To verify our hypothesis, we rank the reviewers based on their degree in the network and calculate the mean citation of the papers reviewed by the reviewers in the top and the bottom 25% of the rank list. We observe that the papers reviewed by the top 25% reviewers receive much higher citations than the those reviewed by the bottom 25% reviewers (refer to fig. 6.3(a)).

6.3.2 Betweenness centrality (BC)

Betweenness centrality of a node quantifies the position of a node based on the number of shortest paths the node is part of. For every pair of nodes in the network there exists a shortest path between them. Betweenness centrality of a node (v) is the fraction of all such paths that pass through v. In the reviewer-reviewer interaction network, a high centrality value would indicate assignment by multiple editors and that this node acts as a bridge between them. We again rank the reviewers based on the betweenness centrality values and calculate the average citation of the papers. We find that the papers accepted by the top 25% reviewers tend to be cited more compared to those accepted by the bottom 25% (refer to fig. 6.3(b)).

6.3.3 Closeness centrality (CC)

Formally closeness centrality of a node in a network is the inverse of the sum of length of its shortest path to all other nodes in the network. Hence higher centrality value indicates that the node is more closer to all other nodes in the network. In the reviewer-reviewer interaction network, a reputed reviewer will be assigned by multiple reviewers and hence will be closer to the other reviewers in the network. This is represented in fig. 6.3(c), where we show that the papers accepted by top 25% most central reviewers are cited more often compared to the bottom 25% reviewers.

6.3.4 Clustering coefficient (Clus)

Clustering coefficient of a node is measured as the fraction of connections among the neighbors of the node. For the reviewer-reviewer interaction network, every reviewer assigned by a common editor is connected to every other reviewer in the network. A reviewer assigned by many editors would actually act as a bridge between two cliques and hence would have a lower clustering coefficient value compared to a reviewer who is part of a single clique (always assigned by a single editor). This is further demonstrated in fig. 6.3(d) where we observe that the papers accepted by reviewers having lower clustering

coefficient tend to be cited more.

6.3.5 PageRank (PR)

PageRank is a link analysis based algorithm that calculates for each node its relative importance within the network. Specifically, PageRank outputs a probability distribution which is used as the likelihood of a random walker to end up in a specific node. We simulate PageRank on the reviewer-reviewer interaction network to obtain the relative importance of each node. Further analysis indicates that the papers accepted by the top 25% reviewers (based on PageRank) are cited more often compared to those accepted by the bottom 25% reviewers (refer to fig. 6.3(e)).

The above results thus indicate that simple network properties of the reviewer-reviewer interaction network could be highly effective in predicting the long-term citation of the paper at the time of publishing.

6.4 Supporting features

Most of the works [39, 240] related to predicting long-term citation of papers considers a wide set of author related features, papers-centric features and citation pattern of the paper in the first few years from the date of its publication. Further, our dataset allows us (unlike the existing datasets) to look into several other features related to the review-process like the review report, the behavior of the assigned referee, the number of rounds of reviews the paper went through and others. We consider the above features as well as those existing in the previous literature (wherever available) as the set of supporting features. The features are categorized into (i) paper based, (ii) review report based, (iii) author based and (iv) reviewer based. Apart from investigating the effectiveness of a feature in determining the long-term citation of the paper, we also point out some interesting observations that we could make while analyzing the dataset.

6.4.1 Paper based features

We have already observed that the average number of citations received by accepted papers is 31.80; for rejected papers the corresponding value is 9.45 (refer to table 6.1). Note that for each paper we consider the citations accrued by it till 2015 from the date of publication. We further consider all the accepted and the rejected papers and segregate them based on the number of citations received. We consider different citation buckets and plot the fraction of accepted and rejected papers in each of these buckets in fig. 6.4**(Left)**. Note that the bucket sizes are in increasing powers of 2. Typically, the buckets are ≤ 1, 2, $(> 2$ and $\leq 4)$ and so on. It can be clearly observed that accepted papers are cited more often compared to the rejected ones. Nevertheless, on further analysis we find that there could be a few exception cases where the rejected paper could make a place in higher impact journals (compared to JHEP) and, eventually, receive a high volume of citations in future. We present two such pathological cases below – **Case 1:** Rejected after two rounds of review, later accepted at Physics Letters B, citations: 1209; **Case 2:** Rejected after one round of review, later accepted at Computer Physics Communications, citations: 929. We perform a thorough analysis of these irregular cases later in the paper.

6.4.1.1 Number of review rounds (RR)

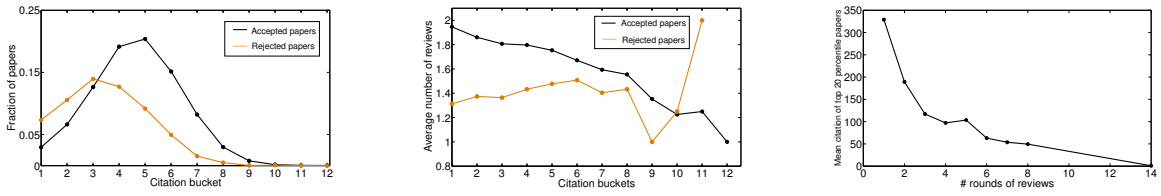


Figure 6.4: (Left) Fraction of accepted and rejected papers in different citation buckets. **(Middle)** Average number of reviews for accepted and rejected papers in different citation buckets. For both **(Left)** and **(Middle)** bucket sizes are in increasing powers of 2. E.g. ≤ 1, 2, $(> 2$ and $\leq 4)$ and so on. **(Right)** Average citation of the top 20 percentile papers for a given number of rounds of review request.

We next check whether the **review rounds** improve the quality of the paper. To this aim we segregate the papers based on the number of citations into different buckets and for each bucket we calculate the average number of reviews the papers received in that bucket. The

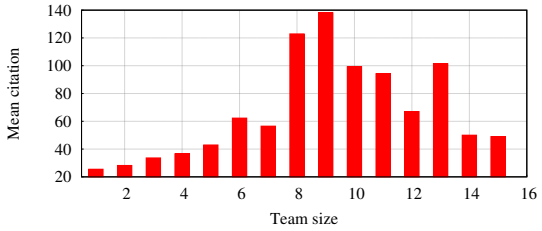


Figure 6.5: Average citation versus team size. Note that we segregate the papers based on the teams size and calculate the average citation.

bucket sizes are again in increasing powers of 2. Typically, the buckets are ≤ 1, 2, $(> 2$ and $\leq 4)$ and so on. We plot the results in fig. 6.4**(Middle)**. We observe that for accepted papers, the low cited ones on average tend to get accepted after more rounds of reviews while the high cited ones undergo lesser rounds of reviews before getting accepted. It can hence be concluded that going through the review process multiple times does not necessarily improve the quality of the papers much that are eventually accepted. For the rejected papers we observe a contrasting trend indicating that the review process indeed helped in improving the quality of the paper in the long run possibly enhancing the chances of its acceptance at a different venue later. For the accepted papers we further classify them based on the number of reviews they received and calculate the average citations of the top 20 percentile papers (ranked by citations) in each class (refer to figure 6.4**(Right)**). We observe that the average citation drops as the number of reviews increases further suggesting that papers accepted after higher number of reviews often fail to create a high citation impact. This indicates that the number of rounds of review could be an indicator of the long-term citation of the paper.

6.4.1.2 Team size (TS)

The authors in [39] hypothesized that there exits an optimal team size for which the citations received by the paper is maximum. We hence segregate the papers based on the team size and calculate the mean citation of the papers. We observe that team size 9 (refer to fig. 6.5) is the optimal as the papers with 9 authors gets more citation on average.

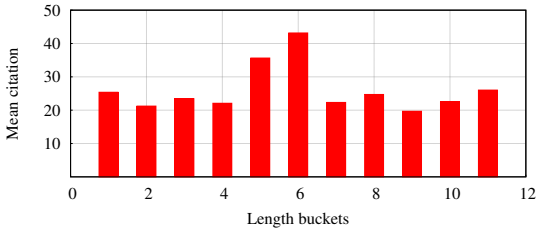


Figure 6.6: Mean length of referee reports in terms of number of words at different rounds of review. Typically the buckets are < 100, $(\geq 100, < 200)$ and so on

6.4.2 Review report based features

In this subsection we analyze whether certain features could be extracted from the reports sent by the reviewers that could be an indicator of the long-term citation of the paper. Note that we have two types of reports – **Referee report**: Report sent by the assigned referee to the editors and **Editor report**: Report sent by the editor to the authors based on the referee report. We primarily focus on the referee reports as editorial reports are in almost all cases a reiteration of the referee reports.

6.4.2.1 Length of the reports (RL)

We start by looking whether the length of the review reports sent by the reviewers are indicative of the quality and hence the long-term citation of the paper. To this aim we segregate the papers based on the length of the report and calculate the mean citation of each of these buckets. The lengths are bucketed with sizes typically < 100, $(\geq 100, < 200)$ and so on. We observe that there exists an optimal length (between 500 and 600 words) for which the citation obtained by the corresponding paper is maximum (refer to figure 6.6).

6.4.2.2 Sentiments (SNT)

We next perform sentiment analysis on the review reports. To determine the sentiment of a report we use a method described in [152] which performs a graph-based word sense disambiguation and lexical similarity analysis using a pre-existing knowledge base. A sentiment score of 0 indicates that the document is neutral, a positive score indicates a

Table 6.2: Mean values of percentages of various categories of words in review reports of high and low cited papers where the means differ significantly.

Category	Dimension	High cited papers	Low cited papers
Linguistic	Future tense	1.17	1.05
	Negation	0.72	0.84
Cognitive	Insight	3.52	3.16
	Causation	2.60	2.38
	Inclusive	3.70	3.43
	Exclusive	1.28	1.52
Affective	Positive emotion	2.84	2.70

positive sentiment and a negative score indicates a negative sentiment.

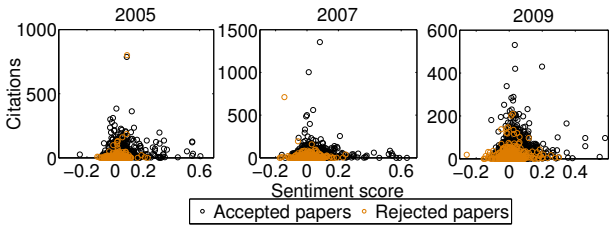


Figure 6.7: Sentiment score versus citations for both accepted and rejected papers for the years 2005, 2007 and 2009. We find similar trends for other years as well.

To determine whether the overall sentiment of the reviews of a paper is related to the number of citations received by it, we plot for each paper (both accepted and rejected) the number of citations it received against the sentiment score of the first round of review in fig. 6.7. Note that we segregate the papers based on the year of the publication or the rejection. We observe that the highly cited papers mostly have reviews with neutral or positive sentiment. Accepted papers with positive reviews are, on average, found to receive 25.27 citations while those with negative reviews are found to receive 13.56 citations. However, there are cases where the accepted paper received highly positive reviews but was not cited. Conversely, there are cases where the sentiment was neutral but the paper garnered a large number of citations. Shown below are two such examples – **Case 1:** Year of publication: 2006, sentiment score: 0.0234 (almost neutral), citations: 5812; **Case 2:** Year of publication: 2008, sentiment score: 0.65 (highly positive), citations: 6.

For the rejected papers we observe that those which received neutral reviews but were rejected, tend to garner higher citations later compared to the ones which received negative reviews. There are certain exceptions as well, two of which are – **Case 1:** Year of rejection: 2010, sentiment score: 0.27 (positive), citations: 1; **Case 2:** Year of rejection: 2007,

sentiment score: -0.14 (negative), citations: 711. Manual investigation of the review text shows that the papers which are highly cited after rejection were mainly rejected for not being in the scope of JHEP and not because of flawed results.

6.4.2.3 Linguistic quality indicators (LQI)

Here we check whether there are linguistic quality features present in the review reports which can serve as an indicator of the future impact of the paper. To our aim we use the LIWC[3] (Linguistic Inquiry and Word Count) text analysis tool [175]. The tool provides, as output, percentage of words in different categories for an input text. The categories are broadly divided into linguistic (21 dimensions like pronouns, articles etc.), psychological (41 dimensions like affect, cognition etc.), personal concern (6 dimensions), informal language markers and punctuation apart from some general features like word count, words per sentence etc. We apply the LIWC tool on the review reports for our analysis and mainly focus on the linguistic and the psychological categories. Next we check whether the LIWC features discussed earlier can also serve as indicators differentiating high and low cited papers. We rank the papers based on the number of citations they have received and consider the top 10% as highly cited and the bottom 10% as low cited papers. Note that we only consider the papers that were published before 2012 so that the papers have at least three years of citation history. In table 6.2 we report the mean percentage of words in different LIWC categories across all the papers (both high and low cited). We find several quality indicators here as well. The key observations are: (i) future tense is used more significantly in case of review reports of highly cited papers compared to low cited papers. On manually investigating the reviews of some of the highly cited papers we observe that statements like "its result will become a useful addition to .." are prevalent; (ii) insightful and inclusive words are also used to a greater extent in review reports of highly cited papers compared to low cited papers; (iii) positive words are also more prevalent in highly cited papers as well. Thus, these indicators show that the reviewers were, in many cases, indeed able to guess the quality of the paper as is evident from the review reports.

[3] http://liwc.wpengine.com/

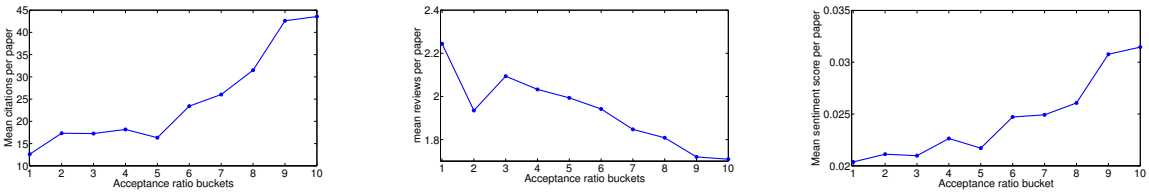


Figure 6.8: (Left) Mean number of citations per paper versus acceptance ratio. **(Middle)** Mean number of reviews per paper versus acceptance ratio. **(Right)** Mean sentiment score per paper versus acceptance ratio. Note that in each case we use acceptance ratio buckets where buckets correspond to acceptance ratio (≥ 0.1 and < 0.2), (≥ 0.2 and < 0.3) and so on.

6.4.3 Author based features

We next look into some of the author based features like author reputation and author productivity to determine how they influence the long-term citation of the paper.

6.4.3.1 Author reputation (AR)

We analyze whether there are some specific authors whose papers always get accepted and similarly there are others whose papers always get rejected. For each author we define a metric called **acceptance ratio** which is the fraction of submitted papers accepted in JHEP. Formally, acceptance ratio of an author i is defined by: $acceptance\,ratio_i = \frac{accept_i}{accept_i + reject_i}$. where $accept_i$ and $reject_i$ represents respectively the number of accepted and rejected papers of the author i in JHEP. We use this metric as a proxy for author reputation. We observe that mean acceptance ratio across all the authors is 0.56. In fact, for almost 7% of the authors, the acceptance ratio is 1. Next we check whether the authors with high acceptance ratio have higher citations per paper. To this aim we segregate authors based on the acceptance ratio and calculate the mean number of citations per paper for these authors (refer to fig. 6.8**(Left)**). We observe an increasing trend suggesting that the authors with higher acceptance ratio tend to have higher citations.

To check whether the authors having higher acceptance ratio are also reviewed less, we again segregate the authors based on the acceptance ratio and calculate the average number of reviews received per paper for these authors (refer to fig. 6.8**(Middle)**). We observe a decreasing trend implying that papers of authors with higher acceptance ratio are indeed

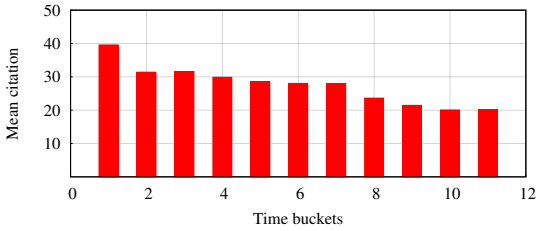


Figure 6.9: Mean citation of the papers versus the average time(in days) between two submission. Note that we use time buckets where buckets correspond to < 100,(≥ 100 and < 200), (≥ 200 and < 300) and so on.

