Social Network Analysis of Mobile Streaming Networks

Shazia Tabassum
LIAAD, Inesctec
University of Porto
Porto, Portugal
Email: up201402360@fe.up.pt

***Abstract*—Call data generated from mobile phones reflect a social network structure. Analyzing the topology, behavior and dynamics of such networks is one of the prevailing interests in network science. We propose to analyze call networks as a spatio-temporal evolutionary stream. Initially, we explored some of the dynamics of call activity in evolving call networks. To overcome the space and time limitations of analyzing massive call networks, we made use of sampling algorithms to generate samples in real-time. We also discussed sampling at a precise level of socio-centric and ego-centric network. We delineated and evaluated some sampling methods and algorithms. Analyzed the properties of evolutionary call network and proposed some potential contributions in the realm of sampling and exploring activity patterns. We also discussed some prospective aspects of influence analysis such as family influence.**

1. Introduction

Mobile data is generated from a number of wireless sensing and GPS enabled devices, regnant are mobile phones. One of the fastest evolving data generating from these mobile devices is Call data. Besides being spatio-temporal in nature this data arrives continuously at high speed and volume proportional to the number of devices. Batch processing such data requires high cost in data warehousing etc. Furthermore, the results may get antiquated. Stream processing ceaselessly manipulates high speed data, while maintaining the latest results. Ergo, we consider streaming processing is an exemplary way of processing high velocity data. It is one of the major challenges of data mining community to learn from changing and evolving nature of high velocity streams in real-time. Therefore, processing streams typically require real-time incremental analytical methods.

Over the past decade, researchers are interested in building a virtual world of connections, relationships, and interactions of real world entities to study the complexities, behavior and dynamics of networks formed by them. Most of the works in social network analysis are based on social networking applications, with a petty work in the field of call networks. Our work would focus on social network analysis of mobile data streams such as call network. We would focus on methods to analyze the evolution of high velocity networked data, to deduce actionable patterns.

In the next section, we present some interesting results and discuss some potential contributions in the scope of understanding the dynamics of call activity patterns (section 2.1), sampling socio-centric networks (section 2.2), sampling ego-centric networks (section 2.3) and influence analysis (section 2.4). In section 3 we derive some conclusions and discuss future work.

2. Potential Contributions

2.1. Analysis of Call Activity Graph

We made use of an anonymised temporal call stream of 300 million calls over 31 days made by 11 million subscribers. On an average 10 to 280 calls are made during mid-night and mid-day respectively. We modeled telecommunication call graphs as nodes corresponding to callers and callees. The edges between them represent calls. We weighted our samples based on frequency of calls.

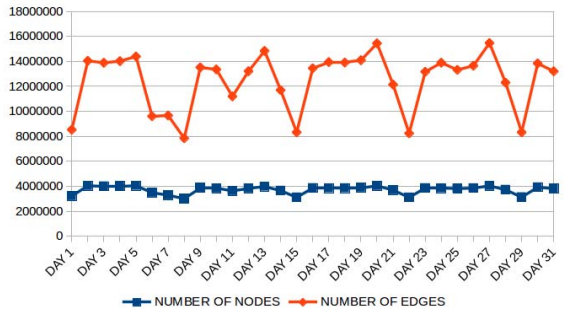

Figure 1. Evolution of nodes and edges in call graph stream

The snapshots of number of nodes and edges at the end of each day, as the stream evolves over 31 days are shown in Figure 1. The plot exhibits a high call activity on weekdays and decreased call activity on weekends comparatively. We also observe peeks on Fridays. Some users are active on weekends while others are inactive. Some decreased call activity on weekdays maps to the real-world holidays. We can also detect real-time events based on the real-time

2375-0324/16 $31.00 © 2016 IEEE
DOI 10.1109/MDM.2016.84

streaming call activity patterns. As we exemplified above, with the help of call activity patterns alone we can be able to apprehend some insightful social behaviors from the real world. When this kind of temporal network combined with the spatial information can provide remarkable acumen in the day to day activities of users/entities. The real-time streaming scenario would make it more powerful.

