

Characterizing Temporal Anomalies in Evolving Networks

N.N.R. Ranga Suri¹, M. Narasimha Murty², and G. Athithan^{1,3}

¹ Centre for AI and Robotics (CAIR), Bangalore, India
{rangasuri,athithan.g}@gmail.com

² Dept of CSA, Indian Institute of Science (IISc), Bangalore, India
mnm@csa.iisc.ernet.in

³ Presently working at Scientific Analysis Group (SAG), Delhi, India

Abstract. Many real world networks evolve over time indicating their dynamic nature to cope up with the changing real life scenarios. Detection of various categories of anomalies, also known as outliers, in graph representation of such network data is essential for discovering different irregular connectivity patterns with potential adverse effects such as intrusions into a computer network. Characterizing the behavior of such anomalies (outliers) during the evolution of the network over time is critical for their mitigation. In this context, a novel method for an effective characterization of network anomalies is proposed here by defining various categories of graph outliers depending on their temporal behavior noticeable across multiple instances of a network during its evolution. The efficacy of the proposed method is demonstrated through an experimental evaluation using various benchmark graph data sets.

Keywords: Mining network data, Evolving networks, Anomaly detection, Graph anomalies, Temporal outliers.

1 Introduction

Graph based representation of network structure gave rise to the rapid proliferation of research work on social network analysis. Similarly, graph based representation is preferred for modeling the dynamism inherent to many real life applications. Thus, graph mining plays an important role in any meaningful analysis of the network data. An emerging research problem related to graph mining is to discover anomalies [4,1], also known as outliers, in graph representation of the network data. Due to the networked nature of the data pertaining to many real life applications such as computer communication networks, social networks of friends, the citation networks of documents, hyper-linked networks of web pages, etc [7], anomaly/outlier detection in graphs [1,3,12] turns out to be an important pattern discovery activity. It provides the basis for realizing certain application specific tasks such as fraud detection in on-line financial transactions, intrusion detection in computer networks, etc [4].

In case of IP networks comprising of several individual entities such as routers and switches, the behavior of the individual entities determines the ensemble behavior of the network [15]. In this context, network anomalies typically refer to circumstances when network operations deviate from normal behavior, resulting in many unusual traffic patterns noticeable in such network data. These traffic patterns arise due to different types of network abuse such as denial of service attacks, port scans, and worms; as well as from legitimate activity such as transient changes in customer demand, flash crowds, or occasional high-volume flows [10]. These traffic anomalies happen to be the major security concern to the network administrators and detecting such anomalies in evolving networks [14,11] is essential for ensuring a healthy operational status of these networks.

Connected with any dynamic network, the underlying graph representation undergoes time dependent changes in its connectivity structure. This will have a consequence on the local connectivity and thus influences the temporal behavior of various graph entities such as nodes, edges and sub-graphs. As a result, a graph entity that qualifies to be an outlier (irregular connectivity pattern) at the current instance may no more continue to be the same at a later point of time due to the changes in connectivity structure of the graph. Additionally, some new irregularities may arise in a different portion of the graph hinting at the presence of some new outliers. If the graph connectivity structure is altered further, some intermediate outliers may cease to exist and further new types of outliers may start appearing in the graph. Thus, it is important to track such evolving scenarios for a subjective analysis of the graph representation towards understanding and characterizing the outlier dynamics. Accordingly, study of evolving graphs/networks has been an active research problem as evident from some recent efforts in this direction [2,14].

Motivated by the above discussion, a novel framework for characterizing temporal anomalies in evolving networks is envisaged here. This involves identifying various graph outliers and characterizing their dynamics across multiple instances of the graph representation of a dynamic network at various points of time during its evolution. Accordingly, a novel algorithm is proposed in this paper for detecting various temporal outliers in a dynamic graph and for producing a semantic categorization of the detected outliers by defining different categories of temporal outliers.

Highlights of the work reported in this paper are as follows:

- The problem of detecting anomalies in dynamic network data is an active research area [11,14].
- Characterizing temporal anomalies (outliers) is essential in *many application contexts* such as ensuring the security of information networks.
- A thorough exploration of the *temporal behavior of graph outliers* by defining various categories of temporal outliers based on their occurrence patterns.
- A novel algorithm for effective detection of various temporal outliers and for analyzing their dynamics.