Table 6.3: The F-statistics value for all the features used for predicting the long-term citation of the paper.

Feature	Deg	BC	CC	Clus	PR	RR	TS	RL	SNT	AR	AP	RAC	TA	DR
F-statistics	**26.1**	**29.21**	**27.72**	**17.82**	**23.34**	6.17	**25.6**	14.1	0.94	**18.52**	**16.49**	3.49	8.68	7.59

reviewed less. Although there are authors with high acceptance ratio whose papers are reviewed less, they are often highly cited indicating the overall effectiveness of the review process. We further study the sentiment score of the review reports for authors with different acceptance ratios. For authors in a given acceptance ratio bucket we calculate the average sentiment score of the review reports of their papers. We observe that the authors having higher acceptance ratios tend to have more positive reviews on average compared to the others with lower acceptance ratios which is indicated by the increasing trend in the curve in fig. 6.8(**Right**).

6.4.3.2 Author productivity (AP)

It is established in the literature [240] that the more papers an author publishes, more are his chances of getting cited. We hence use it as a feature in predicting the long-term citation of the paper. We calculate for each author the mean time (s_t) between two submissions. We use s_t as proxy for author productivity as low s_t would indicate higher productivity rate and vice versa. The papers are segregated based on the corresponding author's s_t and then the mean citation is calculated. Each bucket correspond to < 100,(≥ 100 and < 200), (≥ 200 and < 300) and so on. We observe that more frequent the submission, more is the chance of getting citation (refer to figure 6.9).

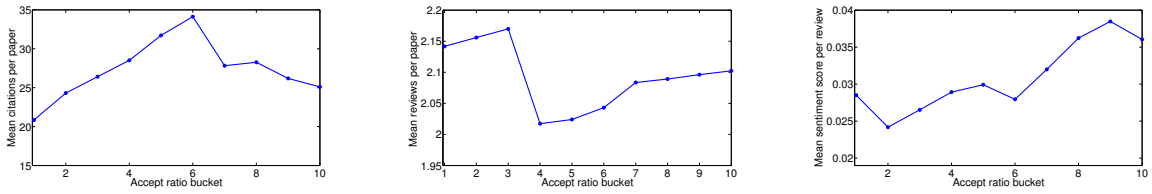


Figure 6.10: (Left) Mean number of citations per paper versus accept ratio. **(Middle)** Mean number of reviews per paper versus accept ratio. **(Right)** Mean sentiment score per paper versus accept ratio. Note that in each case we use accept ratio buckets where the buckets correspond to accept ratio (≥ 0.1 and < 0.2), (≥ 0.2 and < 0.3) and so on.

6.4.4 Reviewer based features

The success of the peer-review process is immensely dependent on the reviewers as they determine the quality of a paper and, consequently, the quality of the journal. We hence investigate certain reviewer behaviors (pointed out in [203]) that could be indicative of his/her performance.

6.4.4.1 Accept ratio (RAC)

For each reviewer we define a metric called accept ratio (this is different from the acceptance ratio defined in the previous section) Formally, the **accept ratio** for reviewer j is $accept\,ratio_j = \frac{accept_j}{accept_j + reject_j}$ where $accept_j$ and $reject_j$ respectively represent the number of papers reviewer j accepted and rejected. The mean accept ratio across all the reviewers is 0.62. An accept ratio of 1 for a reviewer would mean that he accepted all the papers that were assigned to him.

We start by investigating how well the reviewers were able to anticipate the quality of the paper. To this aim we segregate papers based on their assigned reviewer's accept ratio and calculate the mean number of citations received. In fig. 6.10**(Left)** we plot the number of citations per paper given the accept ratio of the assigned reviewer. We observe that most cited papers were reviewed by reviewers with accept ratio between 0.6 and 0.7. Surprisingly, the papers reviewed by reviewers with very low and very high accept ratio tend to garner less citations. This indicates that some papers could have been accepted just because the assigned reviewer is oriented to accept most of the papers. In fact, manual inspection indicates that many of the rejected papers that garnered large number of citations later

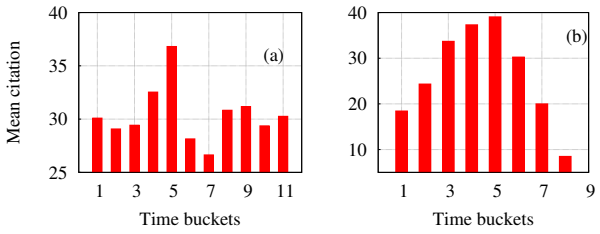


Figure 6.11: Mean citation of the papers versus (a) time since the last assignment for the assigned reviewer and (b) time taken by the reviewer to send the report. Note that for both the cases the times are divided into equi-sized buckets. For (a) bucket sizes are 100 each while for (b) it is 25.

on were mostly reviewed by reviewers with low accept ratio. We further investigate the number of reviews a reviewer with a given accept ratio suggests before he accepts or rejects a paper. For this we again segregate the papers based on the assigned reviewers accept ratio and calculate the number of rounds of reviews, these papers received on average. We present the results in fig. 6.10(**Middle**). Since in most of the cases the same reviewer is assigned in each round of review, we observe that reviewers with a low accept ratio tend to recommend more rounds of reviews while those with higher accept ratio tend to suggest lesser number of review rounds. This indicates that the reviewers with low accept ratio often fail to improve the quality of the paper as is evident from the mean number of citations these papers receive albeit dragging the paper through multiple rounds of reviews.

To complete the analysis we also investigate the average sentiment score of the papers based on the accept ratio of the assigned reviewers. In fig. 6.10(**Right**) we plot the average sentiment score of the papers with similar accept ratio of the assigned reviewer. We observe an increasing trend indicating that the reviewers with very high accept ratio always tend to give more positive reviews as compared to others with lower accept ratio.

6.4.4.2 Time since last assignment (TA)

In lines of [203] we consider the time (in number of days) since the last assignment for the assigned reviewer as an indicator of reviewer's performance and hence an indicator of the long-term citation of the paper. To verify our hypothesis we segregate the papers based on the assigned reviewer's time till last assignment and calculate the mean citation. The times

are bucketed with bucket size typically < 100, ($\geq 100, < 200$) and so on. We observe in fig. 6.11(a) that there does exist an optimal time for which the citation of the accepted paper is maximum. Further if the time since last assignment is too low or too high the long-term citation is low.

6.4.4.3 Delay in submitting the report (DR)

We further check whether the time taken by the assigned reviewer in submitting the report is also as an indicator for his performance and hence that of the long-term citation of the paper. To this aim we calculate for each paper the time between assigned reviewer acknowledging to review the paper and the reviewer sending back the report. The papers are segregated based on this time and then the mean citation is calculated. The times were binned with typical bucket sizes being > 25, ($\geq 25, < 50$) and so on. We observe from fig. 6.11(b) that the citation is maximum when the reviewer sent back the report between 50 and 75 days. The citations are comparatively less if the time is too high as well as too low.

6.5 Determining the fate of the paper

In this section we design a regression model to calculate the long-term citation impact of a paper. We perform our prediction for papers that were accepted in JHEP between 2007 and 2012. Note that for each paper, its citation till 2015 is available. Hence papers published in 2004 would have a higher citation on average compared to papers which were published in 2010 (say) due to higher exposure time. Thus, instead of calculating the exact citation value we predict the citation rank (for each year we rank the papers based on the citations they have accrued till 2015). Further note that the papers are sorted based on the date of submission and given a paper we construct the reviewer-reviewer interaction network until its submission date (excluding). This ensures that there is no data leakage. Similarly for supporting features like acceptance ratio of an author we consider information only up to his/her last submission.

Network features only: Considering only the network features, we obtain the best result

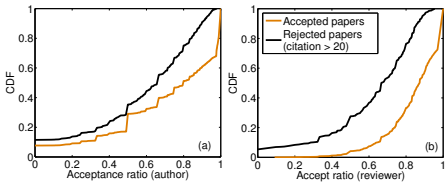


Figure 6.12: CDF of (a) acceptance ratio of authors and (b) accept ratio of reviewers for accepted papers and highly cited rejected papers.

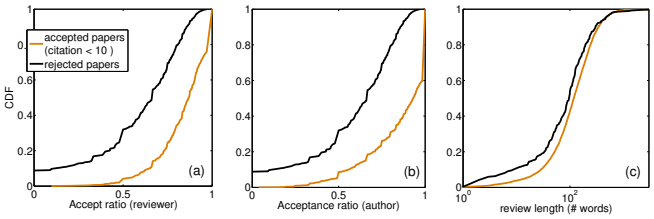


Figure 6.13: CDF of (a) accept ratio of reviewers, (b) acceptance ratio of authors (c) length of the review text (# words) for rejected papers and low cited accepted papers.

using support vector regression (RBF kernel) with parameters $C = 100$ and $\gamma = 0.01$. We perform a 10-fold cross-validation and obtain a high R^2 of **0.79** and a low $RMSE$ of **0.496**.

Network + supporting features: Considering both the network and the supporting features we obtain a further overall improvement. In specific, using support vector regression (RBF kernel) we obtain a high R^2 of **0.81** and a low $RMSE$ of **0.46**. The parameters were set as parameters $C = 100$ and $\gamma = 0.02$. We further calculate the F-Statistic values for all the features used in the regression task (refer to table 6.3) and observe that the network features, are in general, are more suited to the task of prediction.

Thus our system is correctly able to predict the citation rank of the paper. We believe our system could be useful in assisting the editors in deciding whether to accept or reject the papers especially in cases where the reports are contradictory.

6.6 Irregular cases

In this section we investigate in more detail the irregular cases, i.e., the highly cited rejected papers and the low cited accepted papers. Note that we only consider papers

which were published before 2012 so that each paper gets at least three years of exposure to the scientific community for garnering citations.

Highly cited rejected papers: We previously observed that on average the accepted papers tend to be cited more often than the rejected ones. Nevertheless, we find several papers (call it the set P) which were rejected at JHEP but were able to acquire more citations after getting accepted elsewhere. We consider only those papers in P that have at least 20 citations which is twice the average citation of the rejected papers. Manually looking into some of the review text we observed that in several cases the reviewer found the topic of research to be interesting but out of JHEP's scope. On deeper investigation we found that acceptance ratio of the authors of these papers is lower than that of the authors of accepted papers (fig. 6.12(a)) indicating that author's reputation might have played a role in the rejection. We further observe that the accept ratio of the reviewers, these papers were assigned to, to be significantly less than that of the accepted papers (fig. 6.12(b)). This indicates that the papers got assigned to stricter reviewers and hence the rejection.

Low cited accepted papers: We now look into the complementary i.e., the papers which were accepted but failed to make impact on the scientific community and accrued very low citations (typically < 10). Ideally, these papers should have been rejected, hence we investigate how different these are in terms of author's acceptance ratio and reviewer's accept ratio. While the authors of these papers have higher acceptance ratio (fig. 6.13(a)), they were also assigned to reviewers who are less strict (fig. 6.13(b)). These observations indicate that either the contributing author's reputation might have played a role in their acceptance or were lucky to have been reviewed by lenient referees. Review report also seem to be sloppy in many cases with the reviewer not even mentioning the reason for acceptance. We also observe the length of the review report (in terms of the number of words) on average to be less than that of the rejected papers (fig. 6.13(c)).

6.7 Publisher's views

We further requested the JHEP journal administrators to survey our findings. The publishers pointed out the following observations to be of great significance – (i) that reviewers

excessively accepting/rejecting often fail to judge correctly the quality of the article, could be useful in assigning referees; (ii) the number of review request does not necessarily improve the quality of the article. This observation could help in improving the efficiency of the peer-review process since both cost and time are involved for each round of review request; (iii) since only 10% of the submissions were assigned to multiple reviewers and in most cases they failed to reach consensus, the publishers felt the need to investigate this issue more deeply by having frequent multiple assignments henceforth; (iv) the framework for predicting the long-term impact of the paper would be extremely helpful in assisting the editors in taking decisions. This might also aid in tracking the performance of the referees; (v) that a significant fraction of authors have high acceptance rate at JHEP indicates a presence of at least a weak bias in the peer-review process and hence needs to be investigated;

6.8 Conclusion

In this paper, we provided a framework for predicting the long-term citation of the paper which can be extremely helpful in assisting the editors in deciding whether to accept, reject or opt for a third opinion. We demonstrated that very simple positional properties extracted from the reviewer-reviewer interaction network are exceedingly important in determining the long-term citation of the paper. In specific, if we plug in these features in a regression model, we obtain R^2 = **0.79** and $RMSE$ = **0.496** in predicting the long-term citation of a paper. In addition, we also introduce a set of supporting features, based on the various properties of the paper, the authors and the assigned referees which further improved the prediction (R^2 = **0.81** and $RMSE$ = **0.46**).

In the process of designing these features, we also made some key observations which are summarized below -
(i) the papers which went through lesser number of review rounds tend to be cited more on an average while the papers that were accepted after going through higher number of rounds are cited less on an average (although exceptions exist for both cases); (ii) although the reviewers tend to avoid highly polar words (negative or positive) in their review reports, the overall sentiment in the reports of accepted papers is more positive whereas the same is more negative for the rejected papers; (iii) the authors with higher acceptance

ratio tend to be cited more on an average compared to those with lower acceptance ratio (iv) reviewers excessively accepting or rejecting most of the assigned papers often fail to correctly judge the quality of the paper; (v) deeper investigation of the irregular cases revealed that the reputation of the author is often influential in the acceptance or rejection of the paper; Apart from being a large-scale study that attempts to provide quantitative evidences supporting its necessity, ours is the first work that proposes definitive ways of improving the effectiveness of the scientific peer-review system.

6.9 Anomalous behavior

In the peer-review process each submission is assigned to an editor who in turn assigns one or more reviewers with the task of judging the quality of the contributions of the submitted paper. The reviewer submits a report to the editor who in turn takes the final decision as to accept or reject the paper based on the report. Therefore, the editors and the reviewers are the two important entities of the peer-review system and they are mainly responsible for ensuring that flawed research does not get into the literature while at the same time correctly identify impactful contributions for publication. So in our setting we define the following two cases to be anomalous -
(i) Accepted papers having low citation (research wrongly judged as impactful).
(ii) Rejected papers having high citation (quality research wrongly judged as flawed).
In this section we look into the anomalous behavior of the two important entities of the peer-review process: (i) the editors and (ii) the reviewers.

6.9.1 Editor

We begin by analyzing the anomalous behavior of the editors. We define the behavior of an editor to be anomalous if the papers assigned to her are on average cited less when accepted or are cited more when rejected. In specific, we investigate different factors related to the editor that can lead to such anomaly.

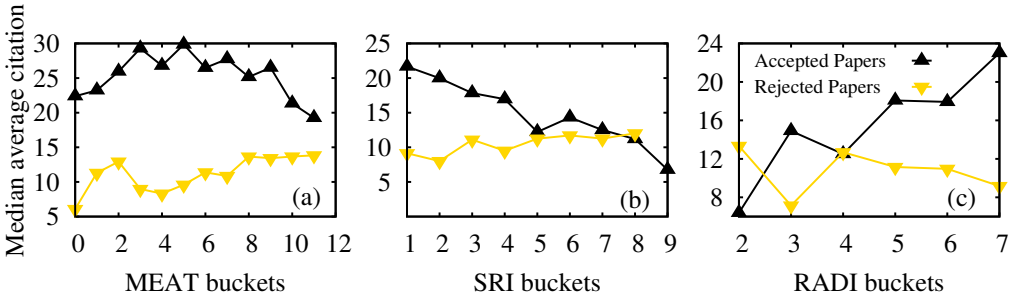


Figure 6.14: (a) Median Average citation (MAC) versus $MEAT$. $MEAT$ values are bucketed into 12 bins of equal size with range(1, 498.8).(b) MAC versus SRI and (c) MAC versus $RADI$. For both (b) and (c), the x-axis values are bucketed by values corresponding to (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on.

