Dynamic Community Analysis in Decentralized Online Social Networks

Barbara Guidi✉, Andrea Michienzi, Giulio Rossetti

University of Pisa
Department of Computer Science
Largo B. Pontecorvo, 56127
Pisa, Italy
Email:{guidi, rossetti}@di.unipi.it
a.michienzi@studenti.unipi.it

Abstract. Community structure is one of the most studied features of Online Social Networks (OSNs). Community detection guarantees several advantages for both centralized and decentralized social networks. Decentralized Online Social Networks (DOSNs) have been proposed to provide more control over private data. One of the main challenge in DOSNs concerns the availability of social data and communities can be exploited to guarantee a more efficient solution about the data availability problem. The detection of communities and the management of their evolution represents a hard process, especially in highly dynamic social networks, such as DOSNs, where the online/offline status of user changes very frequently. In this paper, we focus our attention on a preliminary analysis of dynamic community detection in DOSNs by studying a real Facebook dataset to evaluate how frequent the communities change over time and which events are more frequent. The results prove that the social graph has a high instability and distributed solutions to manage the dynamism are needed.

Keywords: Decentralized Online Social Networks, P2P, dynamic community, data availability

1 Introduction

Static features, such as clustering coefficient or centrality of Online Social Networks (OSNs) have been largely studied. In particular, the community structure is one of the most studied feature of OSNs and it has attracted wide attention. The general notion of community refers to the fact that nodes tend to form clusters which are more densely interconnected through social relationships, relatively to the rest of the network. Communities reflect the behaviour of users and a high percentage of shared contents are generated by communities (or groups) of social users. During the last ten years, the increase of the amount of social data produced by social users, has put users inside several privacy issues. Centralized solutions for OSNs have been considered the main weak point in the problem

of guarantee a certain level of privacy. To overcome this issue, decentralized solutions, known as Decentralized Online Social Networks (DOSNs), have been proposed. The decentralization includes several benefits, in particular in terms of privacy preserving, but it introduces new challenges that have to be faced. In particular, the problem of data availability is one of the most important ones. Current proposals manage the problem of data availability through a user-centric point of view, and no approaches take into account groups (or communities) of users.

Several studies are proposed to manage the community detection in dynamic environments, such as Mobile Social Networks or Opportunistic Networks. However these studies manage scenarios in which mobile devices make contact with each other and they consider a community as a group of connected nodes.

By considering the importance of a community-centric point of view and the high level of dynamism in DOSNs, this work proposed a preliminary study of dynamic communities by using a real Facebook dataset. In detail, we define a set of community change events that are important to manage the data availability problem and we study the dynamic communities in ego networks to evaluate how frequent the communities change over time and which events are more frequent. All our studies show the need of a distributed approach to manage the problem of the high instability of the social graph over time when we consider the online presence of users.

The important contribution of this work is that, even we consider a specific scenario, our contribution could be applied to other distributed systems, by taking into account the specific constraints.

This paper is organized as follow. In Section 2 we describe the related work. In Section 3 we introduce the dynamic community analysis in DOSNs. A preliminary analysis is showed in Section 4. Finally, conclusions and future work are presented in Section 5.

2 Related Work

In this section we describe the two fields involved in our work. First of all, we introduce current DOSN proposals by describing their characteristics. Afterwards, we describe the state of art in the dynamic community detection field.

2.1 DOSN's approaches

DOSNs [7] have been proposed in order to overcome the privacy issues of the centralized OSNs. The decentralization of most of the current proposals is implemented by a P2P network. Diaspora[1], with about 669,000 users, is one of the most successful DOSN proposal currently active and deployed in a decentralized way. PeerSoN [2] is one of the most well-known DOSN after Diaspora. It is implemented as a two-tier system in which the first tier is used for the lookup service,

[1] https://joindiaspora.com/

instead the second tier is used for the communication between peers and the exchange of users' profile. SafeBook [6] uses a social overlay named *Matryoshkas*, which is composed by concentric rings of peers built around each peer. The social overlay guarantees a trusted data storage and an obscure communication through indirection. LifeSocial [10] proposes a solution to the privacy issue by using public-private key pairs to encrypt profile data which are store in a DHT. DiDuSoNet [12] is built on a Dunbar-based social overlay and it is focused on the data availability issue by introducing the concept of Point of Storage (PoS). The number of replicas each profile has is minimized by considering only two replicas. A similar approach is Cachet [17] which replicates profiles on the DHT. Cachet does not minimize the number of replicas and it does not manage the problem of consistency raised to keep all replicas up-to-date.