2.2. Sampling Socio-Centric Network

Sampling is the process of selecting a subgraph from an original graph to represent the characteristics of it, at a given point of time. As real-time mobile streaming networks are temporal, unbounded and huge to fit in memory it is difficult to analyze them with a commodity machine. We need real-time evolving samples that can represent the huge networks, while maintaining a trade-off between accuracy of results and cost of computation over huge networks. What if we have a real-time evolving sample of the stream with the similar properties and topology of the original graph? There are a number of algorithms proposed for sampling of streams [1], [2], [3], and [4] etc. However, there are no solutions to match all the properties of graphs. If the sample matches few properties, which sample would yield proper estimates for directed and weighted evolutionary graphs? In this section, we refer to the work we carried out in [5]. We implemented three sequential algorithms, space saving [6], reservoir sampling [1] and a biased random sampling algorithm [5] to generate sample streams in real time. These algorithms are briefed below.

2.2.1. Space Saving Algorithm. The Space Saving Algorithm (SSA) [6] is the most approximate and efficient algorithm for finding top frequent elements from the stream. The algorithm maintains partial information of interest as it monitors only a subset of elements from the stream. It maintains counters for every element in the sample and increments its count when the element re-occurs in the stream. If a new element is encountered in the stream it is replaced with an element with the least counter value and its count is incremented.

2.2.2. Reservoir Sampling. This is a well known algorithm of Reservoir Sampling (RS), denoted as Algorithm R in [1]. The author mentioned in his work that all the algorithms using a reservoir of elements from the original data to generate samples are a kind of reservoir sampling. In algorithm R the author maintained a reservoir of elements with a predefined sample size. In the streaming scenario, initially the reservoir is filled with the initial elements from the stream. Every element after that, is computed for the probability of being inserted and a random number is generated to pick an element already in the sample. If the probability of the new element is greater than the probability of the selected element then the new element replaces and old one, if not it is discarded. By the end every element in the sample is selected with equal probability. Consequently, the items are inserted into the reservoir with decreasing probability. Therefore, it leads to samples with very old items from the stream, as also discussed in [4].

2.2.3. Biased Random Sampling. We have known the random sampling techniques with all the elements in the sample with equal probability. In this section we present a biased random sampling technique which we have discussed in [5] where we sample items/objects with unequal probability. Biased Random Sampling (BRS) is based on the idea of reservoir sampling but it ensures that every item in the stream definitely enters the reservoir. As a general initial step, the reservoir is filled with the first items from the stream. Then, we do not compute the probability of later items, as every item definitely enters stream. For replacing an item already in the sample, a random number r is generated, where $1 \leq r \leq sizeofreservoir$. The element at the position of random number is replaced with the new item from stream. Here the probability of every item entering the reservoir is equal as every item enters the stream, but the probability of every item in the reservoir is not equal. Hence, this technique is biased towards new items from the stream. It can be used in the scenarios where old items are considered stale or not useful.

Below we present two methods we used for implementing the above algorithms. One is the node based method and the other is an edge based method.