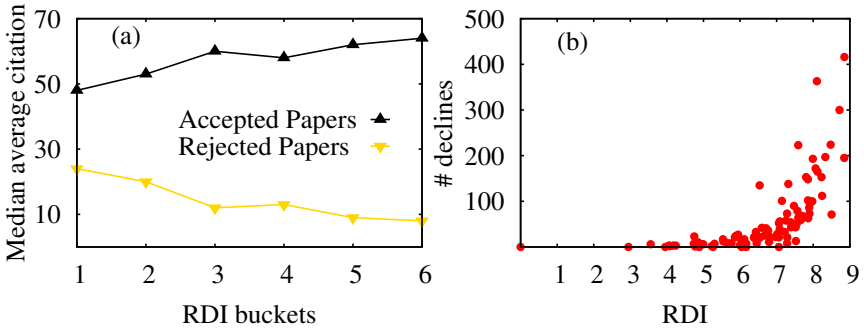


Figure 6.15: (a) Median Average citation versus SRI. SRI values are bucketed by values corresponding to (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on. (b) RDI versus number of declines. Increasing trend indicates higher the RDI, higher is the number of declines.

6.9.1.1 Mean Editor Assignment Time (MEAT)

For each editor we obtain the time span (in days) between any two consecutive assignments and calculate the average time span between the two assignments. Formally, we define for editor i, $MEAT_i$ as

$$MEAT_i = \frac{1}{n-1} \sum(\delta_{j+1} - \delta_j)$$

where n is the total number of assignments to the editor i and δ_j is the date of the j^{th} assignment. In figure 6.14(a) we bin the editors based on the $MEAT$ and calculate the median average citation of the papers assigned to the editors in each bin. We observe

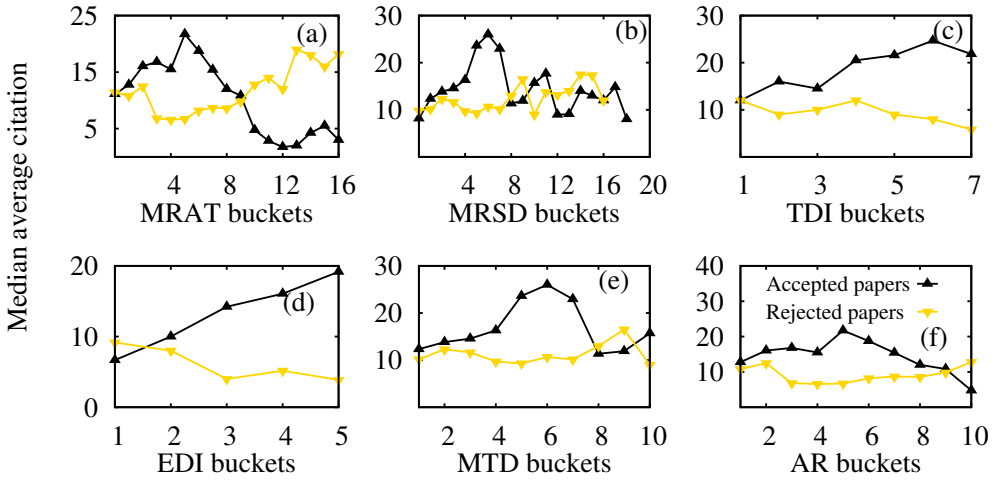


Figure 6.16: (a) Median Average citation (MAC) versus $MRAT$. $MRAT$ values are bucketed into 20 buckets of equal size with range(1,498.8),(b) MAC versus $MRSD$ (c) MAC versus TDI, (d) MAC versus EDI, (e) MAC versus MTD and (f) MAC versus AR. For both (c),(d) and (e), the x-axis values are bucketed by values corresponding to ($\geq$ 0 and $<$ 0.1), ($\geq$ 0.1 and $<$ 0.2) and so on. For (b) and (f) values (x-axis) are divided into 10 buckets of equal size.

that for accepted papers very low or very high $MEAT$ values lead to lower citations. An exact opposite behavior is observed for rejected papers. This indicates that editors who are assigned time and again (low $MEAT$) or rarely (high $MEAT$) often fail to judge the quality of the papers assigned to them.

6.9.1.2 Self Review Index (SRI)

Self Review Index (SRI) measures the fraction of papers for which the editor assigned herself as the reviewer. Formally, for an editor i, we define SRI_i as

$$SRI_i = \frac{\varrho_i}{\rho_i}$$

where ρ_i is the number of papers i was assigned as editor while ϱ_i is the number of papers i assigned herself as reviewer. We observe that with increasing values of SRI the median average citation for accepted papers decreases while that for rejected papers increases (refer to figure 6.14(b)).

6.9.1.3 Referee-Author pair Diversity Index (RADI)

We observe that editors in numerous cases assign papers from a certain author to only a certain reviewer. To investigate whether this allows for less impactful research from this author getting accepted, we define a metric which we call **Referee-Author pair Diversity Index (RADI)**. Formally we define for editor i, the $RADI_i$ score as

$$RADI_i = -\sum_{j,k} p_{j,k} \log p_{j,k}$$

where $p_{j,k}$ denotes the proportion of times a paper from author k was assigned to reviewer j by the editor i. In figure 6.14(c) we bin the editors based on the $RADI$ and calculate the median average citation of the papers assigned to the editors in each bin. We observe that more the diversity score higher is the citation of the accepted papers and correspondingly lower is the citation of the rejected papers.

6.9.1.4 Referee Diversity Index (RDI)

As a following step, we check whether an editor always chooses from a fixed set of reviewers or a diverse set of reviewers while making a paper assignment and, more importantly, does this influence the performance of the editor in terms of the impact of the reviewed paper. We define for each editor(i) a metric called **Referee Diversity Index (**RDI_i**)** as -

$$RDI_i = -\sum_{j} p_j \log p_j$$

where p_j denotes the proportion of times reviewer j was assigned a paper by editor i. More diverse the set of reviewers higher is the score. In figure 6.14(b) we bin the editors based on the RDI and calculate the median average citation of the papers assigned to the editors in each bin. We observe that more the diversity score, higher is the citation of the accepted papers and correspondingly lower is the citation of the rejected papers.

A summary statistic of all the above factors that may be used to identify anomalous editors are noted in Table 6.4.

The dataset allows us to find out the cases when the reviewer declined to review a paper on being assigned by an editor. We observe that editors with high RDI are also declined more often. In figure 6.15(b) we plot RDI value and the number of declines for each editor. An increasing trend indicates that more diversely the editor tries to select reviewers more she gets declined by the reviewers. This in many cases may force the editor to be less proactive and always select from a specific set of 'reliable' referees.

6.9.2 Reviewer

In this section, we investigate anomalous behavior of the referees. Recall that we define the behavior of a reviewer to be anomalous if the papers accepted by her are low cited or the papers rejected by her are highly cited. As in case of the editors, here also we investigate different factors that could be indicative of such anomalous behavior.

6.9.2.1 Mean Reviewer Assignment Time (MRAT)

This is essentially same as MEAT. For a reviewer i, we define $MRAT_i$ as

$$MRAT_i = \frac{1}{n-1} \sum (\delta_{j+1} - \delta_j)$$

where n is the total number of assignments of reviewer i and δ_j is the date of the j^{th} assignment. In figure 6.16(a) we plot $MRAT$ (binned) and median average citation of the papers reviewed for each reviewer. We observe that papers reviewed by reviewers with low $MRAT$ (high frequency of assignment) tend to be cited less and increases as $MRAT$ increases. This is followed by again a steep decrease in citation. This indicates that the reviewers assigned very frequently are often less reliable while those assigned only occasionally are also not likely to correctly judge the quality of the paper.

6.9.2.2 Mean Report Sending Delay (MRSD)

We argue that the time taken by a reviewer to send back the review report could be an indicator of his performance. If a reviewer on average sends back the review very quickly it is highly likely that the review was done in a hurry. Similarly, if the report was sent after being reminded by the editor numerous times, it is also highly likely the review report could be anomalous. For a reviewer we calculate the time delay between the date of her assignment and the date she sent back the report for each of her assignments. To measure $MRSD$ we calculate the mean value of all the delays. Note that we do not consider the assignments which the reviewer declined. Formally, for a reviewer i, we define $MRSD_i$ as

$$MRSD_i = \frac{1}{n}\sum(\delta_i - \Delta_i)$$

where n is the total number of assignments, Δ_i is the date of assignment and δ_i is the date when the report was received by the editor. On plotting against median average citation we observe a similar trend as was observed in case of $MRAT$ (refer to figure 6.16(b)). Papers reviewed by reviewers with low $MRSD$ value are often less cited, indicating that reviewers sending back their report very quickly often do it in a hurry and fail to correctly judge the quality of the paper while those taking very long to send report are prone to failure as well.

6.9.2.3 Topic Diversity Index (TDI)

JHEP associates with each submission a set of keywords which roughly indicates the domain of the work. We use these associated keywords as a proxy for topic. For each reviewer, we segregate all the keywords of the papers reviewed by her which we call the keyword corpus for the reviewer. Formally for a reviewer i, we define TDI_i as

$$TDI_i = -\sum_j p_j \log p_j$$

where p_j is the proportion of keyword j in the keyword corpus for reviewer i. We segregate the reviewers based on the diversity score and calculate the median average citation of the

papers reviewed by them. We observe that the median average citation for reviewers with low TDI are low mainly because the number of papers reviewed by them are also less. The value increases with increasing TDI (refer to figure 6.16(c)). The reviewers with low TDI are often the ones who have reviewed a very small number of papers while the reviewers with high TDI are mostly assigned papers by a large number of editors.

6.9.2.4 Editor Diversity Index (EDI)

Reviewers could be selected for review by a large set of editors or could only be selected by a single or a small set of editors. We check whether a reviewer selected by many editors is more reliable compared to one who is selected by a single or a very small set of editors. To this aim we assign each reviewer a score called Editor Diversity Index, EDI_i which is defined as

$$EDI_i = -\sum_j p_j \log p_j$$

where p_j represents the proportion of times reviewer i was assigned by editor j. We segregate the reviewers based on EDI and calculate the median average citation of the papers reviewed by them. We observe that as EDI increases median average citation also increases (refer to figure 6.16(d)) indicating that reviewers assigned by multiple editors are often more reliable.

6.9.2.5 Mean Time to Decline (MTD)

We further investigated the cases where the reviewer declined the assignment. In specific, we calculated the time delay (in days) between the date she was assigned and the date she conveyed her decision of declining to review. For each reviewer we define **Mean Time to Delay**, MTD_i as

$$MTD_i = \tfrac{1}{d}\sum_j (\mu_j - \Delta_j)$$

Table 6.4: Features used for detecting anomalies.

	Factor	Mean	Median	Max	Min	St. Dev
Editor	$MEAT$	35.06	29.1	108.25	3.28	23.19
	RDI	6.57	6.79	8.85	0.0	1.44
	$RADI$	8.86	9.21	11.94	0.0	1.87
	SRI	0.28	0.25	0.85	0.0	0.19
Reviewer	$MRAT$	363.3	193.7	5389	26.9	508.9
	$MRSD$	19.28	17.50	122	16.5	11.45
	TDI	4.07	3.96	8.10	1.0	1.44
	EDI	1.12	0.91	4.58	0.0	1.19
	AR	0.65	0.71	1.0	0.0	0.2
	MTD	3.86	3.12	69.0	1.0	4.96
	DFI	0.19	0.12	1.0	0.0	0.30

where d is the number of assignments that reviewer i declined and μ_j and Δ_j are respectively the date of assignments and date of reply for paper j by reviewer i. We segregate the reviewers based on their MTD values and calculate the median average citation. We observe that the reviewers who delay often in reporting their decision to the editor of being unable to review usually tend to fail in judging a paper quality when they do review (refer to figure 6.16(f)).

6.9.2.6 Acceptance Ratio (AR)

Acceptance Ratio (AR) of a reviewer is defined as the proportion of papers accepted by the reviewer. For a reviewer i, AR_i is formally defined as

$$AR_i = \frac{a_i}{a_i + r_i}$$

where a_i and r_i respectively denote the number of papers accepted and rejected by reviewer i. We observe that reviewers with high AR often accept less impactful papers while reviewers with very low AR often fail to identify quality research (refer to figure 6.16(e)). Note that the reviewers are segregated based on their respective AR values while the

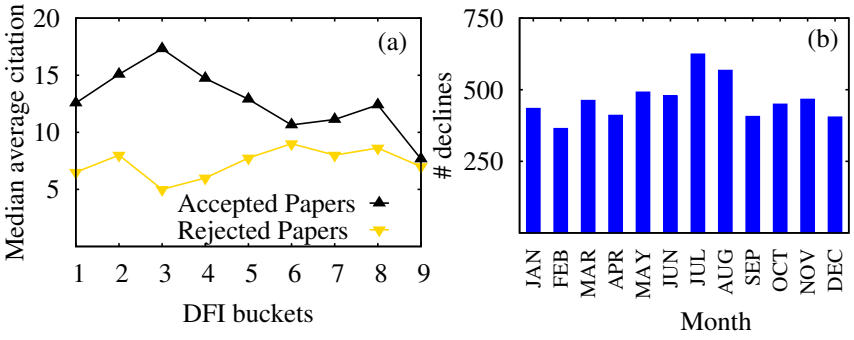


Figure 6.17: (a) Median Average citation versus DFI. DFI values are bucketed by values corresponding to (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on. (b) Number of declines versus the month of the year.

median average citation is calculated. They are segregated into bins based on the AR values where typically the bins are (≥ 0 and < 0.1), (≥ 0.1 and < 0.2) and so on.

6.9.2.7 Decline Fraction Index (DFI)

Decline Fraction Index (DFI) for a reviewer is the fraction of times she declined to review. For a reviewer i, we define DFI_i as

$$DFI_i = \frac{\vartheta_i}{\theta_i}$$

where θ_i is the total number of assignments while ϑ_i is the number of times i declined to review. In figure 6.17(a) we plot median average citation versus DFI. We observe that for accepted papers the citation is higher for lower DFI values and it drops as DFI increases indicating that reviewers declining too frequently often fail to judge the quality of the paper assigned to them.

A summary statistics of all the above factors that may be used to identify anomalous referees are noted in Table 6.4.

We further looked into the data and made some interesting observations which are summarized below -

(i) A bulk of the instances where a reviewer declined to review occurred in the month of

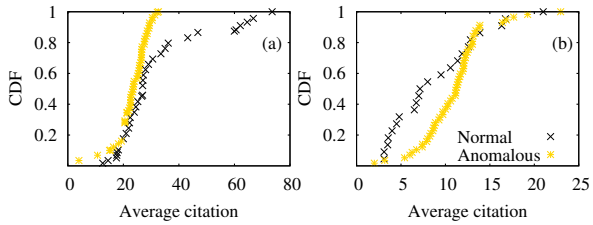


Figure 6.18: Cumulative distribution function of the average citations for the two sets of editors (anomalous and normal).

July and August. This is represented in figure 6.17(b). This probably relates to the vacation time in the Europe and the US. (ii) Of the 4035 reviewers 756 of the reviewers have not been assigned a paper for the last two years. On further investigation we observed that among these there are 505 such reviewers who in their immediate last review assignment agreed to review but did not send back the report.

6.10 Identifying anomalous Editors and Reviewers

In the previous sections we discussed how different factors indicate anomalous behavior of referees and editors. In this section, we check whether we can use them to automatically differentiate between normal and anomalous editors and referees. We propose separate unsupervised models for editors and reviewers.

6.10.1 Editors

For each editor i, we measure $MEAT_i$, RDI_i, $RADI_i$ and SRI_i which form a feature vector. We also consider the editors who were assigned at least 5 papers and accepted at least 1 paper before 2013. To detect anomalies we use the $k - means$ clustering setting with $k = 2$. The two clusters are of sizes 25 and 68 respectively. Clearly this set of 25 editors are the anomalous ones. In figure 6.18 we plot the cumulative distribution of average citation of accepted (figure 6.18(a)) and rejected (figure 6.18(b)) papers. We observe that citation of accepted papers assigned to anomalous editors are significantly lower while citation of rejected papers are significantly higher compared to those assigned to normal editors.

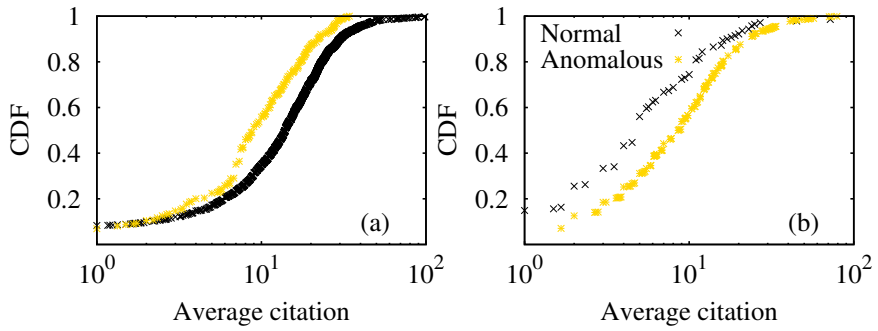


Figure 6.19: Cumulative distribution function of the average citations for the two sets of reviewers (anomalous and normal).