2.2 Dynamic Community Detection

Dynamic Community Discovery is a relatively novel task in complex network analysis [1, 3], its goal being identify and track trough time clusters of highly connected nodes in a dynamic network. In a preliminary survey [14] two high level categories of online Dynamic Community Discovery algorithms are identified depending on how the community evolution is handled: (i) *Temporal Smoothness* approaches run the community discovery process from scratch on each graph evolution step (e.g. network snapshot); (ii) *Dynamic Updates* approaches incrementally update the communities as time goes by looking both at their previous states and at novel network perturbations. In static community discovery a formal and shared definition of community is still missing: such ill-posedness applies even to the dynamic extension of the problem, thus leading to several detection and quality criteria. Since there are countless ways to define what a dynamic community should look like most of the literature on the subject focus not on reaching consensus on community topology but on the description of approaches able to track elementary communities evolution patterns. Following such rationale, several works converged on the definition and adoption of a stable set of events that can be used to describe dynamic community life-cycles [18, 4, 19]: Birth, Death, Growth, Contraction, Merge, Split.

3 Towards the dynamic community analysis in DOSNs

Several approaches propose to manage the problem of community detection in social networks take into account the evolution of the social graph in term of friendship relationship (or co-authorships [22, 21]), or in term of interactions between users (or call graphs [11]).

Focusing on a single user, its friendship relationships do not change so frequently. Instead, interactions of each nature (calls, emails, posts, tweets, etc...) suffer of a different level of dynamism. However, the study of the interactions graph represents a different evaluation of the social graph, because the interaction graph is an abstraction of the social graph that should be represented as a

weighted and usually directed graph [13]. In a distributed system which wants to provide social services, such as a DOSN, an interest evaluation concerns the study of dynamic community by considering the temporal behaviour of users. As showed in our previous work [20], the static view of an ego network and as a consequence its communities are completely different when we consider the time-varying ego network.

In the follow, we describe more in detail our DOSN's architecture by explaining how our architecture is organized. Moreover, we explain the problem of data availability, which is the main goal treated by our DOSN [12]. Finally, we give our definition of the events occurred during the normal activity of a DOSN which involve the dynamic communities.

3.1 DOSN: our scenario

A current trend of DOSNs is the usage of a social overlay which represents in some way the friendship relationships between users. The network topology resulting is generally known as a Friend to Friend network (F2F) in which users only make direct connections with people they know. Usually in OSNs, the social graph of each user is referred by using a well-known social network model known as *Ego Network*. The *Ego Network* [15] of a user represents a structure built around the ego which contains his direct friends, known as *alters* and may also include information about the direct connections between the alters. Formally, each vertex $u \in V$ can be seen as an *ego* and $EN(u) = (V_u, E_u)$ is the ego network of u where $V_u = \{u\} \cup \{v \in V | (u, v) \in E\}$, $E_u = \{(a, b) \in E | \{a, b\} \subseteq V_u\}$ and E is the set of edges present in the original graph. $N(u) = V_u - \{u\}$ is the set of adjacent nodes of u.
A F2F network can be formally represented by using an *Ego Network* to model the social graph and we assume a one-to-one mapping between the users of the OSN and the nodes of the DOSN [12].

3.2 Data Availability problem

Data availability is a real hard problem for every distributed environment. Replication is the most used technique to manage this challenge.

In our scenario, the problem has a big constraint inserted to maintain a high level of privacy inside the system. The constraint concerns how data should be stored: replica nodes are chosen by exploiting friendship relations.

To manage the problem of data availability, proper techniques must be introduced in order to ensure that data of the ego users will be available on a subset of their alters.

In our previous works [12, 9], we have exploited a friendship-based replication schema. A friendship-based replication schema chooses replica nodes by taking into account the friendship relationships between users. Indeed, consider an ego node e, only its friend nodes can be chosen to be its replica nodes.

This replication schema is applied also in other DOSN proposals, such as My3 [16]. However, the data availability could be guided from both friendship

relationships and a content-based point of view. For sake of clarity, a content based point of view concerns the problem to find group of users which are interest to a same content to minimize the number of replicas. Groups of users can be defined with a *community* and this approach can be named as a *community-based replication technique*. The presence of densely connected groups of nodes can be exploited to increase the level of data availability and to minimize the replicas. A possible approach could be exploit the community structure to store at least one replica of the whole profile or of interest content for the users belonging to the community. As discussed in Section 3, ego networks in DOSNs suffer of a high level of dynamism and for this reason, we are interested in studying how communities evolve during the online activity of the system due to the online/offline of users to understand which community change events could happen and the frequency of them.