The rest of the paper is organized in four sections. Section 2 gives a brief discussion on the relevant literature. Subsequent section describes the proposed

method for the study of outlier dynamics in evolving networks/graphs. An experimental evaluation of the proposed method is furnished in Section 4 bringing out its empirical findings. Finally, Section 5 concludes this paper with a discussion and a few directions for future work on this problem.

2 Related Literature

A method for spotting significant anomalous regions in dynamic networks has been proposed [11] recently. According to this method, the anomalous score of a weighted edge in an undirected connected graph is determined based on a statistical measure. The aim is to find contiguous regions having adjacent anomalous edges forming large regions of higher score, named as Significant Anomalous Regions (SAR) in a dynamic network. The anomaly score of such a region is quantified by aggregating the scores of the participating edges. Given an edge-weighted graph $G = (V, E, W)$, the anomaly score of a temporal network region $R = (G', [i, j])$, where $G' = (V', E')$ is a connected sub-graph of G , in the time interval $[i, j]$ is given by

$$score_G(R) = \sum_{e \in E'} \sum_{t=i}^j w^t(e) \quad (1)$$

where $w^t(e)$ is anomaly score associated with an edge e at time point t .

This method is basically an edge-centric approach where the anomalous scores of edges determine the anomalous regions in a graph. Also, computing aggregate score over time may not highlight the temporal characteristics of the primitive outliers such as node/edge outliers in a graph.

In another recent effort, a multi-graph clustering method [13] was proposed exploiting the interactions among different dimensions of a given network. Though the multi-graph scenario is conceptually different from multiple instances of an evolving graph, establishing a relationship between these two problem settings may enable leveraging the methods developed for one setting in effectively solving the other. Similarly, other recent efforts [2,14] on dynamic graphs are aimed at modeling some specific time varying properties of such graphs in a meaningful manner. A substructure-based network behavior anomaly detection approach, named as Weighted Frequent Sub-graphs method, was proposed [6] to detect the anomalies present in large-scale IP networks. According to this method, patterns of abnormal traffic behavior are identified using multivariate time series motif association rules mining procedure.

A method to capture the evolving nature of abnormal moving trajectories was proposed in [5] considering both current and past outlieriness of each trajectory. Such a method is intended to explore the behaviors of moving objects as well as the patterns of transportation networks. It identifies the top- k evolving outlying trajectories in a real time fashion by defining an appropriate outlier score function. Unlike the method described in [11] computing a simple summation of the anomalous scores over time, this particular method utilizes a decay function in

determining the evolving outlier score so as to mitigate the influence of the past trajectories.

2.1 Outlier Detection in Graphs

As mentioned in the previous section, the essential graph mining task here is to detect various anomalies/outliers present in the graph representation of a dynamic network at a specific point of time, i.e. detecting outliers in a single graph instance. Among the methods available in the literature addressing this requirement, the method proposed in [12] is an early one using the minimum description length (MDL) principle. The main idea of this method is that sub-graphs containing many common sub-structures are generally less anomalous than sub-graphs with few common sub-structures. Thus, it is suitable mainly for applications involving many common sub-structures such as the graphs describing the atomic structure of various chemical compounds. Similarly, the method proposed in [3] is meant for anomalous link (edge) discovery defining a novel edge significance measure.

A recent work on anomaly detection in graph data [1], known as ‘OddBall’, makes use of the notation of *egonet* of a node defined as the induced sub-graph of its 1-step neighbors. According to this method, the number of nodes N_i and the number of edges E_i of the egonet G_i of node i in a graph $G = (V, E)$ follow a power law relationship, named the Egonet Density Power Law (EDPL) defined as

$$E_i \propto N_i^\alpha, \quad 1 \leq \alpha \leq 2. \quad (2)$$

Consequently, two types of anomalous egonets, named as *near cliques* and *near stars*, are determined by measuring the amount of deviation from the power law relationship. Thus, the outlierness score of an egonet sub-graph G_i is computed as the distance to the least squares fitting line as defined in [1].

$$out_score(G_i) = \frac{\max(E_i, CN_i^\alpha)}{\min(E_i, CN_i^\alpha)} \log(|E_i - CN_i^\alpha| + 1) \quad (3)$$

Finally, the anomalous sub-graphs are indicated in a scatter plot, referred to as EDLP plot, showing their deviation from the fitting line.