6.10.2 Reviewers

Similarly for each reviewer i we associate a feature vector of size seven consisting of $MRAT_i$, $MRSD_i$, TDI_i, EDI_i, AR_i, MTD_i and DFI_i. We filter out reviewers who have reviewed at least 5 papers and accepted at least 1 before 2013. This reduces our set of reviewers to 2328. By using $k - means$ clustering ($k = 2$), we obtain two clusters of size 339 and 1999. On plotting cumulative distribution of the average citation for accepted (refer to figure 6.19(a)) and rejected papers (refer to figure 6.19(b)), we observe that the papers accepted by anomalous reviewers are cited significantly lesser while those rejected by them are cited significantly higher compared to the normal referees.

6.11 Profiling Anomalous Reviewers

In this section we analyze in more details the performance of the anomalous reviewers. To this aim, we consider for each reviewer, the sequence of papers accepted by her and the citation accrued (within the first three years from publication) by each of these papers. A decreasing trend would suggest decline in performance of the reviewer. Depending on the trend we observe three broad categories within the set of anomalous reviewers

(i) performance deteriorates constantly over time (proportion = 42.5%, figure 6.20(a)).

(ii) performance is good for initial few papers but deteriorates in the long run (proportion = 22.6%, figure 6.20(b)).

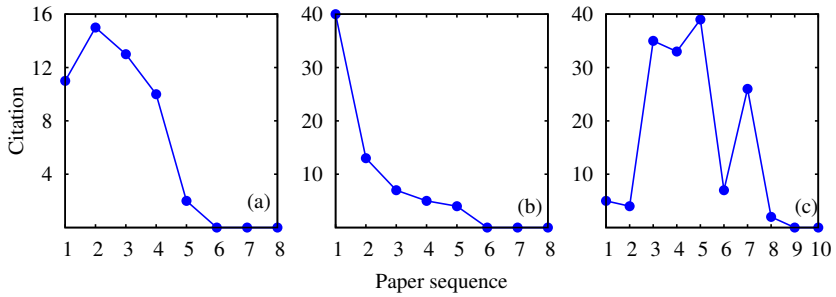


Figure 6.20: Mean citation profile of the reviewers in the three categories.

(iii) performance fluctuates but has a deteriorating trend in the long run (proportion = 34.9%, figure 6.20(c)).

6.12 Conclusion

In this paper we provided a framework for identifying anomalous reviewers and editors based on their review history considering Journal of High Energy Physics as a case study. We identified several factors that are indicative of anomalous behavior of the editors as well as reviewers. In specific for editors we observed that - (i) high frequency of assignment, (ii) selecting from a very small set of referees for reviewing, (iii) assigning same reviewer to papers of same author and (iv) assigning herself as reviewer instead of assigning someone else could be indicative of anomalous behavior of the editor.

Similarly for reviewers we observe that - (i) high frequency of assignment, (ii) delay in sending report, (iii) assignment from only a single editor or a very small set of editors (iv) very high or very low acceptance ratio and (vi) delay in notifying the editor about her decision to decline are also indicative of anomalous behavior and often leads to under-performance. Based on these factors we were able to identify anomalous referees and editors using an unsupervised clustering approach.

Future directions: We believe our findings could be useful in better assignment of editors and reviewers and thereby improve the performance of the peer-review system. Assigning good reviewers is an important part of the peer-review process and our findings allow for

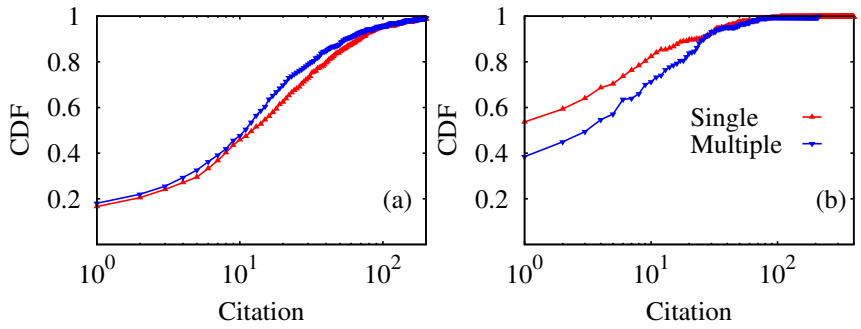


Figure 6.21: Citation distribution of the multi-refereed and single-refereed papers for (Left) accepted and (Right) rejected papers for JHEP dataset.

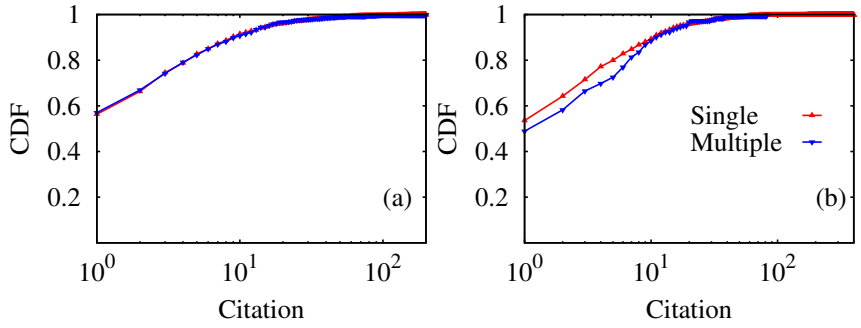


Figure 6.22: Citation distribution of the multi-refereed and single-refereed papers for (Left) accepted and (Right) rejected papers for JSTAT dataset.

identifying under-performing referees. This could be a first step towards developing a reviewer-recommendation system whereby the editors are recommended a set of reviewers based on their performance. We plan to come up with such a system in subsequent works.

6.13 Single vs multiple referee system

In this section we systematically compare the papers which are reviewed by multiple peers (which we call multi-refereed) with those which are reviewed by a single peer (which we call single-refereed). To this aim we first compare the long term citations of both of these classes of papers. We further analyze the review reports received from the peers and check whether the reviewers disagreed in their opinion about the paper.

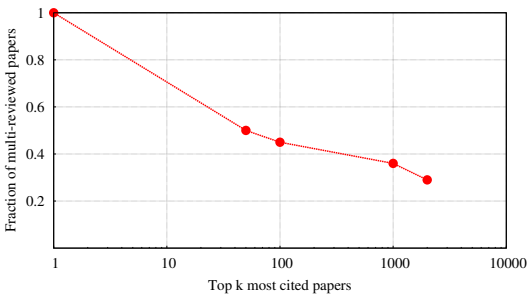


Figure 6.23: Fraction of multi-refereed papers in the top k most cited papers where $k = 1, 50, 100, 500, 1000, 2000$.

6.13.1 Citation

We first look into the long term citations of the papers which are reviewed by multiple peers as well as those reviewed by a single peer. We consider the accepted papers and the rejected papers separately. In figure 6.21(a) we plot the cumulative distribution of the citations received by the accepted papers belonging to the single and the multi-refereed classes for JHEP. We observe that the accepted papers which are multi-refereed tend to garner less citations (mean 30.62) in the long run as compared to the single-refereed (mean 36.48) ones $(p < 0.01)$. An exact opposite trend is observed in case of the rejected papers (mean = 12.8 (multi), 8.42 (single), $p < 0.02$) (refer to figure 6.21(b)).

We repeat the same experiment for the JSTAT papers as well. The corresponding plots for the accepted and the rejected papers are shown in figures 6.22(a) and 6.22(b) respectively. Although for accepted papers the mean citation (26.82) for the single referee case is marginally larger than the multi-referee case (mean citation $= 24.26$), for rejected papers, mean citations for multi-refereed (10.92) and single-refereed (6.86) papers are significantly $(p < 0.02)$ different. These results together indicate that when working in groups the reviewers often fail (on average) to correctly judge the quality of the paper.

6.13.2 Top cited papers

The above results would indicate that it is always better to assign a single reviewer instead of multiple reviewers. However, a deeper and a more microscopic analysis portrays a very different picture. On computing the proportion of multi-refereed papers in the top k most

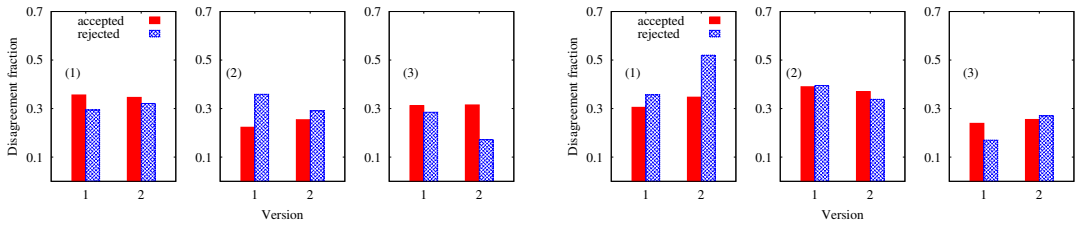


Figure 6.24: Fraction of cases where the reviewers disagreed with respect to (1) length (2) sentiment and (3) content for (a) JHEP and (b) JSTAT datasets.

cited papers (where $k = 1, 50, 100, 500, 1000, 2000$) we observe that the fraction is very high for really top cited papers and then it decreases (refer to figure 6.23). In fact, the most cited paper in our dataset is also multi-refereed. This indicates that sincere, knowledgeable multi-review indeed helps in identifying impactful papers. We also observe that for 100 least cited papers only 26 of them are multi-reviewed which again indicates single-review may lead to several unappealing entry.

6.13.3 Conflict in review reports

There may be various reasons behind the average under-performance of multiple reviewer system, one of which could be the difference in opinions among the reviewers whereby the paper may have got accepted despite a subset of reviewers not recommending the paper. Understanding this aspect from the dataset is difficult as although the review reports from the respective peers are accessible, their final decisions (accept/reject/major-review) are not explicitly available making it difficult to understand the differences in opinions. Therefore we leverage on traditional natural language processing tools to segregate papers according to reviewers' agreement/disagreement and then check their impact. Accepted and rejected papers are considered separately. We specifically look into three factors - (i) length (ii) sentiment and (iii) content of the review report of all the involved referees. Note that in this work we do not aim to identify the most distinguishing features for identifying or quantifying differences in opinion from the review reports. The proposed features albeit crude, are sufficiently adequate to unfurl the dissonance among the referees which is our prime intention.

6.13.3.1 Length

We start by comparing the length of the review reports sent by each peer assigned for a given paper. Our hypothesis is that if the peers have a similar opinion on the assigned paper, the length of their reports would be typically close. To this aim, we calculate the standard deviation of the lengths (i.e., number of words) of the reviewer reports for every round of review for each paper and use it as a metric to judge whether a particular round is discordant (i.e., the reviewers disagreed) or concordant (i.e., the reviewers agreed). If the standard deviation is greater than the length of the smallest-length review, we classify the review round as discordant. Otherwise, we classify the review round as concordant. In figure 6.24(1) we plot the fraction of multi-refereed papers where the peers disagreed in case of JHEP. The corresponding result for JSTAT is shown in figure **??**(1). We observe that for JHEP the peers disagree in almost 35% of the cases in the first round of the review for the papers which are finally accepted and close to 30% for the papers which are rejected. The corresponding numbers for JSTAT are 30.5% and 35.7% respectively. In fact, even in the second round of reviews the peers tend to disagree as well (refer to figure 6.24(1) and **??**(1)).

6.13.3.2 Sentiment

We next analyze the sentiments latent in the review reports. If for a document all the reviewers are of positive opinion or of negative opinion we consider it as a concordant case. If the opinion of at least one of the reviewers differ from the rest we consider it as discordant. To calculate the sentiment score of the reports we use nltk-sentiwordnet toolkit[4]. In figures 6.24(2) and **??**(2) we plot the fraction of discordant cases for both the accepted and rejected papers for JHEP and JSTAT respectively. We observe that for JHEP, across the two versions on average the referees disagree in around 20% of the cases for the accepted papers and in around 30% of the cases for the rejected papers. In case of JSTAT the disagreement is even higher with the corresponding numbers being around 40% and 35% respectively.

[4]http://www.nltk.org/

Table 6.5: Jaccard similarity between the cases identified as discordant by the different metrics.

	Length	Sentiment	Content
Length	1.0	0.63	0.37
Sentiment	0.63	1.0	0.29
Content	0.37	0.29	1.0

6.13.3.3 Content

To further analyze if the contents in the review reports from multiple referees are conflicting, we perform a term frequency-inverse document frequency (TF-IDF) based analysis to estimate the extent of difference in the referee opinions. For this purpose, we first need to construct a vocabulary representing the typical review report for an accepted (rejected) paper. Toward this objective, we collate the final round report of all the single-refereed accepted (rejected) papers. These are the texts where we are sure whether the review led to an acceptance (a rejection) of the paper. Next we construct the respective TF-IDF vectors corresponding to the accepted (rejected) papers from the collated text. Let us call these *acceptance* (*rejection*) vectors. Finally, we compare the TF-IDF vector corresponding to each report of the multi-refereed papers to check if it aligns more with the acceptance vector or the rejection vector. If the reviews from all the referees align to the same vector we consider it as a concordant case. Otherwise we consider it as a discordant case. In figures 6.24(3) and **??**(3) we plot the fraction of discordant cases for JHEP and JSTAT respectively. We observe that for JHEP, across the two versions/rounds the reviewers disagree on average in almost 30% of the cases for accepted papers and close to 25% on average for rejected papers. For JSTAT the reviewers disagree on 24% cases in version 1 and 26% cases in version 2 for accepted papers. The corresponding numbers for the rejected papers are 17% and 27% respectively.

In table 6.5 we present the Jaccard similarity between the discordant cases as detected by the above three metrics (length, sentiment and content) to check whether the same cases are identified by all the three metrics. We observe high Jaccard similarity between cases identified as discordant by length and sentiment. This similarity is a bit lower with the cases identified as discordant by content analysis.

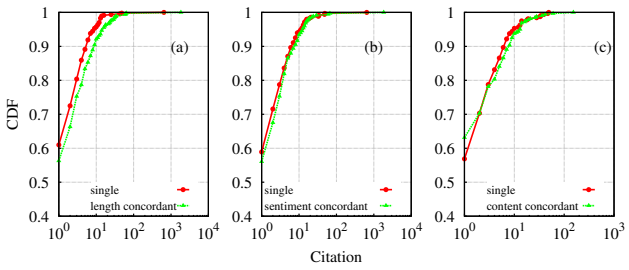


Figure 6.25: Cumulative distribution function of citations for single refereed papers and concordant multi-refereed papers in terms of (a) length, (b) sentiment and (c) content for JHEP.

6.13.4 Impact of discordance

If we consider the concordant (where the reviewers agreed) and discordant (where the reviewers disagreed) papers separately, we observe that the concordant papers are often highly cited in the long run. In figure 6.25 we plot the cumulative distribution function of citations received by the concordant papers and that received by the single-refereed ones for the JHEP dataset. In all the three cases (length, sentiment and content), average citation (32.65, 35.45, 31.67 respectively) of concordant multi-refereed accepted papers tend to be more compared to that of the single-refereed ones (30.39). We observe an exact opposite behavior in case of the discordant papers (refer to figure 6.26). More specifically average citation of length discordant papers (22.33) differ significantly ($p < 0.03$) from average citation of single refereed ones. Similar trend is observed for sentiment discordant (mean $= 19.23$, $p < 0.02$) and content discordant (mean $= 21.36$, $p < 0.03$) cases. This goes to show that if the reviewers are chosen correctly, the multi-referee setup can perform better than a single-referee setup.

As we shall see in the next section, the impact of the papers can be understood in further details if we investigate the performance of the reviewers assigned to review the papers.

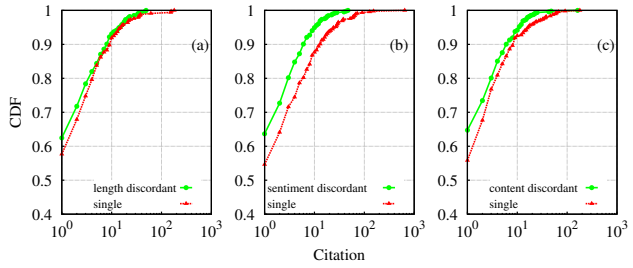


Figure 6.26: Cumulative distribution function of citations for single refereed papers and discordant multi-refereed papers in terms of (a) length, (b) sentiment and (c) content for JSTAT.