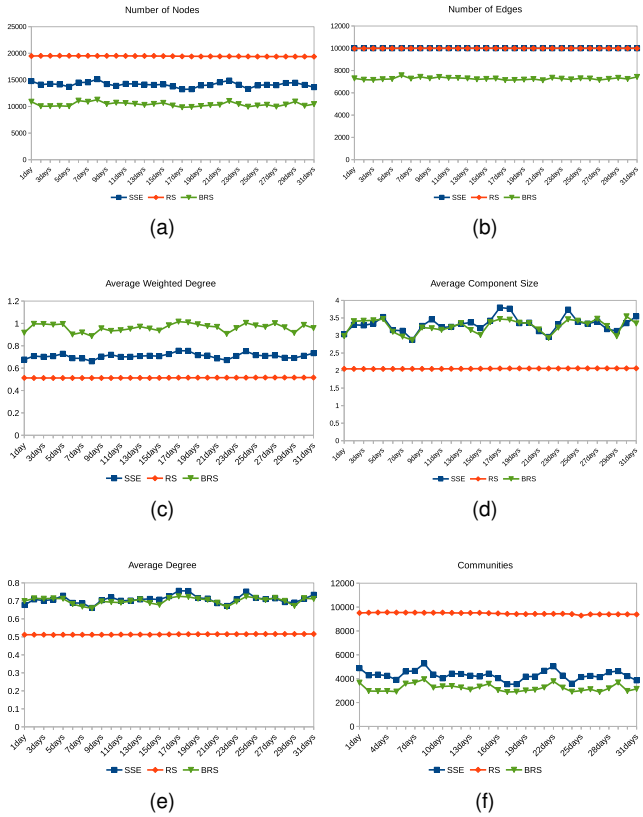

Figure 2. Graph metrics over samples

2.2.4. Node Based Method. Node based methods in general, sample a set of nodes from the original graph. The resultant samples contain a set of vertices from the graph stream and showing no connections between them. The samples posses only nodes and no structure, therefore we also acquire the corresponding edges of nodes in real time. As the number of edges incident upon the sampled nodes increases, so are the adjacent nodes. As a result, we have a subgraph with increased number of nodes, derived from the associated edges. The time for computation of such methods also increases substantially with the added time for acquiring edges and their corresponding nodes.

2.2.5. Edge Based Method. As the name suggest, these samples are generated by selecting a subset of edges from the original graph. The resultant graph is a subgraph of original graph with nodes and edges. The algorithms that can be implemented in the node based method, can also be implemented using edge based method in our scenario.

For conducting experiments we used the call network stream described in section 2.1. We employed a number of graph metrics over the sample snapshots at the end of 31 days streams, which are depicted in figure 2 and 3. From the above experiments we observe that, RS is biased to nodes with low degree centralities and BRS and SSA nodes exhibit higher degree centralities compared to it. BRS best suits for measuring weighted centralities based on frequency of edges. Hence, it is also suitable for running real-time queries for finding frequent items over the sample. BRS and SSA sample communities with high average degree centralities that shows a better community structure when compared to RS. Therefore BRS and SSA would be more suitable for applications exploring community structure. SSA and BRS generates samples with better component structure compared to RS. RS and BRS has good performance with runtime compared to SSA. For using samples to run queries like top frequent items, SSA would be appropriate as it samples top frequent edges, while not considering other factors.

The above results suggests the biases of the implemented algorithms in generating samples when compared to each other and the topology of original graph. Nevertheless there is more scope for comparing the above samples with the ground truth of the original network stream at any point of time, which requires highly scalable techniques. One of the prospective work is to determine what percentage of samples yields what percentage of accurate results. Furthermore the network is always growing and we need to generate relatively growing samples.

2.3. Sampling Ego-Centric Network

An ego centric network maps the relationships of an ego with alters and also between themselves. In [7], [8], [9], [10] the authors discuss the importance and properties of ego network. [11] proposed an ego-centric network sampling approach for viral marketing applications. The authors employed a variation of forest fire algorithm for sampling

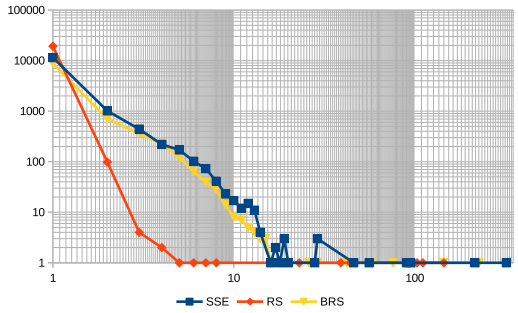

Figure 3. Degree distribution of samples

ego network. They compared the degree and clustering coefficient distribution of sampled ego networks with the original ego network.

Capturing the ego networks of high velocity streaming graphs over a period of time is highly infeasible as it can reach over millions of redundant edges for an active user in few days. Sampling techniques are generally used to create representative specimens of large scale socio-centric temporal networks. In this section, we refer to our edge based sampling method with forgetting factor over an evolving ego network (SEFF) introduced in [12]. SEFF method samples ego networks as they evolve, while maintaining the freshness of the ego network, with the latest ties and most stronger relationships from past, based on an attenuation factor. We also made use of an exhaustive list of node level and graph level metrics to evaluate and compare the samples with the original network.