The spectral approach for detecting subtle anomalies in graphs [16] is a more recent method, which is of relevance to our work. This approach detects the anomalous sub-graphs in the input graph by exploring the minor eigenvectors of the adjacency matrix of the graph. The underlying assumption is that there exist some minor eigenvectors with extreme values on some entries corresponding to the anomalies in the graph.

3 Proposed Method

Given the graph representation of network data, there may exist various types of graph outliers in the form of node outliers, edge outliers and sub-graph outliers as

discussed in the previous section. The process of graph evolution over time results in some interesting observations in terms of these graph outliers. A framework for capturing this phenomenon is proposed here as shown in Figure 1. The rationale behind this setting is to explore various types of outliers present in a graph at various points of time independently and then to characterize them based on their appearance at different points of time during the network evolution.

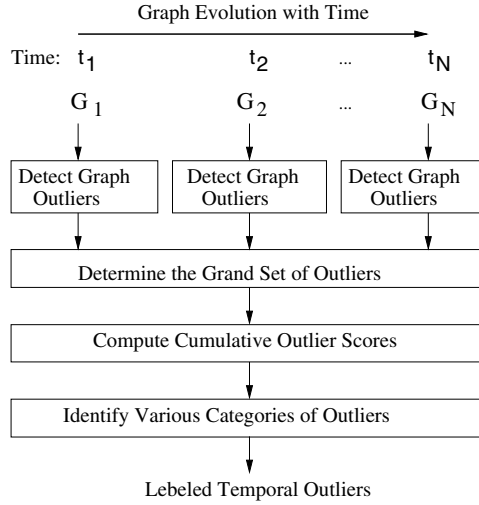


Fig. 1. Proposed framework for detecting temporal outliers

3.1 Preliminaries

The mathematical notation along with some definitions required for developing a novel algorithm for temporal outlier detection is introduced here. We use the two terms ‘outliers’ and ‘anomalies’ in an interchanging manner as they both indicate the data objects deviating from the normal data. Likewise, the two terms ‘network’ and ‘graph’ are used to refer to the same data object.

Let $G = \{G_1, G_2, \dots, G_N\}$ be a dynamic graph data set comprising of N instances of an evolving graph corresponding to the analysis time period $[t_1, t_N]$ consisting of N discrete points of time. Similarly, let $\Psi_i = \{\psi_{i,1}, \psi_{i,2}, \dots, \psi_{i,k}\}$ denote the top- k outliers present in the graph instance G_i at time point t_i and $S_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,k}\}$ be their outlier scores respectively.

Definition 1 (Temporal outlier). *A graph-based outlier that is present in one or more instances of an evolving network/graph.*

Definition 2 (Cumulative outlier score). *The cumulative outlier score of a temporal outlier is computed using its outlier scores at different points of time during the analysis.*

The exact computation of the cumulative outlier score depends upon the specific algorithmic logic considered based on the application context. The set of temporal outliers detected by the proposed algorithm is indicated by Ω and $\Phi = \{\phi_1, \phi_2, \dots\}$ represents the cumulative outlier scores of the detected temporal outliers.

In our characterization of the dynamics of the temporal outliers, we resort to a categorization of them based on their time varying characteristics. To this end, every category of temporal outlier that is noticed during the graph evolution is assigned an explicit label to mark it distinctly depending on its behavior. In order to capture various behavioral patterns of the temporal outliers and to convey the underlying semantics associated with each such pattern, we define the following five categories of temporal outliers noticeable in dynamic graphs.

Definition 3 (Halting outlier). *A temporal outlier O_H that may cease to exist over time during the network evolution.*

$$\exists t_j \in [t_3, t_N] \text{ s.t. } (O_H \in \Psi_p, \forall t_p \in [t_1, t_j)) \wedge (O_H \notin \Psi_j).$$

Definition 4 (Emerging outlier). *A temporal outlier O_E that starts appearing at some point of time.*

$$\exists t_i, t_j \in [t_2, t_N] \text{ s.t. } (O_E \notin \Psi_p, \forall t_p \in [t_1, t_i]) \wedge (O_E \in \Psi_q, \forall t_q \in [t_{i+1}, t_j]) \wedge (t_j > t_{i+1}).$$