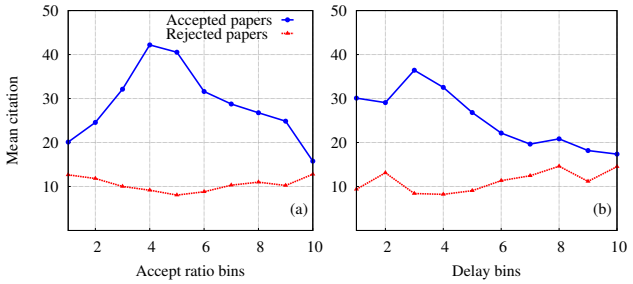


Figure 6.27: Mean citation versus (a) accept ratio (b) assignment delay buckets for the JHEP dataset. Note that the papers are segregated into accept ratio/delay bins and the mean citation is calculated for each bin. Typical bin sizes for accept ratio are < 0.1,$(\geq 0.1$ and $< 0.2)$ and so on while for delay the sizes are < 100, $(\geq 100$ and $< 200)$ and so on.

6.14 Analyzing reviewer tendencies

Reviewers are assigned with the responsibility of judging the quality of a submitted paper and hence their knowledge and training is highly critical. In fact, the decision of acceptance or rejection of a paper depends on the reviewer's perception of the paper.

In [203] authors define a referee/editor to be anomalous (under-performing) if -
(i) papers accepted by him/her have low citation (research wrongly judged to be impactful)
(ii) papers rejected by him/her have high citation (quality research wrongly judged as flawed)
In fact, the authors propose a method for identifying under-performing (anomalous) referees/editors. Leveraging on the same method we observe that the proportion of referees classified as under-performing are approximately 26% and 21% respectively for JHEP and

JSTAT datasets. We make the following general observations -
(i) If we consider all the cases where an under-performing editor assigned multiple reviewers for a submission, in 69.8% cases at least one of them was under-performing. This indicates that unless the reviewers are selected carefully, chances are it might lead to wrong judgment.
(ii) Under-performing reviewers when part of a multi-referee system tend to do a better judgment as compared to cases when they serve as single reviewers. This is illustrated by the fact that average citation of the accepted papers reviewed by multiple reviewers with at least one under-performing reviewer (32.4) is more compared to that of the papers reviewed by a single anomalous reviewer (18.2) across the two datasets.

6.14.1 Factors determining performance of the referees

We next look into factors that could be used to quantify the performance of the reviewers. The quantification can be used as a fitness value for assignment of a new submission (we use these to calculate these quantities in a later section to develop a scheme for automatic referee group selection). Given a submission, we identify two factors that are indicative of reviewer fitness (i) accept ratio and (ii) time since the last assignment.
(i) *Accept ratio*: Given a submission, we consider for each reviewer (i) the fraction of papers (s)he has accepted at the time of submission. We denote this by a_i. To show that it is indeed an indicator we consider all the accepted and the rejected papers as well as their assigned reviewers. We then calculate the accept ratio of the assigned reviewers and compare them against the citations received by each of them. In figures 6.27(a) (JHEP) and 6.28(a) (JSTAT) we plot accept ratio against the citations received by the accepted as well as the rejected papers. Note that we bin the papers based on the accept ratio and calculate the average citation in each case. The typical bin sizes are ≤ 0.1, $(> 0.1$ and $\leq 0.2)$ and so on. We hence obtain 10 such bins numbered 1-10. In case of multi-referee papers the average accept ratio of the reviewers is considered. We observe that for both the datasets very high accept ratio or a very low accept ratio might lead to wrong judgment.
(ii) *Time since last assignment*: Given a submission and its corresponding submission date we calculate for each reviewer (i) the time (in days) between the last review assignment date and the submission date (denoted by d_i). To illustrate the rationale, we again consider

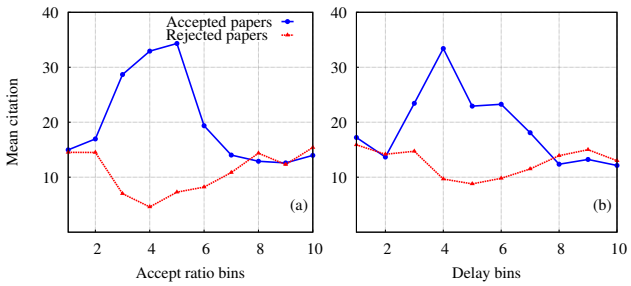


Figure 6.28: Mean citation versus (a) accept ratio (b) assignment delay buckets for JSTAT dataset. Note that the papers are segregated into accept ratio/delay bins and the mean citation is calculated for each bin. Bin sizes are same as figure 6.27.

all the accepted and the rejected papers and calculate the delay for the assigned reviewers. We again bin the delay values and calculate the average citation (refer to figures 6.27(b) and 6.28(b)). Typical bin sizes are ≤ 100, $(> 100$ and $\leq 200)$ and so on (10 such bins are obtained numbered 1-10). We observe that the reviewers who are assigned very close to their last assignment or those who have not been assigned for a long time often fail to correctly judge the quality of the paper correctly as the papers accepted by them are cited less on average while those rejected are cited more.

6.14.2 Action with under-performing reviewers

A naive solution could be to not assign the under-performing reviewers and only assign the best performing ones, but is not feasible since the number of referees is limited and they often decline assignments. Hence a better solution would be to group them such that the overall performance improves. To this aim, we first divide the reviewers into 3 classes separately for accept ratio and time since last assignment. A reviewer i with accept ratio $a_i < 0.3$ is assigned "Low" (L), with $0.3 \leq a_i < 0.6$ is assigned "Medium" (M) and with $a_i \geq 0.6$ is assigned "High" (H). Note that reviewers in M were the best performing referees (refer to figures 6.27(a) and 6.28(a)). Each multi-refereed paper (by exactly 2 referees) is classified into one of the six classes (LL, MM, HH, LH, MH, LH) based on the class of each referee and the average citation of the papers in each class is noted (figure 6.29(a)). We observe that when both the referees belong to M class (MM) the performance is naturally well. More importantly, reviewers in L and H class perform better when paired with a referee from M class (even better than MM). On repeating the same experiment with

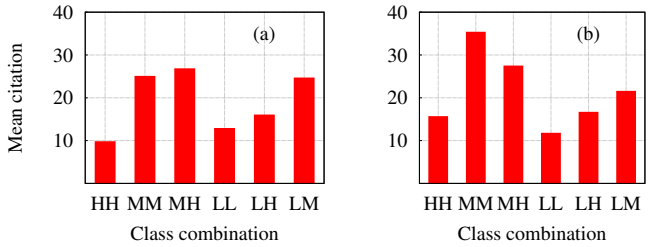


Figure 6.29: Mean citation for papers belonging particular class combination with respect to (a) accept ratio (b) time since last assignment. For example LL would represent a paper reviewed by referees both belonging to class L.

Table 6.6: Proportion of discordant cases (length, sentiment and content) in each reviewer class combination with respect to (accept ratio, time since last assignment).

	Length	Sentiment	Content
MM	0.176,0.159	0.194, 0.143	0.164, 0.152
LM, HM	0.173, 0.162	0.243, 0.192	0.237, 0.216
LL, HH, LH	0.312, 0.254	0.371, 0.286	0.293, 0.311

time since last assignment, we observe a similar trend (figure 6.29(b)) with MM class performing the best followed by MH and LM. Note that in this case reviewers with $d_i < 100$ are assigned class L, with $100 \leq d_i < 300$ are assigned M and rest are assigned H. More importantly the discordant cases mostly occur for class combinations LL, HH and LH (refer to table 6.6). In fact, MM has the least proportion of discordant cases. This further indicates the correct reviewer grouping is critical in curtailing the discordant cases which is one of the prime reasons behind the multi-reviewer system failing. Note that the above results are obtained for JHEP dataset and a similar pattern is observed for JSTAT as well.

6.15 Forming reviewer groups

The results in the previous section together lead us to believe that the editors in numerous cases fail to assign reviewers which may further lead to flawed papers being accepted into the literature while at the same time quality research being overlooked. We hence proceed to propose a framework that could recommend compatible referee groups which could then be assigned to a submitted paper. Note that such a system would be required to -

(i) recommend multiple items (referees) to a single user (submission).
(ii) recommend only homogeneous (compatible) items together while making multiple recommendations.
(iii) provide a scheme to rank the groups since there could be numerous compatible groups and recommending all of them would be infeasible.
Genetic algorithm has already been found to be very effective in obtaining desired groups in collaborative learning setting [10, 153] and we argue that assigning multiple referees to a submission is similar to forming compatible referee groups. More importantly, for such a framework a scheme for ranking the groups is inherently present (we provide a detailed description later in this section). We hence leverage the idea of GA based team formation framework to obtain the best possible combination of referees which might lead to better judgment of the quality of the work.

6.15.1 Problem definition

Given a submission S, a set of reviewers $R = \{R_1, R_2, \ldots, R_n\}$ and the number of reviewers to be assigned for a paper k, the goal is to obtain a set of potential reviewer groups $G = \{G_1, G_2, \ldots, G_u\}$ (each of size k) who could be assigned to referee a submission.

6.15.2 Methodology

We now proceed to develop a solution to the above problem based on a genetic algorithm framework which is inspired by the method proposed in [10]. Typically, chromosomes are represented as simple strings of data and instruction. In our case chromosomes represent solutions while reviewers are chosen as genes. The details of our algorithm is as follows –

6.15.2.1 Selecting a population

Genetic algorithm starts with an initial population represented by the chromosomes. Ideally the assigned reviewers should be knowledgeable and experienced in the topics

related to the submitted paper. Although information about the associated topic of the papers are not present, each paper in the dataset is associated with a set of keywords (at least 1 and at most 4, assigned by the publisher) which we use as a proxy for the topic. While the JHEP dataset consists of 201 unique keywords, JSTAT dataset has 562. This again indicates that the papers in JSTAT are more diverse. Hence for a given submission we first extract the keywords. Given the keywords, we extract from the pool of referees only those peers who have reviewed, till the date of the submission in question, at least one paper that has one (or more) keywords in common with that of the submission. This set ($\bar{R}$) represents the potential population of reviewers for the query paper.

6.15.2.2 Initialization

Once the initial population $\bar{R}$ of reviewers is obtained, we classify them based on the accept ratio and time from the last assignment. For each reviewer i we obtain the accept ratio a_i and time from the last assignment d_i (scaled between 0 and 1). Finally, we calculate the fitness score for the reviewer i as -

$$f_i = \alpha * a_i + (1 - \alpha) * d_i \tag{6.1}$$

where α is a tuning parameter. Based on the fitness score we classify each reviewer into one of the three classes. Reviewers with fitness score less than 0.33 are classified as class 1 (RC_1), between 0.33 and 0.66 are classified as class 2 (RC_2) and the rest as class 3 (RC_3). We then club the reviewers randomly into non-overlapping groups of size k. This forms the initial generation with each group represented as a chromosome. Note that a generation forms a solution set for the above problem.

6.15.2.3 Fitness evaluation

Once a generation is obtained, we proceed to calculate the fitness of each chromosome (group) and, thereby, calculate the fitness of the whole generation. The generation level fitness (FG_{gen}) is calculated according to function 1 for a generation gen. For each class RC_i we initially calculate the number of reviewers in RC_i per group (represented by

RPC_i). Now for every group g_j we find the number of reviewers of each class RC_i which we represent by $RC_i^{g_j}$. We penalize the chromosome if the constraint (refer to line 10 of function 1) is violated by not adding anything to the final score; otherwise, a value of 1000 is added to the fitness score. Note that 1000 is a representative value and for practical purposes we can use any positive value (C). More importantly this scoring scheme further allows for ranking the groups.

Function 1: generation_level_fitness()

```
1  G, RC_i
2  FG_gen ← 0
3  |G| ← Number of groups
4  |RC_i| ← Number of peers in RC_i
5  |RPC_i| ← Number of peers of class RC_i per group (i.e., |RPC_i| = |RC_i|/|G|)
6  |RC_i^{g_j}| ← Number of actual peers of class RC_i in g_j
7  for each group g_j ∈ G do
8      for each RC_i do
9          if |RC_i^{g_j}| ≥ |RPC_i| then
10             FG_gen = FG_gen + 1000
11 return FG_gen
```

To illustrate on how the fitness function is calculated we consider here a toy example. Consider a population of 16 reviewers (with ids 0 - 15), group size of 4 and 3 reviewer classes (1, 2, 3) with number of reviewers in each class being 6, 5 and 5 respectively. Let the reviewers with ids 0 - 5 are assigned to class 1, 6 - 10 to class 2 and rest are assigned to class 3. So, $|RPC_1| = 1.5$, $|RPC_2| = 1.25$ and $|RPC_3| = 1.25$. Each generation consists of 4 groups. Let us consider a group g with reviewers 0, 1, 6 and 12. For this group $|RC_1^g| = 2$, $|RC_2^g| = 1$ and $|RC_3^g| = 1$. Since $|RC_1^g| > RPC_1$, it contributes a score of 1000. Similarly $|RC_2^g|$ and $|RC_3^g|$ does not contribute to any score. Hence the score contributed by this group to the generation is 1000. Sum of this score across all the groups in the generation equals the generation level fitness. Note that our algorithm is inherently fair because the proportion of referees in each class is maintained in the recommended set of referees.

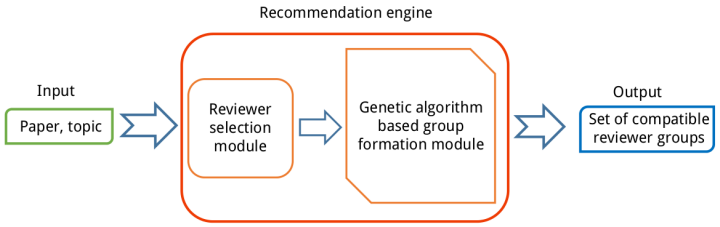


Figure 6.30: Block diagram demonstrating the work flow of our system.

6.15.2.4 Crossover

Crossover is the process in which two chromosomes combine the genetic material to create a new generation such that the new one possesses the genetic material of both. Although several techniques for crossover exist [81], we here consider the one-point crossover technique. One random position in the chromosome is chosen to determine the crossover point. In the offsprings produced, the data till the crossover point are copied verbatim from the parents while the data beyond the crossover point are swapped between the two parents. As a result, these new chromosomes or offsprings share some similar features from the parents chosen. If the crossover is not applied, offsprings are exact copies of the parents. We have set the crossover rate for our experiments to 70%. Note that mutation process can also be used to maintain genetic diversity but we only stick to crossover for our experiments. The process of crossover is repeated several times and the fitness score of the generation is calculated each time. The generation with the best score is confirmed as the best possible reviewer grouping for the submitted paper.

6.15.3 Evaluation

To evaluate the performance of our algorithm, we initially separate out the multi-refereed papers from both the datasets. Based on the long term citations received by each, we classify them as high and low cited. Specifically we rank them based on their citations and top 25 percentile are classified as highly cited ones and bottom 25 percentile are classified as low cited ones. The classification is done separately across both accepted and rejected papers. We hypothesize that -

(i) For accepted papers, highly cited papers represent the cases where the reviewer group was chosen correctly. Moreover, if the reviewer group which was originally assigned the paper is also one of the groups recommended by our algorithm, we consider it as a true positive case.
(ii) Similarly for rejected papers, low cited papers also represent the cases where the reviewer group was chosen correctly and if our recommendation matches we consider it as true positive as well.
(iii) For accepted papers, low cited papers represent the cases where the reviewer group was not chosen correctly. If for such a paper the original reviewer group does not feature in the list of recommended groups, we consider it as true negative.
(iv) Similarly, for rejected papers, high cited papers also represent the cases where the reviewer group was not chosen correctly and our hypothesis follows.
Since our algorithm allows for ranking of the groups based on the fitness score, we recommend top k (how the results vary with k is shown later) percentile groups. A block diagram representing the work flow of our system is presented in figure 6.30.