2.3.1. Sampling Ego Network with Forgetting Factor. This method starts by building an ego network of specific ego and begin to scrape together all the adjacent ties to the ego and their adjacent ties (depth=2). This is done by using a set for storing adjacent nodes. For every recurring edge, the edge weight of the corresponding edge is incremented by maintaining in a hash table. A forgetting factor is imposed over edges, following successive grace periods. In our experiments, we use a grace period of 1 day. This means we apply the forgetting factor over the ego network as soon as the stream enters a new day, i.e we forget the old edges each of a kind (i.e edges between a pair of nodes), by some fixed percentage defined by the forgetting factor. The forgetting factor is given by two parameters, an attenuation factor α and a threshold θ. Where $0 < \alpha < 1$ and also $0 < \theta < 1$. After every grace period or update time t the tie strength between two nodes is given by the equation 1.

$$w_t = w_t + (1 - \alpha)w_{t-1} \tag{1}$$

where w_t is the tie strength between any two nodes in the ego network at time t. After every successive grace period, the edge weight is decreased by α and consequently the alter/alters adjacent to the corresponding edge are removed if the edge weight decreases than the threshold value θ. α=1 gives maximum forgetting i.e it forgets the whole

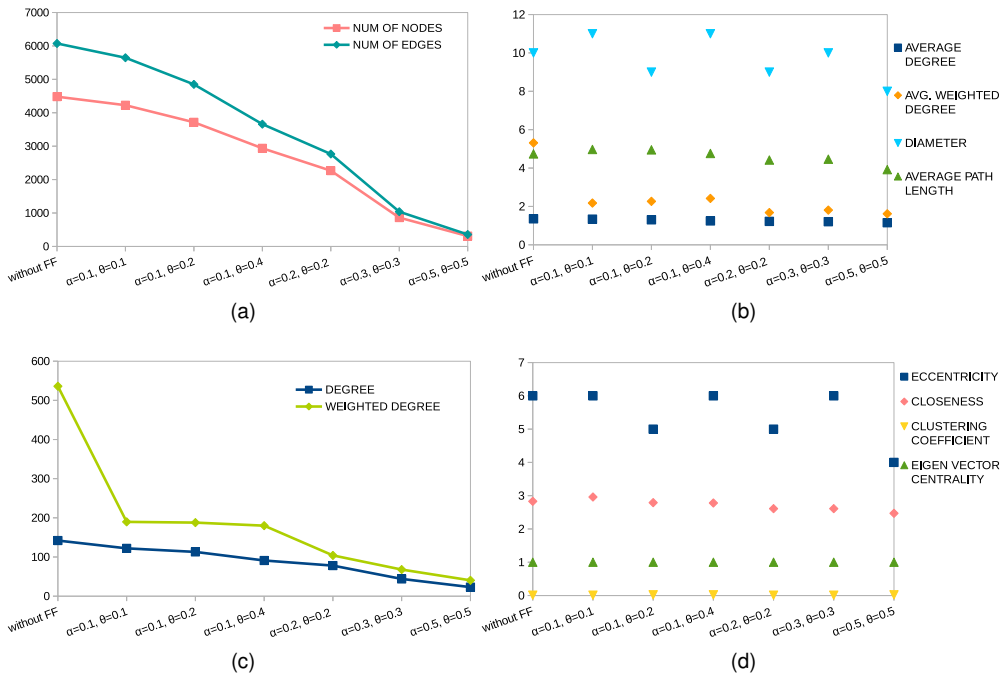

Figure 4. Metrics over ego networks with and without forgetting factor

network except the network of current day. $\alpha = 0$ gives the original network. If the removed edge corresponds to an alter adjacent to the ego, the adjacent alter gets removed, along with the second level alters adjacent to the alter itself, if the above condition is satisfied. If the removed edge is a second level edge, not having a direct connection to ego then the corresponding node alone is removed. Following this strategy, we get the most active alters in the ego network at the end of each day.