Definition 5 (Alternating outlier). *A temporal outlier O_A that is noticed at different points of time intermittently.*

$$\exists t_i, t_j, t_k \in [t_1, t_N] \text{ s.t. } (O_A \in \Psi_i) \wedge (O_A \notin \Psi_j) \wedge (O_A \in \Psi_k) \wedge (O_A \in \Psi_p, \forall t_p \in (t_i, t_j)) \wedge (O_A \notin \Psi_q, \forall t_q \in (t_j, t_k)) \wedge (t_k > t_j > t_i).$$

Definition 6 (Repeating outlier). *A temporal outlier O_R that keeps appearing at all points of time during the network evolution.*

$$O_R \in \Psi_i, \forall t_i \in [t_1, t_N].$$

Definition 7 (Transient outlier). *A temporal outlier O_T that exists only at one point of time.*

$$\exists t_i \in [t_1, t_N] \text{ s.t. } (O_T \in \Psi_i) \wedge (O_T \notin \Psi_j, \forall t_j \in [t_1, t_N]) \wedge (t_j \neq t_i).$$

It is important to note that a halting outlier is different from a transient outlier, where the former one disappears after being present for more than one time point, while the later one exists precisely at one point of time. Also note that the set of top- k outliers Ψ_i present in a graph instance G_i may consist of one or more categories of temporal outliers defined above.

In view of the five categories of temporal outliers defined above, the proposed framework is named as *NetHEART* indicating that the detection of various network anomalies, by way of discovering Halting, Emerging, Alternating, Repeating and Transient (HEART) categories of outliers in dynamic networks, is essential for an effective management of these networks such as ensuring the security of the IP networks.

Algorithm 1. Detecting temporal outliers in evolving graphs**Input:** A dynamic graph G with its N instances at different points of time.**Output:** A ranked list Ω of temporal outliers and their scores Φ .1: For each graph instance G_i perform the following steps.

- (a) Construct the corresponding edge list L_i .
- (b) Apply the graph outlier detection procedure described in [1] on L_i .
- (c) Determine the set of top- k outliers Ψ_i along with their scores S_i .

2: Determine the grand set of outliers across all graph instances as

$$\Omega = \Psi_1 \cup \Psi_2 \cup \dots \cup \Psi_N \quad (4)$$

3: For each temporal outlier $O_j \in \Omega$, determine its cumulative outlier score as

$$\phi_j = \frac{1}{N} \sum_{i=1}^N \xi_{i,j} \quad (5)$$

where

$$\xi_{i,j} = \begin{cases} s_{i,p} & \text{if } \exists p \text{ s.t. } O_j = \psi_{i,p} \\ 0 & \text{otherwise.} \end{cases}$$

4: Determine the set of repeating outliers that exist across all graph instances as

$$\Omega^* = \Psi_1 \cap \Psi_2 \cap \dots \cap \Psi_N \quad (6)$$

5: Similarly, determine all other categories of outliers based on the Ψ_i 's.6: For each outlier $O_j \in \Omega$, label it as per the category it belongs to.7: Arrange the set Ω of temporal outliers in descending order of their scores.**3.2 Proposed Algorithm for Temporal Outlier Detection**

According to the framework shown in Fig. 1, the initial task is to detect outliers present in a single graph instance. Though this task can be accomplished using any one of the established methods like [1,16], we have considered the method described in [1]. The rest of the framework deals with collating the instance specific outliers and characterizing their dynamics across multiple instances of a dynamic graph. Accordingly, a novel method for detecting various categories of temporal outliers in evolving graphs is proposed here as per the steps presented in Algorithm 1.

As per Equation 4, the grand set of outliers (Ω) consists of at most $N * k$ elements and at least k elements. Similarly, Equation 5 determines the cumulative outlier score of a temporal outlier by computing the average of its scores corresponding to various graph instances. Unlike the other recent methods [11,5], the detection of a temporal outlier is not affected by its cumulative outlier score. However, this score indicates the significance of a temporal outlier over all the graph instances.

The execution time of the proposed algorithm is mainly contributed by the time required to complete the first step. For a small k value with a reasonably small analysis time period ($N \leq 10$), the rest of the steps can be completed in a constant amount of time. Therefore, the computational complexity of the proposed algorithm is determined by that of the specific method employed for performing the first step, such as the method described in [1].