3.3 Dynamic Community Analysis in DOSNs

A real interest in studying the dynamic community in distributed environment is to understand how the network changes and in particular, after defining what we intend as community, how the community evolves during the time. In this paper a community is identified with nodes that are densely linked to each other, directly or through other nodes. We represent an ego network e as a set of n snapshots $(EG_1^e, EG_2^e, ..., EG_n^e)$. Each snapshot of an ego network e at time i, identified as EG_i^e, contains a set of communities $C = (C_i^1, C_i^2, ..., C_i^m)$ We are interest to evaluate the evolution of communities in term of the community change events explained in detail in [22]. For sake of readiness, communities events are merge, split, death, and birth. To evaluate the similarity between communities, we use a revised version of the similarity metric proposed in [22]. Consider an ego network e and two snapshot EG_i^e and EG_j^e, the revised similarity metric is introduced by the Eq. (1),

$$sim(C_{i-1}^p, C_i^q) = \frac{|V_{i-1}^p \cap V_i^q|}{max(|V_{i-1}^p|, |V_i^q|)} \tag{1}$$

where C_i^q is the community q included in EG_i^e and C_{i-1}^p is the community p included in EG_{i-1}^e. Instead, V_{i-1}^p is the set of nodes contained in C_{i-1}^p and V_i^q is the set of nodes contained in C_i^q.

Thanks to this similarity metric, each community in a time instant i is compared with each community of the time instant $i-1$.

Moreover, we need to redefine all the possible community change events (merge, split, death, birth) to be applied in a DOSN. We propose our definition of the four events:

- *Birth*: we say that a community C_i^p is born at time i if, given the set of communities $C_{i-1}^* = \{C_{i-1}^1, C_{i-1}^2, \cdots, C_{i-1}^k\}$ at time $i-1$, $\forall C_{i-1}^j \in C_{i-1}^*$, we have that $sim(C_i^p, C_{i-1}^j) = 0$. This means that all the communities discovered at the previous time instant $(i-1)$ do not share any node with C_i^p.

- *Death*: we say that a community C_{i-1}^p is dead at time i if, given the set of communities $C_i^* = \{C_i^1, C_i^2, \ldots, C_i^k\}$ at time i, $\forall C_i^j \in C_i^*$, we have that $sim(C_{i-1}^p, C_i^j) = 0$. This means that all the communities discovered at time i do not share any node with C_{i-1}^p.
- *Merge*: we say that a set of communities $C_{i-1}^* = \{C_{i-1}^1, C_{i-1}^2, \ldots, C_{i-1}^k\}$ merge into a community C_i^p if, for each community $C_{i-1}^j \in C_{i-1}^*$, we have that $sim(C_{i-1}^j, C_i^p) \geqslant k$, where k is the similarity threshold defined in [22]. This means that k% of mutual friends between C_i^p and each community in C_{i-1}^* are included in C_i^p.
- *Split*: we say that a community C_{i-1}^p splits into a set of communities $C_i^* = \{C_i^1, C_i^2, \ldots, C_i^n\}$ if, for each community $C_i^j \in C_i^*$, we have that $sim(C_i^j, C_{i-1}^p) \geqslant k$ where k is the similarity threshold as described in [22]. This means that a community C_i^p is divided in a set of community identified by C_{i-1}^*.

3.4 How community change events affect the data availability

In this study we refer to the events proposed in [22] and we do not consider the event *survive*, usually referred as *growth* and *shrink*, because it is less relevant in term of data availability, due to the fact that this event gives little information about the evolution of the communities in the network.

Considering the problem of data availability in DOSNs and our proposed community-based replication technique explained in Sec. 3.2, the events *birth*, *death*, *split* and *merge* can affect the level of availability and the number of replicas. *Birth* events are critical and they are one of the main issue that has to be faced. Indeed, a newly formed community may have little to no information about the most fresh contents created by the ego and nodes inside such communities must find a way to retrieve it. *Death* events are reported mainly to give us more information about node churn in such dynamic context, but are no concern in a replication technique because offline nodes do not need any content. *Merge* and *split* events are important because, in the former case, nodes that belong to different communities converge in the same community, so they should merge the available information and probably a few replicas of data can be dropped. In the latter case, splitted communities suggest that communities may become more distant over time, so the content may need to be redistributed and replicated over the newly formed communities.