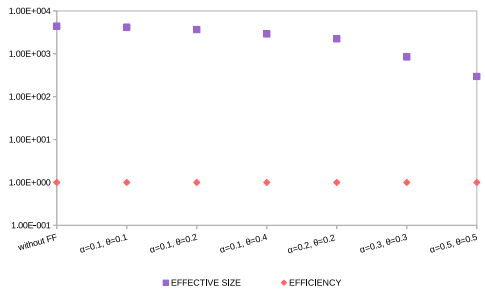

Figure 5. Efficiency and effective size of ego networks

2.3.2. Experimental Evaluation. The experiments are conducted using the real-world stream of call network described in section 2.1. Sample streams are generated over 31 days for different values of α and θ. We made use of an extensive list of metrics to compare the sample snapshots with the original network as shown in Figures 4 and 5. We observed that the SEFF method preserves the efficiency of network and importance of ego while decreasing redundancy in the network.

As a proposed work, we intend to analyze the evolution of ego network over a time period of 31 days or more and compare the evolving samples with the degree distributions of all the nodes in an ego network. The prospective work also involves analyzing the evolution of a number of egos with different properties such as an ego with high degree centrality, betweenness , closeness etc.

2.4. Influence Analysis

Discovering sets of key players and analyzing their influence is also one of the vital problems in social network analysis. In this phenomenon some nodes can have intrinsically higher influence than others due to network structure. The global measures are often associated with nodes in the network rather than edges. The edges are rather associated with the strength of relationships between nodes. [13] discusses the importance of Influential analysis. We would consider the edge based and node based measures to analyze the influence, such as determining the pivotal nodes with top tie strengths as shown in Figure. 6. In this figure, we depict a sample snapshot of 10^4 top frequent edges using [6] algorithm from an evolving call network at the end of 31 days. The above graph is illustrated using Fruchterman-Reingold layout algorithm [14]. We can observe few dense connections in the center of the graph and many sparsely connected nodes at the periphery. Additionally, we propose to analyze the cascading behavior and strength of mutual exchange of information between nodes.