4 Experimental Evaluation

To demonstrate the efficacy of the proposed algorithm, an experimental evaluation has been carried out considering two real life dynamic graph data sets. For ease of illustration and simplicity in describing the results, only three graph instances ($N = 3$) have been considered corresponding to each one of these graph data sets as described below.

4.1 Data Sets

The DBLP Computer Science Bibliography from the University of Trier contains millions of bibliographic records bundled in a huge XML file [9]. Various sub-sets of these records concerned with AAAI publications have been considered here, denoted as DBLP-AAAI data set, as per the details furnished in Table 1.

Table 1. Details of the dynamic graph data sets

Data Set Name	Instance	# Nodes	# Edges	Graph Instance Details
DBLP-AAAI	G_1	5,581	25,276	Publications upto 2008
	G_2	6,219	28,622	Publications upto 2010
	G_3	7,551	35,628	Publications upto 2012
AS-2000	G_1	2,107	8,979	as19991231.txt
	G_2	3,570	14,783	as20000101.txt
	G_3	6,474	26,467	as20000102.txt

The graph of routers comprising the Internet can be organized into sub-graphs called Autonomous Systems (AS). Each AS exchanges traffic flows with some neighbors (peers). It is possible to construct a communication network of who-talks-to-whom from the BGP (Border Gateway Protocol) logs. The experiments in this paper have been carried out on three most recent graph files taken from the Autonomous Systems collection [8] as per the details furnished in Table 1, denoted as AS-2000 data set hereafter.

4.2 Outlier Detection and Characterization

As per the first step of the proposed algorithm, various node-based outliers present in each graph instance were detected using ‘oddball’ method [1]. The

resulting top- k ($=10$) outliers on the DBLP-AAAI graph instances are shown in Fig. 2 indicated using node IDs along with their outlier scores. Subsequently, the grand set consisting of 13 temporal outliers ($|\Omega| = 13$) was determined (as per Equation 4) and the cumulative outlier score of each temporal outlier has also been computed accordingly (as per Equation 5). Then, various categories of temporal outliers (as defined in Section 3.1) present in this graph data set were identified as listed in Table 2, and the same have been depicted in Fig. 2 using a different line style for representing each category. Finally, a ranked list of labeled temporal outliers is produced as the output.

Similarly, the AS-2000 graph data set was also subjected to the computational steps listed in Algorithm 1, and various categories of temporal outliers thus detected have been furnished in Table 2 for completeness.

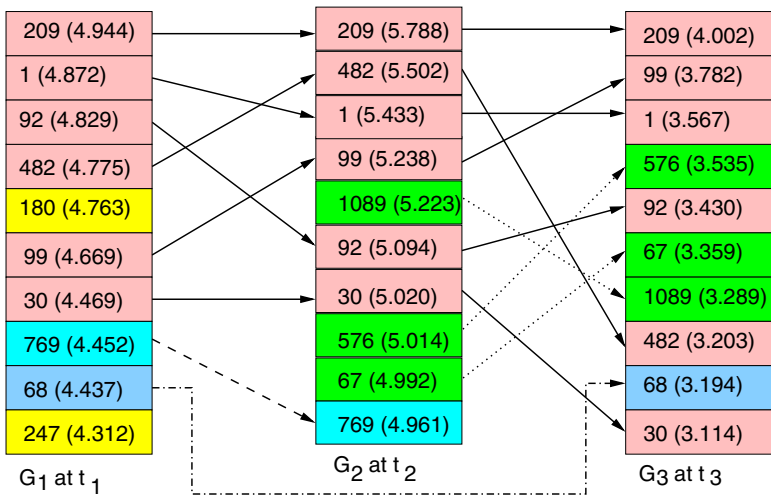


Fig. 2. Time varying characteristics of graph outliers in DBLP-AAAI data set

Table 2. Various categories of temporal outliers identified in dynamic graph data sets

Graph File	Repeating Outliers	Transient Outliers	Halting Outliers	Emerging Outliers	Alternating Outliers
DBLP-AAAI	209, 1, 92, 482, 99, 30	G_1 : 180, 247 G_2 : — G_3 : —	769	1089, 576, 67	68
AS-2000	4006, 4544, 1746, 3563	G_1 : 2551, 2041, 145, 6467 G_2 : 6261, 701 G_3 : 5779, 5000, 11329, 2828	721, 3847	6113, 668	—

It is important to note that the temporal behavior observed in this experimentation may vary with the number of top ranked outliers (k) considered corresponding to each graph instance and the number of instances (N) considered in each graph data set, as the outlier ranking is basically a relative measure.