Sensitivity to the parameters: As mentioned earlier, our algorithm requires two parameters, (i) α and (ii) the crossover rate. In figure 6.31(a) we plot the true positive rate for different values of α. We observe that TP (averaged over accepted and rejected cases) improves with increasing value of α after which it drops for both JHEP and JSTAT datasets. Optimal result is obtained at $\alpha = 0.6$ for JHEP and $\alpha = 0.7$ for JSTAT. We hence set the value of α at 0.6 and 0.7 respectively for JHEP and JSTAT datasets.
We further look into the crossover rate as well. In figure 6.31(b), we plot TP for different crossover rates. TP is observed to be lower for low crossover rate. This is because lower crossover rate allows for very low change of the chromosomes and since we set the number of generations to a constant On the other hand, for very high crossover rate changes in the chromosome occurs at higher rates which increases the chances of obtaining the best solution. Optimal TP is obtained at 70% crossover rate and hence we set the crossover rate at 70% for our experiments. The number of generations we examine is set at 500.
Key results: For both JHEP and JSTAT mostly two reviewers are assigned per submission albeit there are cases where three reviewers have also been assigned. For generating the results, we only consider the cases where the number of assigned reviewers were two. We calculate for each reviewer the accept ratio and time from the last assignment at the point

of submission. We then run our algorithm for each submission falling in highly cited and low cited accepted and rejected papers. Note that we rank the papers based on citations it has accrued and consider the top and bottom 25 percentile to be high and low cited papers respectively. In table 6.7 we present the true positive and true negative values for both JSTAT and JHEP at different values of k (note that we rank the groups based on the fitness score and recommend groups in the top k percentile). With $k = 15$, for JHEP we could recommend the correct group of reviewers in 81% of the cases (represented by the true positive value) for accepted papers and 75% of the cases for rejected papers. Similarly, we could correctly identify 84% of the true negative cases for the accepted papers and 73% for the rejected papers. Corresponding values for JSTAT are 79% (accepted), 75% (rejected) and 81% (accepted), 71% (rejected) respectively. These results indicate that our algorithm is indeed very effective in recommending referee pairs. We further look into the cases where our algorithm failed to obtain the correct grouping. We observe that in most of these cases a new reviewer (without any previous review history) was assigned. Since our algorithm recommends reviewers leveraging the review history, these cases were missed. We further check our results for $k = 5$ and $k = 10$ which are also reported in table 6.7. With $k = 5$ we were able to obtain TP of 0.61 and 0.59 respectively for accepted and rejected papers in case of JHEP. For JSTAT the corresponding values are 0.63 and 0.59. More importantly we could obtain a high TN of 0.91 (accepted) and 0.89 (rejected) for JHEP and 0.92, 0.82 for JSTAT. This is because we are recommending a smaller number of groups which reduces the possibility of recommending a particular undesired/desired group.

Comparison with baseline algorithms: As our algorithm presents computational overhead a natural question to ask is whether it is worth it. We hence design a simple baseline (random) to compete with our algorithm. For a submission, initial population of suitable reviewers is obtained (in the same way as our GA based algorithm) and each reviewer is classified into one of L, M, H categories. A subset of reviewers is then selected randomly in such a way that the proportion of reviewers in each class (in the population) is maintained in the chosen set. We then randomly form groups of types MM, ML and MH since these are the best performing reviewer groups (refer to figure 6.29). To compare, we obtain the same number of groups as obtained from our proposed algorithm. We further consider two other ablation type baselines - (i) GA with only accept ratio (only a_i, fitness score is calculated with only a_i), (ii) GA with only time since last assignment (only d_i, fitness score calculated with only d_i). Note that this can be achieved by setting α to 1

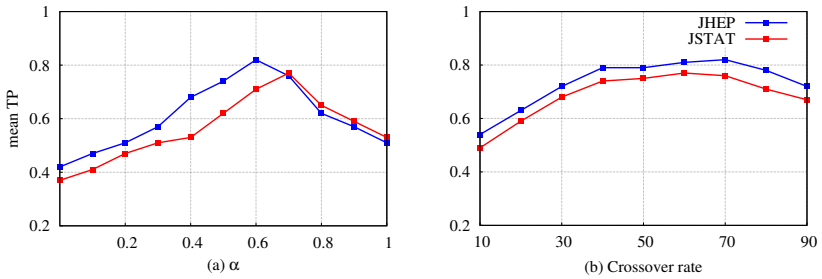


Figure 6.31: Mean true positive (averaged over accepted and rejected papers) value while recommending a set of reviewer groups for papers across different values of (a) α and (b) crossover rate. Experiments repeated for both JHEP and JSTAT datasets. $k = 15$

and 0 respectively. In figure 6.32 we plot the TP and TN for the baseline algorithms and our proposed algorithm (k is set at 15). We consider five random instances and report the mean value for the baseline. We observe TP (averaged over accepted and rejected cases) to be much lower (0.37 (JHEP) and 0.45 (JSTAT) for random, 0.51 and 0.53 for only a_i, 0.42 and 0.37 for only d_i) compared to our algorithm. TN is compared to be higher because of the less possibility of randomly obtaining a particular undesired/desired group. As a second level of evaluation we further consider overlap in the set of groups proposed by our algorithm with those obtained from the baseline algorithm. The purpose of this is to show that the top k groups proposed by our algorithm based on the fitness function cannot be obtained through random allocation. In specific, we measure the Jaccard similarity between the set of groups obtained through baseline and those obtained through our algorithm. Across all the papers we obtain a low similarity value of 0.31 which further corroborates our hypothesis.

6.16 Importance of the editor

Results from the previous section suggests that given the reviewer history and the related topic, our algorithm is able to correctly recommend a set of reviewers. It might therefore be tempting to believe that a peer-review system can be made to function without even the intervention of the editor. However, we would like to point out that this is not the case and our system simply recommends a set of reviewer groups to the editor from which (s)he

Table 6.7: True positive (TP) and true negative (TN) values across accepted and rejected papers measured for different values of k. Results are reported for JHEP and JSTAT datasets.

	JHEP		JSTAT	
K	TP (accept,reject)	TN (accept,reject)	TP (accept,reject)	TN (accept,reject)
5	0.61,0.59	0.91,0.87	0.63,0.59	0.92,0.82
10	0.72,0.66	0.87,0.79	0.73,0.71	0.85,0.76
15	0.81,0.75	0.84,0.73	0.79,0.74	0.81,0.71
Average	0.71,0.66	0.87,0.80	0.72,0.68	0.86,0.76

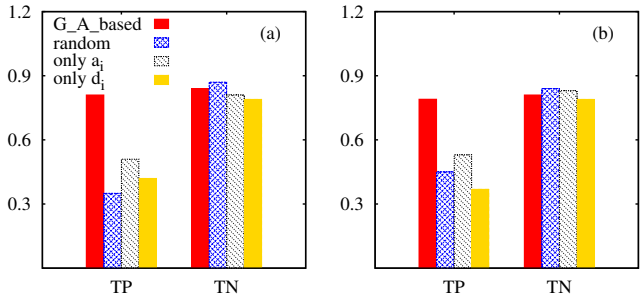


Figure 6.32: Mean true positive (averaged over accepted and rejected papers) value with recommendation by our method (G_A_based) and the baseline. Results are noted for (a) JHEP and (b) JSTAT. $k = 15$.

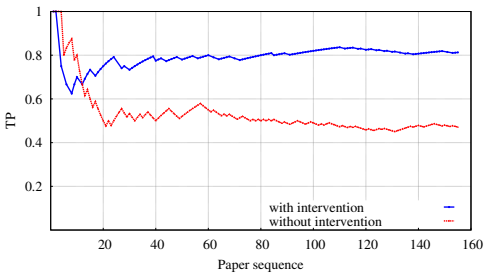


Figure 6.33: True positive value at each point of recommendation of the top 25 percentile (based on citation) accepted papers for JHEP dataset. The papers are sorted by date of submission and x-axis denotes the paper number in the sequence.

has to use his/her expertise and knowledge to choose the most appropriate group. In fact the intervention of the editor to select the reviewer group for each paper is critical to the performance of our scheme in the long run. To illustrate this, we perform the following experiment. The multi-refereed papers are sorted based on their date of submission. The reviewer history for each referee is included till the point of submission of the first multi-refereed paper. For each paper our system recommends a set of reviewer groups; if we consider no editor intervention in our system, the top ranked (according to fitness score) reviewer group is selected from the set of recommended groups and the reviewer history updated accordingly. Note that for our original system, we assumed the editor selected the correct set of reviewers and the reviewer history was updated following the original assignment. Also since, in this simulation settings, we do not know whether the reviewers would have actually accepted or rejected the paper, we flag the paper as accepted with a probability proportional to the current average accept ratio of the assigned referees. The system is allowed to evolve following the above mentioned way.

We now consider the top 25 percentile papers (based on citation) and sort them according to the date of submission. The simulation setup above is then used to recommend a set of reviewer groups. In figure 6.33 we plot the TP value for each paper sorted according to their submission dates. Specifically, we calculate the fraction of correct recommendations at the point of every new submission for both the original system (with editor intervention) and the simulation system (without intervention). We observe that the performance of the system without intervention degrades alarmingly over time compared to the original system. The above results indicate that the expert intervention of the editor in choosing the reviewer groups is extremely important toward proper functioning of the peer-review system.

6.17 Conclusion

We in this paper performed a detailed comparative analysis of cases where a single reviewer was involved in the peer-review process and where multiple reviewers were involved, considering the review history of papers submitted to two leading journals of physics (JHEP and JSTAT). We made an interesting observation that accepted papers which were reviewed by a single referee on average tend to garner a larger number of citations in long term compared to those which were reviewed by multiple referees. An exact opposite behavior is observed in case of rejected papers.

Through a more detailed analysis, we however observed several contradictory trends - although on average single reviewers perform better, however, most impactful papers are largely multi-refereed; the multi-refereed papers generally perform poorly due to discordance between the reviewers. Further we tried to understand the reason behind under-performance and found that frequent assigning of review to a reviewer leads to his performance deterioration. Also those reviewers who have a tendency to be too critical or too liberal fail to identify the real good papers.

A real problem in the process of reviewing is the scarcity of reviewers, hence discarding under-performing reviewers may not be a practical solution. So we checked the performance of these reviewers with high-graded reviewers and find that the quality is much better than their normal average. Hence a technique would be to combine high and low graded reviewers intelligently.

Based on the observations above, we proposed a scheme based on genetic algorithm to recommend reviewer groups to the editors to make reviewer assignments. In fact our system was correctly able to recommend reviewer groups in $\sim 78\%$ cases across the two datasets.

Chapter 7

Conclusion and Future Work

In this chapter we elaborate important contributions from this thesis and finally wrap up this thesis pointing to some future research directions that have been opened from this thesis.

7.1 Summary of Contribution

The demand for data through mobile devices would continue to increase thus leading to an exponential growth in Internet traffic. Merely increasing network capacity by means of infrastructure can not solve Internet traffic issue completely. In addition to it several protocol level modifications are needed. In this regard this thesis contributes in a meaningful way. We discuss contributions of this thesis corresponding to the objectives laid out in the beginning of the thesis.

7.1.1 Efficient Offload using WiFi Network

Cellular networks are becoming heavily congested due to huge data demand. One very promising approach to reduce the load of congested cellular network is to offload its traffic to some other networks like the Wi-Fi network. With availability of cheap storage, Wi-Fi

APs (or base stations) can act as edge server with caching support. Caching of popular files may restrict significant traffic within local network. Applying this notion, this work provides a framework for local area CDN which enables uninterrupted video watching for users with high mobility. Contributions from this work are summarized as follows.

- Use Wi-Fi APs as edge server to bring popular contents closer to users which results in significant reduction in data access delay.

- Develop a methodology for systematic chunk distribution across network based on prior knowledge of human mobility pattern and hence ensure just in time streaming.

- Performance of this system is independent of the city map pattern and hence can be applied to any city with any road map.

- Short association duration with AP for users with high mobility was the main obstacle for smooth streaming; proposed system with caching support enables video streaming for the users with high mobility.

- Simulation results confirm that the proposed system can do a sizeable offload over a wide variation of speed (more than 80% offload in the speed range between 20km/hour - 60km/hour).

- Performance degrades gracefully with increasing traffic, however waiting due to traffic sig-

nal increases performance for high traffic scenario.

- Even with very low density of Wi-Fi AP, system offloads the load significantly (when APs are placed at every 300m, system can offload more than 85%).

7.1.2 Managing Heterogeneous Traffic

Many recent studies pointed out that load distribution across network is not homogeneous which in effect creates issue with overall system performance and user satisfaction. In most of the cases few base stations remain over loaded while others remain under utilized. In recent years Wi-Fi networks are going through an architectural shift. Wi-Fi APs are being connected via high speed wires. This change brings new opportunity to out of band communication among APs. The present work exploits this new architecture to collect a global view of load distribution across network and applies max-flow based strategy for resource allocation and association control. Contributions from this work are summarized as follows.

- Exploiting the architectural shift, proposed system with minimum cost (out of band communication) can create a global load distribution view necessary for optimal association strategy.

- Pressing association control protocol is mapped to the classical max-flow algorithm where APs are considered as source of bandwidth and mobile devices are considered as sink of bandwidth and

maximum flow of bandwidth from source nodes to sink nodes ensures maximum device satisfaction.

- Along with maximum device satisfaction, proposed association control protocol ensures fair chance of association for devices spatially distributed across network.

- With distributed design, proposed association control protocol is highly scalable.

- Proposed association control protocol can admit up to 10% more devices than standard RSSI and LLF based association in the viable load range.

- Simulation results confirm that the proposed system can easily cope up with user mobility.

7.1.3 Restricting Unauthorized Traffic

Among the three objectives of this thesis, this objective is least studied in literature. However, recent studies are pointing out the importance of restricting such traffic both from the point of view of service provider's revenue and unwanted extra traffic in network. In this work we have proposed another dimension of metric to be checked for authentication algorithm namely *shareability* along with existing dimension of security and simplicity to restrict unwanted traffic. An authentication system is proposed based on

user's daily activity to restrict password sharing. This work empirically provides a proof to our hypothesis that *users can remember their own activity mostly while it is very difficult to be guessed by even close friends*. Contributions from this work are summarized as follows.

- This work proposes a new dimension of metric to authentication – *shareability*.

- This work identifies and empirically proves that our daily activity can be used in a meaningful way for authentication purpose.

- An integrated system is developed for evaluating the potential of proposed system which is capable of automatic activity collection, selection of potential activities for challenge generation and an interface for challenge response.

- Proposed system inherently provides security to any number of accounts with dynamic challenges and easy maintainability as users need to remember activities of recent past.

- User survey suggests that they are interested to use such system for password recovery instead of static (hint) question answer based password recovery.

- Our experiment shows that legitimate users can successfully pass through our authentication system in 95% cases whereas impostors can go through only for 5.5%

cases.

- We have made some important observations from this study, for example a) outlier activities do have more potential to be a good candidate of challenge generation, b) text based question with little hint is more effective, c) user can remember well last 2-3 days activities etc.

7.2 Future Direction

In this section we discuss some new research issues that have been opened up by this thesis.

7.2.1 Efficient Offload using WiFi Network

Some important future directions from this study are summarized as follows.

- This work does not consider the file selection strategy for caching and how the cache should be modified with time. So an important direction for an integrated system will be to focus on the file selection strategy which can be modulated based upon feedback from user.

- Feasibility of live streaming support with proposed framework can be studied.

- With this proposed framework, we want to study the feasibility in supporting live streaming and the solution

thereof.

- In a high traffic scenario, due to bandwidth division, clients suffer from low streaming rate. However, in this scenario, a P2P kind of strategy might boost up efficiency significantly. A future direction could be to study the performance of such system in high traffic scenario.

7.2.2 Managing Heterogeneous Traffic

Some important future directions from this study are summarized as follows.

- In this work, we have assumed very low mobility of users and a simplistic mobility protocol. More practical mobility model applicable for university campus and city may be studied.

- An anticipatory association based on mobility prediction will help in faster association. One can, in future, study the feasibility of this scheme and, subsequently explore the effect of such a strategy.

- Another important direction could be to study the effect of a sophisticated channel allocation strategy on reduction of interference during associa-

tion.

7.2.3 Restricting Unauthorized Traffic

Some important future directions from this study are summarized as follows.

- As a proof of concept we explored very few information sources; as a future step, one might explore more information sources like information from wearable devices, smartphone sensors etc.

- Another concept could be to develop architectures for utilizing proposed methodology in different scenarios of authentication like architecture for utilizing proposed system in user's own smartphone will be different from the architecture when it is going to be used by some third party.