4 A case study: Facebook

To evaluate the dynamics in OSNs, we study Facebook through our dataset retrieved by a Facebook application, called SocialCircles![2].

As described in [8], SocialCircles! was able to retrieve the following sets of information from registered users:

[2] `https://www.facebook.com/SocialCircles-244719909045196/`

Friendship We obtained friends of registered users and the friendship relations existing between them.

Online presence We monitored the chat status of users in Facebook. The presence status is identified with 0 if user is offline, 1 if user is in active state and 2 if user is idle.

We were able to obtain two different datasets, the first one introduced in [8] and a second one composed by 240 users monitored for 32 consecutive days. In detail, we sampled all the registered users and their friends every 5 minutes, for 32 days (from 9 March to 10 April 2015). Using this methodology we were able to access the temporal status of about 240 registered users and of their friends (for a total of 78.129 users).

A discrete time model is used to represent the online/offline status of the users during the simulation. In particular, each day of the monitored period consists of a finite number of time slots (i.e., 288 time slots each of 5 minutes), for a total number of 9251 time slots in the whole monitored period.
Figure 1 shows the number of online users for each time slot. The figure shows that there is a clear periodic pattern, probably reflecting the day/night cycle. By analyzing the amount of users online for each time slot, we can see that we have at most around 18000 online users, roughly 23% of the total amount, and at least 3000, 3.8% of the total amount of users.

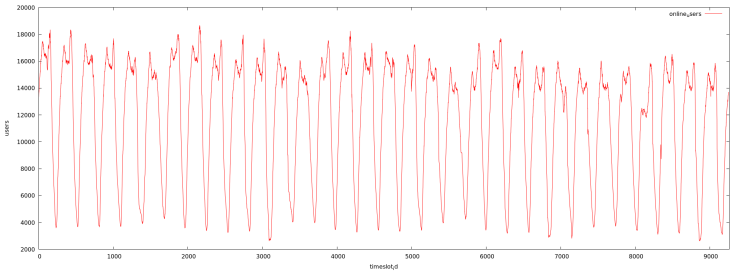

Fig. 1. Online users count during the observed period

4.1 Dynamic community evaluation

For the community discovery algorithm, we choose DEMON [5] among the many that are present in literature. The main reason is that we define a community as a group of clustered nodes that is the community structure found by the label propagation implemented in DEMON. Moreover, DEMON is computationally not expensive. Indeed, it is theoretically linear in time. For the community similarity computation, we are interested only in computing the similarity value for each community at time i, with all the communities at time $i-1$. This saves

a lot of computation of similarity between communities that do not belong to adjacent time slots. We computed the community events as described in section 3 considering two different sets of communities:

- All: in this case we considered all the communities of all ego networks during the observed period of time of 32 days;
- Selected: consider only the communities in the time slots where the related ego was offline (inter-arrival session slots).

With this differentiation we aim to capture a generic, global view of the dynamism of the network in the first case, and a more specific, critical view in the second case. It is very important to understand how the network evolves in time, also when it is not strictly needed for the data availability problem because we need to handle churn.
As a preliminary analysis, we computed some statistical measures on the number and size of the dynamic communities to compare them with the static communities. Table 1 reports the measures for all communities, while table 2 reports the same measures for the static communities. By analyzing the dynamic results, we can say that the network is, as expected, very shattered and not even close to the static view. When considering the number of communities, the high value of standard deviation with respect to the average, suggests that in some particular time slots some ego networks have no community at all. In the static case we have a lower maximum value and an higher average with respect to the dynamic case, which suggests that it is very unlikely to have a dynamic ego network that is similar to the static one. Also the size statistics confirms this fact: static communities tend to be larger than the dynamic ones. We can explain the difference in the two results by recalling the fact that we have at most less than a forth of the users online, as reported in figure 1.

	Min	Max	Mean	Std. Deviation
Number	0	104	2.2814344395195665	3.75809047211548
Size	4	452	17.643563738762122	22.10944505125662

Table 1. Statistical measures on number and size of all dynamic communities

	Min	Max	Mean	Std. Deviation
Number	1	26	9.49583333333333	4.401405173746986
Size	4	1894	99.38788942518802	141.28948531026552

Table 2. Statistical measures on number and size of static communities

To better understand how the events are arranged during the observed time, we decided to make some plots. Figure 2 shows the arrangements of the events