4.3 Exploring the Outlier Dynamics

As brought out in Section 2.1, an EDPL plot [1] indicates two types of node-based anomalies, namely the near-star anomalies appearing below the fitting line and the near-clique anomalies above the fitting line. Accordingly, the EDPL plots obtained corresponding to different instances of the DBLP-AAAI data set are shown in Fig. 3. A keen observation at these plots indicates that as more and more new edges get added during the graph evolution from G_1 to G_3 , one can notice the gradual appearance of more near-clique type of anomalies (triangle points marked in the upper portion of the plot) as shown in Fig. 3(b).

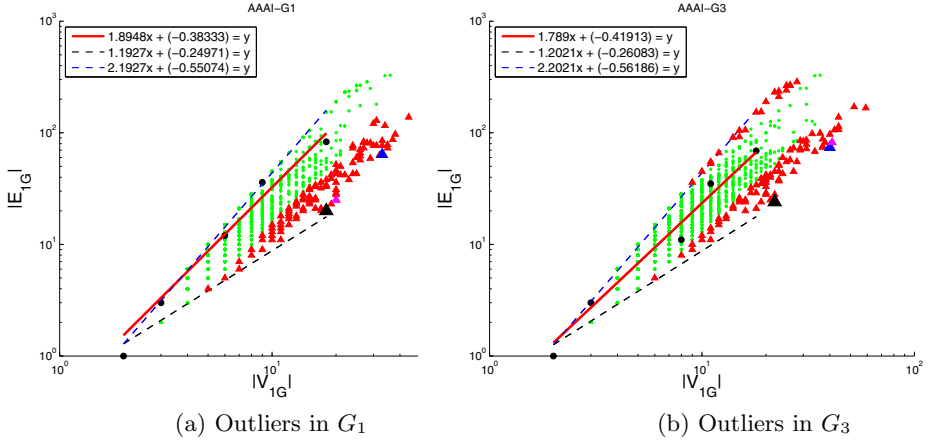
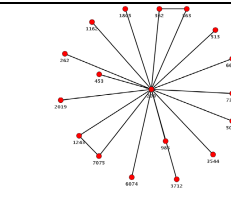
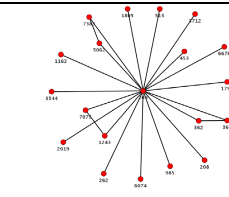
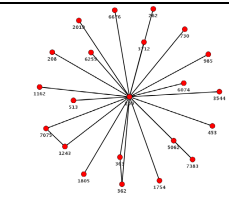
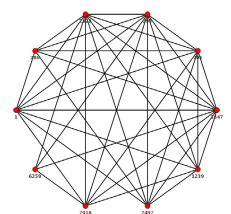
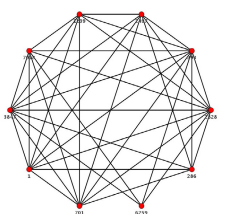
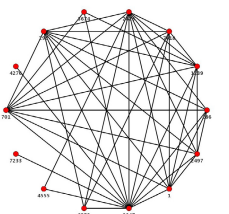


Fig. 3. Top 100 outliers detected in DBLP-AAAI data set

Temporal outliers present in an evolving graph are expected to highlight the inherent semantics of the graph connectivity patterns causing these outliers. To illustrate these semantic aspects, a few of the detected node-based outliers, in the form of egonet sub-graphs, are depicted in Table 3. The number of nodes and the number of edges present in each such sub-graph are also furnished along side in the same table. It is important to note that with the evolution of the graph over time, the underlying egonets have also undergone corresponding change in their connectivity structure. Though there is not any significant change expected in the node-based anomalous sub-graphs individually, their relative position in the top- k outlier rank list may change across various graph instances (refer to Fig. 2) due to the modified connectivity structure of the graph during evolution. For

Table 3. Sub-graphs of various node outliers in different graph instances

Graph File	Egonet in G_1 at t_1	Egonet in G_2 at t_2	Egonet in G_3 at t_3
DBLP-AAAI Node ID: 209	 nodes=18, edges=20	 nodes=20, edges=22	 nodes=22, edges=24
AS-2000 Node ID: 3847	 nodes=10, edges=37	 nodes=10, edges=37	 nodes=14, edges=44

example, the egonet of the node with ID 209 in DBLP-AAAI data set continues to be among the top-10 outliers through out the analysis period. On the other hand, the egonet of the node with ID 3847 in AS-2000 data set ceases to exist among the top-10 outliers at time point t_3 .