- There is always a potential of privacy leaks as this system works with user's private data. So in future it would be important to investigate the methods of securing privacy in greater detail.

Bibliography

[1] Supplementary Materials. `http://tinyurl.com/SI-ComPAS`. [Online; accessed 24-Oct-2016].

[2] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3):409–418.

[3] C. C. Aggarwal, Y. Zhao, and S. Y. Philip. Outlier detection in graph streams. In *ICDE*, pages 399–409. IEEE, 2011.

[4] N. K. Ahmed, J. Neville, and R. Kompella. Reconsidering the foundations of network sampling. In *Proceedings of the 2nd Workshop on Information in Networks*, 2010.

[5] N. K. Ahmed, J. Neville, and R. Kompella. Network sampling: From static to streaming graphs. *ACM TKDD*, 8(2):7, 2014.

[6] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.

[7] R. Albert, H. Jeong, and A. Barabási. Internet: Diameter of the world-wide web. *Nature*, 401(6749):130–131, 1999.

[8] L. A. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152, October 2000.

[9] R. M. Anderson, R. M. May, and B. Anderson. *Infectious diseases of humans: dynamics and control*, volume 28. Wiley Online Library, 1992.

[10] Z. C. Ani, A. Yasin, M. Z. Husin, and Z. A. Hamid. A method for group formation using genetic algorithm. *International Journal on Computer Science and Engineering*, 2(09):3060–3064, 2010.

[11] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[12] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez. Motif-based communities in complex networks. *Journal of Physics A*, 41(22):224001, Sept. 2008.

[13] A. Arenas, A. Fernández, and S. Gómez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, 2008.

[14] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Trans. Knowl. Discov. Data*, 3(4):16:1–16:36, Dec. 2009.

[15] P. Bacchetti. Peer review of statistics in medical research: the other problem. *British Medical Journal*, 324(7348):1271, 2002.

[16] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 44–54, New York, NY, USA, 2006. ACM.

[17] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[18] M. J. Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6), 2007.

[19] E. R. Barnes. An algorithm for partitioning the nodes of a graph. Technical Report RC 08690, IBM US Research Centers (Yorktown,San Jose,Almaden, US), 1981.

[20] J. Baumes, M. K. Goldberg, M. S. Krishnamoorthy, M. Magdon-Ismail, and N. Preston. Finding communities by clustering a graph into overlapping subgraphs. In *IADIS AC*, pages 97–104. IADIS, 2005.

[21] M. Bawa, T. Condie, and P. Ganesan. Lsh forest: self-tuning indexes for similarity search. In *Proceedings of the 14th international conference on World Wide Web*, pages 651–660. ACM, 2005.

[22] V. Belak, S. Lam, and C. Hayes. Towards maximising cross-community information diffusion. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 171–178, Aug 2012.

[23] T. Y. Berger-Wolf and J. Saia. A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 523–528, New York, NY, USA, 2006. ACM.

[24] J. W. Berry, B. Hendrickson, R. A. LaViolette, V. J. Leung, and C. A. Phillips. *eprint arXiv:0710.3800*, 2007.

[25] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips. Tolerating the community detection resolution limit with edge weighting. *Physical Review E*, 83(5):056119, May 2011.

[26] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[27] G. Bianconi, P. Pin, and M. Marsili. Assessing the relevance of node features for network structure. *Proceedings of the National Academy of Sciences*, 106(28):11433–11438, 2009.

[28] 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*, 2008(10):P10008, 2008.

[29] S. Boettcher and A. G. Percus. Optimization with extremal dynamics. *Complex.*, 8(2):57–62, Nov. 2002.

[30] J. Bohannon. Who's afraid of peer review. *Science*, 342(6154), 2013.

[31] B. BollobÃąs. *Modern Graph Theory*. Graduate texts in mathematics. Springer, Heidelberg, corrected edition, 1998.

[32] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.

[33] R. D. Braatz. Papers receive more citations after rejection [publication activities]. *Control systems, IEEE*, 34(4):22–23, 2014.

[34] U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. In G. D. Battista and U. Zwick, editors, *ESA*, volume 2832 of *Lecture Notes in Computer Science*, pages 568–579. Springer, 2003.

[35] F. Breve, L. Zhao, and M. Quiles. Uncovering overlap community structure in complex networks using particle competition. In *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence*, AICI '09, pages 619–628, Berlin, Heidelberg, 2009. Springer-Verlag.

[36] H. Caswell. Improving peer review.

[37] R. Cazabet, F. Amblard, and C. Hanachi. Detection of overlapping communities in dynamical social networks. In *2010 IEEE Second International Conference on Social Computing (SocialCom)*, pages 309–314, Aug 2010.

[38] D. Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *PKDD*, volume 3202 of *Lecture Notes in Computer Science*, pages 112–124. Springer, 2004.

[39] T. Chakraborty, S. Kumar, P. Goyal, N. Ganguly, and A. Mukherjee. Towards a stratified learning approach to predict future citation counts. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 351–360. IEEE Press, 2014.

[40] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[41] C. Chatfield. *The analysis of time series: an introduction*. CRC press, 2013.

[42] D. Chen, M. Shang, Z. Lv, and Y. Fu. Detecting overlapping communities of weighted networks via a local algorithm. *Physica A: Statistical Mechanics and its Applications*, 389(19):4177–4187, 2010.

[43] J. Chen, H. Zhang, Z.-H. Guan, and T. Li. Epidemic spreading on networks with overlapping community structure. *Physica A: Statistical Mechanics and its Applications*, 391(4):1848–1854, 2012.

[44] M. Chen, T. Nguyen, and B. Szymanski. A new metric for quality of network community structure. *ASE Human Journal*, 1(4):226–240, 2013.

[45] W. Chen, Z. Liu, X. Sun, and Y. Wang. A game-theoretic framework to identify overlapping communities in social networks. *Data Min. Knowl. Discov.*, 21(2):224–240, Sept. 2010.

[46] W. Y. C. Chen, A. W. M. Dress, and W. Q. Yu. Community structures of networks. *Mathematics in Computer Science*, 1(3):441–457, 2008.

[47] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumour spreading and graph conductance. In *SODA*, pages 1657–1663, 2010.

[48] X. Chu, J. Guan, Z. Zhang, and S. Zhou. Epidemic spreading in weighted scale-free networks with community structure. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07043, 2009.

[49] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

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

[51] S. Cole, G. A. Simon, et al. Chance and consensus in peer review. *Science*, 214(4523):881–886, 1981.

[52] L. M. Collins and C. W. Dent. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research*, 23(2):231–242, 1988.

[53] J. Copic, M. O. Jackson, and A. Kirman. Identifying community structures from network data, 2005.

[54] L. Danon, A. Díaz-Guilera, and A. Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics*, 2006(11):P11010, 2006.

[55] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[56] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *ACM Symposium on Principles of distributed computing*, pages 1–12. ACM, 1987.

[57] D. A. Dickey and W. A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431, 1979.

[58] F. Ding, Z. Luo, J. Shi, and X. Fang. Overlapping community detection by kernel-based fuzzy affinity propagation. In *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on*, pages 1–4, May 2010.

[59] P. S. Dodds and D. J. Watts. Universal behavior in a generalized model of contagion. *Physical review letters*, 92(21):218701, 2004.

[60] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, 17(5):420–425, Sept. 1973.

[61] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[62] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72:027104, Aug 2005.

[63] J.-P. Eckmann and E. Moses. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the National Academy of Sciences*, 99(9):5825–5829, 2002.

[64] P. Erdös and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[65] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35(2):114–131, 2003.

[66] T. S. Evans. Clique graphs and overlapping communities. *CoRR*, abs/1009.0638, 2010.

[67] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, Aug. 1999.

[68] I. Farkas, D. Ábel, G. Palla, and T. Vicsek. Weighted network modules. *New Journal of Physics*, 9(6):180, 2007.

[69] M. Fatemi and L. Tokarchuk. A community based social recommender system for individuals amp; groups. In *2013 International Conference on Social Computing (SocialCom)*, pages 351–356, Sept 2013.

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

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

[72] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, Jan. 2007.

[73] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *PNAS*, Jan. 2007.

[74] S. Fortunato and A. Lancichinetti. Community detection algorithms: A comparative analysis. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pages 27:1–27:2, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[75] J. Fournet and A. Barrat. Contact patterns among high school students. 2014.

[76] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[77] Q. Fu and A. Banerjee. Multiplicative mixture models for overlapping clustering. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pages 791–796, Dec 2008.

[78] M. Gaertler, R. Görke, and D. Wagner. Significance-driven graph clustering. In *Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management*, AAIM '07, pages 11–26, Berlin, Heidelberg, 2007. Springer-Verlag.

[79] M. Girvan and M. E. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

[80] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Walking in facebook: a case study of unbiased sampling of osns. In *Infocom*, pages 1–9. IEEE, 2010.

[81] D. E. Golberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989:102, 1989.

[82] J. Graffy, R. Bryar, S. Kendall, and D. Crook. Improving peer review. *Primary Health Care Research and Development*, 7(01):1–2, 2006.

[83] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

[84] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 176–183, Aug 2010.

[85] S. Gregory. An algorithm to find overlapping community structure in networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 91–102, Berlin, Heidelberg, 2007. Springer-Verlag.

[86] S. Gregory. Finding overlapping communities using disjoint community detection algorithms. In S. Fortunato, G. Mangioni, R. Menezes, and V. Nicosia, editors, *Complex Networks*, volume 207 of *Studies in Computational Intelligence*, pages 47–61. Springer, Berlin / Heidelberg, 2009.

[87] R. Guimera, M. Sales-Pardo, and L. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004.

[88] R. Guimerà, M. Sales-Pardo, and L. Amaral. A network-based method for target selection in metabolic networks. *Bioinformatics*, 23(13):1616–1622, July 2007.

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

[90] 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)*, 28(1):100–108, 1979.

[91] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[92] M. B. Hastings. Community detection as an inference problem. *Phys. Rev. E*, 74:035102, Sep 2006.

[93] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[94] F. Havemann, M. H. 0003, A. Struck, and J. GlÃd'ser. Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. *CoRR*, abs/1012.1269, 2010.

[95] D. He, D. Liu, W. Zhang, D. Jin, and B. Yang. Discovering link communities in complex networks by exploiting link dynamics. *CoRR*, abs/1303.4699, 2013.

[96] A. Hlaoui and S. Wang. Median graph computation for graph clustering. *Soft Comput.*, 10(1):47–53, 2006.

[97] J. M. Hofman and C. H. Wiggins. A bayesian approach to network modularity, Sept. 2007.

[98] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[99] D. F. Horrobin. The philosophical basis of peer review and the suppression of innovation. *JAMA*, 263(10):1438–1441, 1990.

[100] Y. Hu, H. Chen, P. Zhang, M. Li, Z. Di, and Y. Fan. Comparative definition of community and corresponding identifying algorithm. *Phys. Rev. E*, 78:026121, Aug 2008.

[101] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[102] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[103] B. Hughes. *Random Walks and Random Environments: Random walks*. Number v. 1 in Oxford science publications. Clarendon Press, 1995.

[104] F. J. Ingelfinger. Peer review in biomedical publication. *The American J. of Med.*, 56(5):686–692, 1974.

[105] T. Jefferson, M. Rudin, S. Brodney Folse, and F. Davidoff. Editorial peer review for improving the quality of reports of biomedical studies. *The Cochrane Library*, 2006.

[106] T. Jefferson, E. Wager, and F. Davidoff. Measuring the quality of editorial peer review. *JAMA*, 287(21):2786–2790, 2002.

[107] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. BarabÃąsi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, October 2000.

[108] Z. Jin, J. Wang, S. Zhang, and Y. Shu. Epidemic-based controlled flooding and adaptive multicast for delay tolerant networks. In *Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2010 7th International Conference on*, pages 191–194. IEEE, 2010.

[109] J. Kamahara, T. Asakawa, S. Shimojo, and H. Miyahara. A community-based recommendation system to reveal unexpected interests. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*, pages 433–438, Jan. 2005.

[110] N. Kaplan. A generalization of a result of erdös and rényi. *Journal of Applied Probability*, pages 212–216, 1977.

[111] J. P. Kassirer and E. W. Campion. Peer review: crude and understudied, but indispensable. *JAMA*, 272(2):96–97, 1994.

[112] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Sys. Tech. J.*, 49(2):291–308, 1970.

[113] Y. Kim and H. Jeong. The map equation for link community. *CoRR*, abs/1105.0257, 2011.

[114] Y. Kim, S. W. Son, and H. Jeong. Link Rank: Finding communities in directed networks. Technical Report arXiv:0902.3728.

[115] M. Kimura, K. Yamakawa, K. Saito, and H. Motoda. Community analysis of influential nodes for information diffusion on a social network. In *IJCNN*, pages 1358–1363. IEEE, 2008.

[116] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[117] E. D. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer, 1st edition, 2009.

[118] I. A. KovÃącs, R. Palotai, M. S. Szalay, and P. Csermely. Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PLoS One*, page 12528, 2010.

[119] A. D. Kramer, J. E. Guillory, and J. T. Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.

[120] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. *Phys. Rev. E*, 78:026109, Aug 2008.

[121] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1):159–178, 1992.

[122] V. Labatut. Generalized measures for the evaluation of community detection methods. *CoRR*, abs/1303.5441, 2013.

[123] R. Lambiotte. Multi-scale modularity in complex networks. In *WiOpt*, pages 546–553. IEEE, 2010.

[124] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80:056117, Nov 2009.

[125] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Phys. Rev. E*, 84:066122, 2011.

[126] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[127] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[128] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 04 2011.

[129] A. Lázár, D. Ábel, and T. Vicsek. Modularity measure of networks with overlapping communities. *Europhys. Lett.*, 90(1):18001, 2010.

[130] C. Lee, F. Reid, A. Mcdaid, and N. Hurley. Detecting highly overlapping community structure by greedy clique expansion. In *In Proceedings of the 4th Workshop on Social Network Mining and Analysis, (SNA/KDD10)*, pages 33–42, Aug 2010.

[131] S. Lehmann and L. K. Hansen. Deterministic modularity optimization. *The European Physical Journal B*, 60(1):83–88.

[132] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 100:118703, Mar 2008.

[133] S. Leonardi, A. Panconesi, P. Ferragina, and A. Gionis, editors. *Rumor Spreading in Random Evolving Graphs*. ACM, February 2013.

[134] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *SIGKDD*, pages 631–636. ACM, 2006.

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

[136] F. Li, J. He, G. Huang, Y. Zhang, and Y. Shi. A clustering-based link prediction method in social networks. *Procedia Computer Science*, 29(0):432–442, 2014. 2014 International Conference on Computational Science.

[137] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 685–694, New York, NY, USA, 2008. ACM.

[138] P. Lisboa, H. AL-Mamory, and B. W. Timproving recommendation systems by modeling the stability of implicit behaviour. In *The Post Graduate Network Symposium (PGNet2013)*, pages 354–361, Dec 2013.

[139] R. Lo Cigno, A. Russo, and D. Carra. On some fundamental properties of p2p push/pull protocols. In *Communications and Electronics, 2008. ICCE 2008. Second International Conference on*, pages 67–73. IEEE, 2008.

[140] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Relational clustering by symmetric convex coding. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 569–576, New York, NY, USA, 2007. ACM.

[141] L. Lü, C.-H. Jin, and T. Zhou. Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, 80(4):046122, 2009.

[142] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[143] B. Luëar, Z. LevnajiÄĞ, J. Povh, and M. Perc. Community structure and the evolution of interdisciplinarity in slovenia's scientific collaboration network. *PLoS ONE*, 9(4):e94429, Dec. 2014.

[144] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[145] M. Magdon-Ismail and J. T. Purnell. Ssde-cluster: Fast overlapping clustering of networks using sampled spectral distance embedding and gmms. In *SocialCom/PASSAT*, pages 756–759. IEEE, 2011.

[146] A. S. Maiya and T. Y. Berger-Wolf. Sampling community structure. In *WWW*, pages 701–710. ACM, 2010.

[147] C. D. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval, 2008.

[148] C. P. Massen and J. P. K. Doye. Identifying communities within energy landscapes. *Physical Review E*, 71(046101), 2005.

[149] R. A. McNutt, A. T. Evans, R. H. Fletcher, and S. W. Fletcher. The effects of blinding on the quality of peer review: a randomized trial. *JAMA*, 263(10):1371–1376, 1990.