Event	Min	Max	Mean	Std. Deviation
Split	0	173	61.90498324505457	38.221971154669156
Merge	0	170	61.961301480920845	38.21300775550354
Death	0	117	39.51551183655814	15.25082726706613
Birth	0	98.0	39.51940330775052	16.40523076472613

Table 3. Statistical measures on community events of all dynamic communities

Event	Min	Max	Average	Std. Deviation
Split	0	122	44.72067884553051	26.444005572482837
Merge	0	124	44.78207761323137	26.432679749759142
Death	0	80	26.160415090260557	12.350502534938407
Birth	0	58	26.152307858609966	12.836074158727344

Table 4. Statistical measures on community events of selected dynamic communities

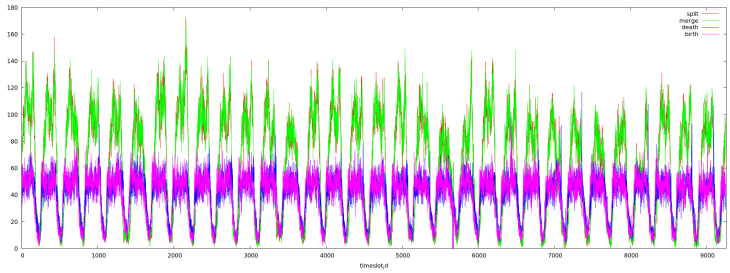

Fig. 2. Community events for each time slot of all dynamic communities

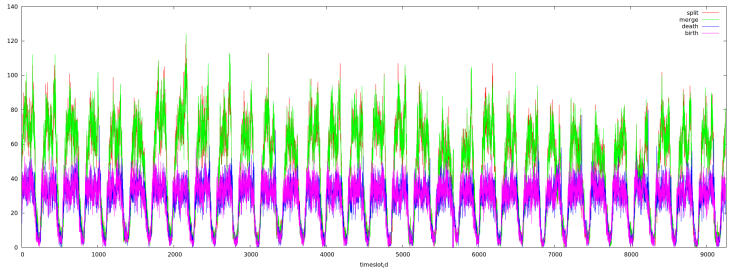

Fig. 3. Community events for each time slot of selected dynamic communities

when considering all communities of all time slots while Figure 3 shows the events for the selected communities. Both the figures show that there is a temporal pattern in the results, suggesting that the behaviour follows a daily cycle, confirming the results in figure 1. Moreover, on the peaks, the number of merge/split events are roughly double the number of death/birth events, while in the nadirs the

number of merge/split events are slightly less than the number of death/birth events. By taking a closer look at the arrangements of the events, we may also observe that peaks and nadirs of merge and split events are slightly moved on the right with respect to the ones of birth and death events, which means that, before observing a variation on the number of split and merge events, we should see a variation in the number of birth and death events. It is also worth noticing that, as expected, at each drop of the events corresponds a peak in deaths, which probably means that we are approaching the night time slots. Dually, at each increase of events, we usually see a peak of birth events, which should correspond to the time slots where people wake up. Another important result is that the two graphs look similar which is sign that the network behaves in the same way both when the ego is online or offline. This is of interest in the sense that all the analysis can be done regardless that an ego is online or not.
Finally, since the events follow a daily cycle, we are interested to see how this events are related to the presence of users on the network. From a comparison between figures 2 and 3 with figure 1 we can see that the more users are online, the more events are observed in the network. This means that, in a community-based replication technique, choosing the replicas when there are less users on the network is somewhat easier because the network is more stable in terms of communities, while, on the other hand, when there are a lot of users online, we need to handle more community events, especially split and merge events.

5 Conclusion and Future works

In this paper we propose a preliminary analysis of dynamic community due to the online/offline status of users in DOSNs. In detail, we focus our attention of the data availability problem that is one of the most important problems in DOSNs and we propose a set of community change events which are important in our scenario. We analyze both how and the frequency of these events by exploiting a real Facebook dataset gathered by our Facebook application (SocialCircles). Results show that DOSNs are affected by a high dynamism and a community-based replication schema needs to be supported by a distributed algorithm able to manage the dynamism of communities. By analyzing the dynamic results, we show that the network is very shattered and not even close to the static view. Moreover, the community change events introduced in this paper have a temporal pattern that is similar to the temporal user behaviour and, in a community-based replication technique, when there are less users the network is more stable in terms of communities, while, when there are a lot of users online, we need to handle more community events. We plan a deep analysis of the instability of the social graph due to the online/offline status of users. In particular, we plan to develop a distributed algorithm to detect the dynamic community, which can be used to address the problem of data availability.