5 Conclusion and Future Work

The problem of characterizing temporal anomalies (outliers) in evolving network (graph) data has been addressed here by proposing a novel framework for exploring their time varying behavior. The proposed method has its merit in defining various categories of temporal outliers taking into account their dynamics. An experimental evaluation of the proposed algorithm for temporal outlier detection has been carried out using two benchmark dynamic graph data sets. The experimental observations confirm the presence of various categories of temporal outliers as defined in this paper, demonstrating the effectiveness of the proposed method.

Employing alternative methods, such as the one described in [16], for discovering the anomalies present in the individual instances of an evolving graph and exploring their temporal behavior. Enhancement to the proposed categorization of temporal outliers by involving some graph in-variants may be other interesting direction to consider.

Acknowledgments. The authors would like to thank Director, CAIR for supporting this work.

References

1. Akoglu, L., McGlohon, M., Faloutsos, C.: **Oddball**: Spotting anomalies in weighted graphs. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6119, pp. 410–421. Springer, Heidelberg (2010)
2. Anagnostopoulos, A., Kumar, R., Mahdian, M., Upfal, E., Vandin, F.: Algorithms on evolving graphs. In: ACM ITCS, Cambridge, Massachusetts, USA, pp. 149–160 (2012)
3. Chakrabarti, D.: AutoPart: Parameter-free graph partitioning and outlier detection. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 112–124. Springer, Heidelberg (2004)
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. *ACM Computing Surveys* 41(3) (2009)
5. Ge, Y., Xiong, H., Zhou, Z.H., Ozdemir, H., Yu, J., Lee, K.C.: TOP-EYE: Top-k evolving trajectory outlier detection. In: ACM CIKM, Toronto, Canada, pp. 1733–1736 (2010)
6. He, W., Hu, G., Zhou, Y.: Large-scale ip network behavior anomaly detection and identification using substructure-based approach and multivariate time series mining. *Telecommunication Systems* 50(1), 1–13 (2012)
7. Kim, M., Leskovec, J.: Latent multi-group membership graph model. In: ICML, Edinburgh, Scotland, UK (2012)
8. Leskovec, J.: Stanford large network dataset collection (2013), <http://snap.stanford.edu/data/index.html>
9. Ley, M.: Dbp - some lessons learned. *PVLDB* 2(2), 1493–1500 (2009)
10. Li, X., Bian, F., Crovella, M., Diot, C., Govindan, R., Iannaccone, G., Lakhina, A.: Detection and identification of network anomalies using sketch subspaces. In: ACM IMC, Rio de Janeiro, Brazil (2006)
11. Mongiovi, M., Bogdanov, P., Ranca, R., Singh, A.K., Papalexakis, E.E., Faloutsos, C.: Netspot: Spotting significant anomalous regions on dynamic networks. In: SDM, Austin, Texas, pp. 28–36 (2013)
12. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: Proc. SIGKDD, Washington, DC, USA, pp. 631–636 (2003)
13. Papalexakis, E.E., Akoglu, L., Ienco, D.: Do more views of a graph help? community detection and clustering in multi-graphs. In: Fusion, Istanbul, Turkey, pp. 899–905 (2013)
14. Rossi, R.A., Neville, J., Gallagher, B., Henderson, K.: Modeling dynamic behavior in large evolving graphs. In: WSDM, Rome, Italy, pp. 667–676 (2013)
15. Thottan, M., Ji, C.: Anomaly detection in ip networks. *IEEE Trans. on Signal Processing* 51(8), 2191–2204 (2003)
16. Wu, L., Wu, X., Lu, A., Zhou, Z.: A spectral approach to detecting subtle anomalies in graphs. *Journal of Intelligent Information Systems* 41, 313–337 (2013)