[150] D. Mollison. *Epidemic models: their structure and relation to data*, volume 5. Cambridge University Press, 1995.

[151] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Struct. Algorithms*, 6(2/3):161–179, Mar. 1995.

[152] A. Montejo-Ráez, E. Martínez-Cámara, M. T. Martin-Valdivia, and L. A. Urena-Lopez. Random walk weighting over sentiwordnet for sentiment polarity detection on twitter. In *WASSA*, 2012.

[153] J. Moreno, D. A. Ovalle, and R. M. Vicari. A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics. *Computers & Education*, 58(1):560–569, 2012.

[154] C. Musto, P. Lops, F. Narducci, G. Semeraro, and M. D. Gemmis. A tag recommender system exploiting user and community behavior.

[155] B. Nadler and M. Galun. Fundamental limitations of spectral clustering methods. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[156] A. Nematzadeh, E. Ferrara, A. Flammini, and Y.-Y. Ahn. Optimal network modularity for information diffusion. *Phys. Rev. Lett.*, 113:088701, Aug 2014.

[157] T. Nepusz, A. PetrÃşczi, L. NÃl'gyessy, and F. BazsÃş. Fuzzy communities and the concept of bridgeness in complex networks. E-print, 2007.

[158] M. E. Newman. Analysis of weighted networks. *Physical review E*, 70(5):056131, 2004.

[159] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[160] M. E. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[161] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404–409, January 2001.

[162] M. E. J. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70(5):056131, Nov. 2004.

[163] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330, Mar. 2004.

[164] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, Jun 2004.

[165] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007.

[166] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[167] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, page 6, 2013.

[168] G. Palla, A.-L. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, April 2007.

[169] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[170] G. Palla, I. DerÃl'nyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.

[171] Y. Pan, D.-H. Li, J.-G. Liu, and J.-Z. Liang. Detecting community structure in complex networks via node similarity. *Physica A: Statistical Mechanics and its Applications*, 389(14):2849–2857, 2010.

[172] S. Papadopoulos, A. Skusa, A. Vakali, Y. Kompatsiaris, and N. Wagner. *eprint arXiv:0902.08*, 2009.

[173] P. Patarasuk, X. Yuan, and A. Faraj. Techniques for pipelined broadcast on ethernet switched clusters. *Journal of Parallel and Distributed Computing*, 68(6):809–824, 2008.

[174] M. Pearson. Drifting smoke rings: Social network analysis and markov processes in a longitudinal study of friendship groups and risk-taking, connections, 2003.

[175] J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth. The development and psychometric properties of liwc2007, 2007.

[176] C. Peterson and J. R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[177] A.-K. Pietilainen and C. Diot. CRAWDAD trace thlab/sigcomm2009/mobiclique/proximity (v. 2012-07-15). July 2012.

[178] J. W. Pinney and D. R. Westhead. Betweenness-based decomposition methods for social and biological networks. In S. Barber, P. Baxter, K. Mardia, and R. Walls, editors, *Interdisciplinary Statistics and Bioinformatics*, pages 87–90. Leeds University Press, 2007.

[179] P. Pons and M. Latapy. Computing communities in large networks using random walks. In p. Yolum, T. Güngör, F. Gürgen, and C. Özturan, editors, *Computer and Information Sciences - ISCIS 2005*, volume 3733 of *Lecture Notes in Computer Science*, pages 284–293. Springer Berlin Heidelberg.

[180] P. Pons and M. Latapy. Computing communities in large networks using random walks. In *International Symposium on Computer and Information Sciences*, pages 284–293. Springer, 2005.

[181] A. Pothen. Graph partitioning algorithms with applications to scientific computing. Technical report, Norfolk, VA, USA, 1997.

[182] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76:036106, Sep 2007.

[183] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[184] A. H. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, and D. Stutzbach. Respondent-driven sampling for characterizing unstructured overlays. In *INFOCOM*, pages 2701–2705. IEEE, 2009.

[185] M. J. Rattigan, M. Maier, and D. Jensen. Graph clustering with network structure indices. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 783–790, New York, NY, USA, 2007. ACM.

[186] J. Reichardt and S. Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett.*, 93(21):218701, Nov. 2004.

[187] A. S. Relman and M. Angell. How good is peer review? *The New Eng. J. of Med.*, 321(12):827–829, 1989.

[188] W. Ren, G. Yan, X. Liao, and L. Xiao. Simple probabilistic algorithm for detecting community structure. *Phys. Rev. E*, 79:036111, Mar 2009.

[189] D. Rennie. Editorial peer review in biomedical publication: the first international congress. *JAMA*, 263(10):1317–1317, 1990.

[190] B. Ribeiro and D. Towsley. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 390–403. ACM, 2010.

[191] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *arXiv preprint cs/0209028*, 2002.

[192] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327–7331, 2007.

[193] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *PNAS*, 104(18):7327–7331, 2007.

[194] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[195] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 105(4):1118–1123, 2008.

[196] M. Sachan and R. Ichise. Using abstract information and community alignment information for link prediction. In *2010 Second International Conference on Machine Learning and Computing (ICMLC)*, pages 61–65, Feb 2010.

[197] S. Sanghavi, B. Hajek, and L. Massoulié. Gossiping with multiple messages. *IEEE Transactions on Information Theory*, 53(12):4640–4654, 2007.

[198] P. Schuetz and A. Caflisch. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 77(4):046112, 2008.

[199] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29). May 2009.

[200] J. Shang, L. Liu, F. Xie, and C. Wu. How overlapping community structure affects epidemic spreading in complex networks. In *2014 IEEE 38th International Computer Software and Applications Conference Workshops (COMPSACW)*, pages 240–245, July 2014.

[201] H. Shen, X. Cheng, K. Cai, and M.-B. Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009.

[202] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, Aug. 2000.

[203] S. Sikdar, M. Marsili, N. Ganguly, and A. Mukherjee. Anomalies in the peer-review system: A case study of the journal of high energy physics. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2245–2250. ACM, 2016.

[204] R. Smith. Peer review: a flawed process at the heart of science and journals. *Journal of the R. Soc. of Med.*, 99(4):178–182, 2006.

[205] S.-W. Son, G. Bizhani, C. Christensen, P. Grassberger, and M. Paczuski. Percolation theory on interdependent networks based on epidemic spreading. *EPL (Europhysics Letters)*, 97(1):16006, 2012.

[206] S.-W. Son, H. Jeong, and J. D. Noh. Random field ising model and community structure in complex networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 50(3):431–437.

[207] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. Monic: Modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 706–711, New York, NY, USA, 2006. ACM.

[208] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: Parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 687–696, New York, NY, USA, 2007. ACM.

[209] S. R. Sundaresan, I. R. Fischhoff, J. Dushoff, and D. I. Rubenstein. Network metrics reveal differences in social organization between two fission–fusion species, Grevy's zebra and onager. *Oecologia*, 151(1):140–149, Feb. 2007.

[210] S. Sur, N. Ganguly, and A. Mukherjee. Attack tolerance of correlated time-varying social networks with well-defined communities. *Physica A: Statistical Mechanics and its Applications*, 2014.

[211] T. Takaguchi, N. Masuda, and P. Holme. Bursty communication patterns facilitate spreading in a threshold-based epidemic dynamics. *PloS one*, 8(7):e68629, 2013.

[212] J. Tang, S. Scellato, M. Musolesi, C. Mascolo, and V. Latora. Small-world behavior in time-varying graphs. *Physical Review E*, 81(5), May 2010. 055101(R).

[213] M. Tang, Z. Liu, and B. Li. Epidemic spreading by objective traveling. *EPL (Europhysics Letters)*, 87(1):18005, 2009.

[214] C. Tantipathananandh and T. Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 827–836, New York, NY, USA, 2009. ACM.

[215] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 717–726, New York, NY, USA, 2007. ACM.

[216] G. Tibély and J. Kertész. On the equivalence of the label propagation method of community detection and a potts model approach. *Physica A: Statistical Mechanics and its Applications*, 387(19–20):4982–4984, 2008.

[217] C. Tong, Y. Lian, J. Niu, Z. Xie, and Y. Zhang. A novel green algorithm for sampling complex networks. *Journal of Network and Computer Applications*, 59:55–62, 2016.

[218] S. Trajanovski, S. Scellato, and I. Leontiadis. Error and attack vulnerability of temporal networks. *Physical Review E*, 85(6):066105, 2012.

[219] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman. E-mail as a spectroscopy: Automated discovery of community structure within organizations. In M. Huysman, E. Wenger, and V. Wulfs, editors, *Proceedings of the First Iternational Conference on Communities and Technologies*, 2003.

[220] J. Valverde-Rebaza and A. de Andrade Lopes. Structural link prediction using community information on twitter. In *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, pages 132–137, Nov 2012.

[221] J. C. Valverde-Rebaza and A. A. Lopes. Link prediction in online social networks using group information. volume 8584, pages 31 – 45, Portugal, 2014. Springer.

[222] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[223] P. Vanhems, A. Barrat, C. Cattuto, J.-F. Pinton, N. Khanafer, C. Régis, B.-a. Kim, B. Comte, and N. Voirin. Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PloS one*, 8(9):e73970, 2013.

[224] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA, 2009. ACM.

[225] E. Volz and L. A. Meyers. Susceptible–infected–recovered epidemics in dynamic contact networks. *Proceedings of the Royal Society of London B: Biological Sciences*, 274(1628):2925–2934, 2007.

[226] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks: [extended abstract]. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1275–1276, New York, NY, USA, 2007. ACM.

[227] A. Wald. On cumulative sums of random variables. *The Annals of Mathematical Statistics*, 15(3):283–296, 1944.

[228] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

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

[230] D. J. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, 2002.

[231] J. Watts and R. Van De Geijn. A pipelined broadcast for multidimensional meshes. *Parallel Processing Letters*, 5(02):281–292, 1995.

[232] Y.-C. Wei and C.-K. Cheng. Towards efficient hierarchical designs by ratio cut partitioning. In *Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers., 1989 IEEE International Conference on*, pages 298–301. IEEE, Nov. 1989.

[233] L. Weng, F. Menczer, and Y.-Y. Ahn. Virality prediction and community structure in social networks. *Sci. Rep.*, 3(2522), 2013.

[234] R. West, H. S. Paskov, J. Leskovec, and C. Potts. Exploiting social network structure for person-to-person sentiment analysis. *arXiv preprint arXiv:1409.2450*, 2014.

[235] R. Winkler. *An Introduction to Bayesian Inference and Decision*. International series in decision processes. Holt, Rinehart and Winston, 1972.

[236] F. Y. Wu. The potts model. *Rev. Mod. Phys.*, 54(1):235–268, Jan. 1982.

[237] Z. Wu, Y. Lin, H. Wan, and S. Tian. A fast and reasonable method for community detection with adjustable extent of overlapping. In *2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 376–379, Nov 2010.

[238] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):43:1–43:35, Aug. 2013.

[239] J. Xie, B. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW)*, pages 344–349, Dec 2011.

[240] R. Yan, C. Huang, J. Tang, Y. Zhang, and X. Li. To better stand on the shoulder of giants. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pages 51–60. ACM, 2012.

[241] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. *KIAS*, 42(1):181–213, 2015.

[242] Z. Ye, S. Hu, and J. Yu. Adaptive clustering algorithm for community detection in complex networks. *PRE*, 78:046115, Oct 2008.

[243] M. Zarei, D. Izadi, and K. A. Samani. Detecting overlapping community structure of networks based on vertex–vertex correlations. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(11):P11013, 2009.

[244] M. Zarei and K. A. Samani. Eigenvectors of network complement reveal community structure more accurately. *Physica A: Statistical Mechanics and its Applications*, 388(8):1721–1730, 2009.

[245] S. Zhang, R.-S. Wang, and X.-S. Zhang. Identification of overlapping community structure in complex networks using fuzzy -means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007.

[246] S. Zhang, R.-S. Wang, and X.-S. Zhang. Uncovering fuzzy community structure in complex networks. *Phys. Rev. E*, 76:046103, Oct 2007.

[247] Y. Zhang, J. Wang, Y. Wang, and L. Zhou. Parallel community detection on large networks with propinquity dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 997–1006, New York, NY, USA, 2009. ACM.

[248] H. Zhou. Distance, dissimilarity index, and network community structure. *Physical Review E*, 67(6):061901, 2003.

[249] L. Zhuhadar, R. Yang, and O. Nasraoui. Toward the design of a recommender system: Visual clustering and detecting community structure in a web usage network. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 354–361, Dec 2012.

Appendix A

List of all Publications by the Candidate

Following is a list of all the publications by the candidate including those on and related to the work presented in the thesis. The publications are arranged in chronological order and in four sections – (i) journals, (ii) conferences, and (iii) workshops/research colloquiums.

Journals

1. **Sandipan Sikdar**, Matteo Marsili and Animesh Mukherjee. "Unsupervised Ranking of Clustering Algorithms by INFOMAX". (communicated to CACM)

2. Marcin Bodych, Niloy Ganguly, Tyll Kruger, Animesh Mukherjee, Rainer Seigmund-Schultze and **Sandipan Sikdar**. "Threshold based epidemic dynamics in systems with memory", *Europhysics Letters (EPL), 116.4(2017):48004*.

3. **Sandipan Sikdar**, Niloy Ganguly, Animesh Mukherjee. " Time series analysis of temporal networks", *European Physics Journal B topical issue on Temporal Network Theory and Applications (EPJB 2016), volume 89(1), 1-11, DOI: 10.1140/epjb/e2015-60654-7*.

4. Tanmoy Chakraborty, **Sandipan Sikdar**, Niloy Ganguly, Animesh Mukherjee. " Citation Interactions among Computer Science Fields: A Quantitative Route to the Rise and Fall of Scientific Research", *Social Network Analysis and Mining (SNAM 2014), Springer, 4:187, pp. 1-18, DOI 10.1007/s13278-014-0187-3*.

Conferences

1. **Sandipan Sikdar**, Tanmoy Chakraborty, Soumya Sarkar, Niloy Ganguly and Animesh Mukherjee. " ComPAS: Community Preserving Sampling for Streaming Graphs". (communicated)

2. **Sandipan Sikdar**, Nitesh Sekhar, Matteo Marsili, Niloy Ganguly and Animesh Mukherjee. " On the effectiveness of multiple reviewers in a peer-review system: A case study of two high impact Physics journals". (communicated)

3. Soumya Sarkar, **Sandipan Sikdar**, Animesh Mukherjee and Sanjukta Bhowmick. "Using Core-Periphery Structure to Predict High Centrality Nodes in Time-Varying Networks". (communicated)

4. **Sandipan Sikdar**, Matteo Marsili, Niloy Ganguly and Animesh Mukherjee. " Influence of Reviewer Interaction Network on Long-term Citations: A Case Study of the Scientific Peer-Review System of the Journal of High Energy Physics", *IEEE/ACM Joint Conference on Digital Libraries (JCDL), Toronto, Canada, 2017*.

5. **Sandipan Sikdar**, Matteo Marsili, Niloy Ganguly and Animesh Mukherjee. " Anomalies in the peer-review system: A case study of the journal of High Energy Physics", *ACM International Conference on Information and Knowledgement Management (CIKM), Indianapolis, USA, 2016*.

6. **Sandipan Sikdar**, Abhijnan Chakraborty, Anshit Choudhury, Gourav Kumar, S. Kumar, Abhijeet Patil, Niloy Ganguly and Animesh Mukherjee. " Identifying and Characterizing Sleeping Beauties on YouTube", *ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW) 2016, San Francisco Poster highlights*.

Workshops/ Research Colloquium

1. **Sandipan Sikdar**. " Significance of scientific peer-review system: A case study of the Journal of High Energy Physics". *JCDL Doctoral Consortium 2017*.

2. **Sandipan Sikdar**, Marcin Bodych, Rajib Ranjan Maity, Biswajit Paria, Niloy Ganguly, Tyll Kruger, Animesh Mukherjee. " On segmented message broadcast in dynamic networks", *IEEE INFOCOM workshop (Netscicom), HongKong, 2015*.

3. Tanmoy Chakraborty, **Sandipan Sikdar**, Niloy Ganguly, Animesh Mukherjee, " Shift of Research Focus in Computer Sciences over the Last Fifty Years: What Citation Analysis Reflects?" *Poster in TechVista organized by Microsoft Research India, Coimbatore, Tamil Nadu, on January 24, 2013 (Awarded Honorable mention award)*.