References

1. Aynaud, T., Fleury, E., Guillaume, J.L., Wang, Q.: Communities in evolving networks: definitions, detection, and analysis techniques. In: Dynamics On and Of Complex Networks, Volume 2, pp. 159–200. Springer (2013)
2. Buchegger, S., Schioberg, D., Vu, L., Datta, A.: Implementing a P2P Social Network - Early Experiences and Insights from PeerSoN. In: Second ACM Workshop on Social Network Systems (Co-located with EuroSys 2009)
3. Cazabet, R., Amblard, F.: Dynamic community detection. In: Encyclopedia of Social Network Analysis and Mining, pp. 404–414. Springer (2014)
4. Cazabet, R., Amblard, F., Hanachi, C.: Detection of overlapping communities in dynamical social networks. In: Social Computing (SocialCom), 2010 IEEE Second International Conference on. pp. 309–314. IEEE (2010)
5. Coscia, M., Rossetti, G., Giannotti, F., Pedreschi, D.: DEMON: A local-first discovery method for overlapping communities. In: Proceedings of the 18th ACM SIGKDD. KDD '12 (2012)
6. Cutillo, L.A., Molva, R., Strufe, T.: Safebook: A privacy-preserving online social network leveraging on real-life trust. Comm. Mag. 47(12) (Dec 2009)
7. Datta, A., Buchegger, S., Vu, L.H., Strufe, T., Rzadca, K.: Decentralized online social networks. In: Handbook of Social Network Technologies and Applications, pp. 349–378. Springer (2010)
8. De Salve, A., Dondio, M., Guidi, B., Ricci, L.: The impact of user's availability on on-line ego networks: a facebook analysis. Computer Communications 73, 211–218 (2016)
9. De Salve, A., Guidi, B., Mori, P., Ricci, L.: Distributed coverage of ego networks in f2f online social networks. In: Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences. pp. 423–431 (2016)
10. Graffi, K., Gross, C., Mukherjee, P., Kovacevic, A., Steinmetz, R.: Lifesocial.kom: A p2p-based platform for secure online social networks. In: Peer-to-Peer Computing. pp. 1–2. IEEE (2010)
11. Greene, D., Doyle, D., Cunningham, P.: Tracking the evolution of communities in dynamic social networks. In: Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining. pp. 176–183. ASONAM '10 (2010)
12. Guidi, B., Amft, T., De Salve, A., Graffi, K., Ricci, L.: Didusonet: A p2p architecture for distributed dunbar-based social networks. Peer-to-Peer Networking and Applications pp. 1–18 (2015)
13. Guidi, B., Conti, M., Ricci, L.: P2p architectures for distributed online social networks. In: High Performance Computing and Simulation (HPCS), 2013 International Conference on. pp. 678–681. IEEE (2013)
14. Hartmann, T., Kappes, A., Wagner, D.: Clustering evolving networks. In: Algorithm Engineering, pp. 280–329. Springer (2016)
15. Marsden, P.: Egocentric and sociocentric measures of network centrality. Social Networks 24(4), 407–422 (2002)
16. Narendula, R., Papaioannou, T.G., Aberer, K.: My3: A highly-available p2p-based online social network. In: Peer-to-Peer Computing (P2P), 2011 IEEE International Conference on. pp. 166–167. IEEE (2011)

17. Nilizadeh, S. and Jahid, S. and Mittal, P. and Borisov, N. and Kapadia, A.: Cachet: a decentralized architecture for privacy preserving social networking with caching. In: Proceedings of the 8th international conference on Emerging networking experiments and technologies. pp. 337–348. CoNEXT '12, ACM (2012)
18. Palla, G., Barabási, A.L., Vicsek, T.: Quantifying social group evolution. Nature 446(7136), 664–667 (2007)
19. Rossetti, G., Pappalardo, L., Pedreschi, D., Giannotti, F.: Tiles: an online algorithm for community discovery in dynamic social networks. Machine Learning pp. 1–29
20. Salve, A.D., Guidi, B., Ricci, L.: Evaluation of structural and temporal properties of ego networks for data availability in dosns. Mobile Networks and Applications pp. 1–12 (2017)
21. Takaffoli, M., Rabbany, R., Zaïane, O.R.: Community evolution prediction in dynamic social networks. In: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on. pp. 9–16 (2014)
22. Takaffoli, M., Sangi, F., Fagnan, J., Zäıane, O.R.: Community evolution mining in dynamic social networks. Procedia-Social and Behavioral Sciences 22, 49–58 (2011)