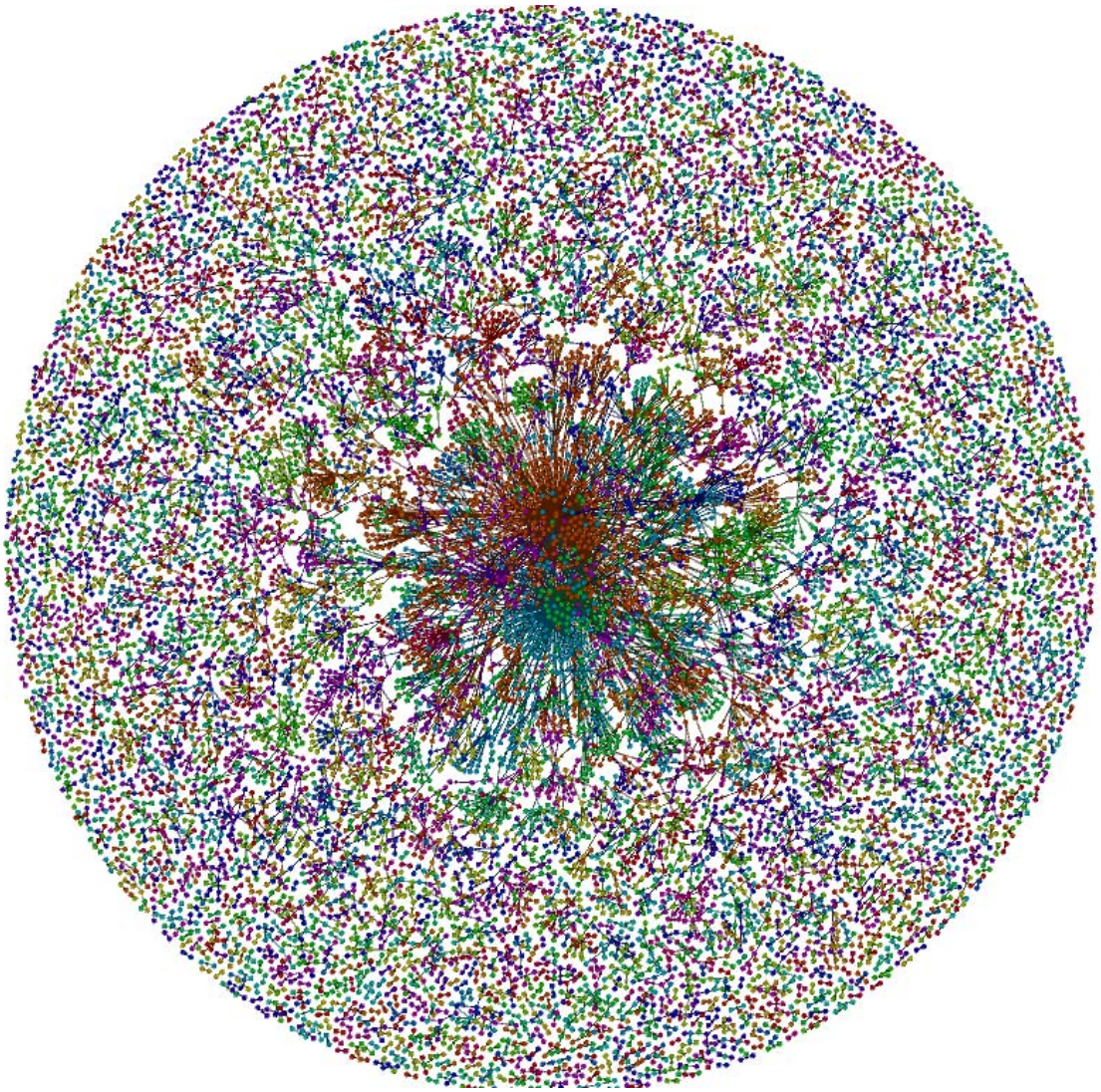

Figure 6. Sample of top frequent 10^4 edges

We also propose to analyze a novel aspect of behavioral influence, i.e. family influence, where people in a family are influenced by each other. A family can be people sharing common geographical location. This aspect can be applied to churn prediction in call networks. Where one node quitting the network greatly influences other nodes in its family. Analysis of user preferences in mobility networks is also a part of our future works.

3. Conclusion

We have discussed above our potential contributions in the area of mobile streaming networks' analysis. We analyzed the call network to understand and extrapolate the behaviors of users. We discussed how the temporal and spatial nature of mobility networks will be helpful in detecting the real-time events and activities. We presented some results and outlined possible interesting perspectives in the sphere of sampling and Influence analysis. We identified some potential endeavors, such as implementing and developing techniques for generating evolving samples, evaluating the samples by comparing the distributions of node measures with the ground-truth, analyzing the streaming ego networks for different positional nodes and examining the behavioral influence and information diffusion between nodes. We have also briefed some interesting concepts such as sampling evolving ego networks and family influence for churn prediction.

Acknowledgments

Author expresses sincere thanks to her supervisor Prof. João Gama, Inesctec, University of Porto. Author acknowledges the support of the European Commission through the

project MAESTRA (Grant Number ICT-750 2013-612944), FCT within project UID/EEA/50014/2013 and also thank WeDo Business for providing the data.

References

[1] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, 1985.

[2] W. Wei, J. Erenrich, and B. Selman, "Towards efficient sampling: Exploiting random walk strategies," in *AAAI*, vol. 4, 2004, pp. 670–676.

[3] M. Papagelis, G. Das, and N. Koudas, "Sampling online social networks," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 3, pp. 662–676, 2013.

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

[5] S. Tabassum and J. Gama, "Sampling massive streaming call graphs," in *31st ACM/SIGAPP Symposium on Applied Computing, Volume: 1*. ACM, 2016, p. In Press.

[6] A. Metwally, D. Agrawal, and A. El Abbadi, "Efficient computation of frequent and top-k elements in data streams," in *Database Theory-ICDT 2005*. Springer, 2005, pp. 398–412.

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

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

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

[10] B. Wellman, "Are personal communities local? a dumptarian reconsideration," *Social networks*, vol. 18, no. 4, pp. 347–354, 1996.

[11] H. H. Ma, S. Gustafson, A. Moitra, and D. Bracewell, "Ego-centric network sampling in viral marketing applications," in *Mining and Analyzing Social Networks*. Springer, 2010, pp. 35–51.

[12] S. Tabassum and J. Gama, "Sampling evolving ego-networks with forgetting factor," in *17th IEEE International Conference on Mobile Data Management*. IEEE, 2016, p. In Press.

[13] D. Ortiz-Arroyo, *Discovering sets of key players in social networks*. Springer, 2010.

[14] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991.