2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Predicting Community Evolution based on Time Series Modeling

Nagehan İlhan
Faculty of Computer and Informatics
Istanbul Technical University
Istanbul,Turkey
nilhan@itu.edu.tr

Şule Gündüz Öğüdücü
Faculty of Computer and Informatics
Istanbul Technical University
Istanbul,Turkey
sgunduz@itu.edu.tr

Abstract—**Communities in real life are usually dynamic and community structures evolve over time. Detecting community evolution provides insight into the underlying behavior of the network. A growing body of study is devoted in studying the dynamics of communities in evolving social networks. Most of them provide an event-based framework to characterize and track the community evolution. A part of these studies take a step further and provide a predictive model of the events by exploiting community features. However, the proposed models require the community extraction and computing the community features relevant to the time point to be predicted. In this paper, we proposed a new approach for predicting events by estimating feature values related to the communities in a given network. An event-based framework is used to characterize community behavior patterns. Then, a time series ARIMA model is used to predict how particular community features will change in the following time period. Distinct time windows are examined in constituting and analyzing time series. Our proposed approach efficiently tracks similar communities and identifies events over time. Furthermore, community feature values are forecasted with an acceptable error rate. Event prediction using forecasted feature values substantially match up with actual events.**

I. Introduction

Social networks consist of a set of vertices, joined in pairs by edges indicating the presence of a relationship between them. One of the key properties of social networks is modular or community structure which refers to an ensemble of densely connected inside and loosely interconnected groups of vertices. Since the dynamics of the communities play crucial role in understanding the topology of the networks, an impressive amount of study has been done on detecting community structures [1]–[3].

In many social networks, the interactions between communities evolve dynamically over time due to the fact that the actors represented as nodes in the network may have multiple roles and thus may change their communities over time. Moreover, there are new nodes or edges are added in the network, whereas some nodes are leaving. However, most studies on investigating the community structure of social networks have typically focused on static networks and neglected the temporal dynamics. Tracking communities in a network can reveal long-term trends of community evolution and patterns on how the underlying network evolve. The interactions established by the community members over time play an important role in shaping the future of the community. As a result of these dynamic interactions, several community events may happen such as: communities grow over time by acquiring new members or shrink by loosing existing members; new communities are emerging while old ones are disappearing; two or more communities can merge to form a new community or they can also split into smaller groups. Community evolution prediction aim at predicting these events and is beneficial from many aspects. It can help to predict the spread of diseases or information. For example, user comments and friendship ties can help us to monitor development of new trends, ideas, political views, etc. Wising up the futuristic knowledge of the community can assist to make accurate recommendations to the community members.

Different studies have been proposed to predict community evolution. Hopcroft *et al.* proposed a method using agglomerative clustering to identify and track stable clusters over time [4]. Tantipathananandh *et al.* proposed a framework consisting of several metrics for the analysis of dynamic social networks and explicitly makes use of temporal changes [5]. Chen *et al.* presented a representative-based approach to uncover distinct possible types of community-based anomalies in evolutionary networks such as grown, shrunken, merged, split, born, and vanished communities [6]. These mentioned studies concern extracting the evolutionary behaviors of the communities however they do not provide information about the future of the community. Gliwa *et al.* proposed a method, namely SCGI, for modeling group evolution and event prediction by utilizing leadership, density, cohesion and group size measures [7]. Ilhan *et al.* proposed a model for tracking the evolutionary dynamics of communities with a broad range of structural features which results in a better prediction accuracy for community events in social networks [8]. However, these studies utilize community features as attributes to classify predetermined community instances to the corresponding event. Usually, these approaches are based on the extraction of the community structure at each time step and then predict the labels of the last time step communities using the features belonging to the corresponding time step.

In this work, we have proposed a novel approach to accurately predict the next event of a community with employing time series ARIMA model. Our main contribution is effectively predicting community events through community feature forecasting. The proposed model directly predicts community features at the next time step thus it avoids discovering communities from scratch. Our approach can be summarized as follows: an evolving network is represented by a series of static

ASONAM '15, August 25-28, 2015, Paris, France
© 2015 ACM. ISBN 978-1-4503-3854-7/15/08 $15.00
DOI: http://dx.doi.org/10.1145/2808797.2808913

snapshots where each snapshot corresponds to a particular point in time by covering the interactions up to that specific time. A community detection algorithm is applied to discover communities. Then, the structural features are extracted by measuring a large scale of the community properties. The set of communities at consecutive snapshots are matched with each other using a similarity measure and significant events of the communities, such as survive, growth, shrink, merge, split and dissolve are identified. A time series is built for the last snapshot communities those the events will be predicted, recording the feature values of the matching communities from past to present using landmark window technique. Each community possesses time series as the number of features. Afterwards, ARIMA is applied to the time series in order to predict next values. The community instances composed of the forecasted feature values are tested on the model to predict event. The process is repeated for distinct window values and both the success rate of the community event prediction model and the matching up rate of predicted events with the actual events are examined.

In order to testify the proposed approach, experiments are performed on two co-authorship networks extracted from two sections of arXiv[1]. Louvain community detection algorithm is used to extract communities. Furthermore, the proposed model is experimented on time series arranged with various window lengths. The experiments verified that our model predicts community events accurately and forecasts community features with an acceptable error rate.

The rest of the paper is organized as follows: we first briefly review literatures on studies of community evolution. Section III describes the proposed approach. The experimental process and discussion of the results is given in Section IV. Finally, Section V concludes the paper and presents future directions.

II. Related Work

Studying the evolution of social networks has driven significant interest by many researchers [9]–[13]. Kumar *et al.* presented a simple model of network growth which captures several aspects of component structure such as: singletons, isolated communities, and a giant component [11]. Backstrom *et al.* considered the structural features that influence the group growth for a period of time instead of focusing on the component structure [10]. They have discovered simple structural features that influence an individual to join a particular group or a given group to grow significantly. However, they have not considered community event prediction. One of the fundamental issue in community evolution is to track the incremental changes of community structure by defining events pertaining to the communities so as to derive information of community interaction over each time step. There are also several studies on the community event prediction. Asur *et al.* formulated a set of critical events between detected communities at two consecutive snapshots [9]. However, they did not cover all forms of events that may occur in a life-cycle of a particular community. Takaffoli *et al.* expanded the set of events by considering new event types [14]. In their later study, event-based investigation of the aggregated graph of the network is improved to track the changes of communities, not only between two consecutive snapshots but also encompassing multiple snapshots [13]. They also analyzed the behavior of the individuals by considering node-specific events and behavioral metrics. Bródka *et al.* proposed a framework for modeling community evolution and event prediction by utilizing node numbers of the communities [15]. Huang *et al.* proposed an approach by incorporating the activity features in measuring the influence of member activities to predict the network evolution [16]. However, the prediction accuracy is relatively low in these studies, since they exploit a limited number of structural features such as node number of communities or activity of community members in prediction. The authors in [8] proposed a framework for tracking the evolutionary dynamics of communities in social networks by covering wide range of community features which performs improved prediction accuracy.

Common approaches characterize the evolution of communities by describing a series of critical events undergone by communities over time implementing community detection algorithm to each snapshot of the evolving network so-called event based two-step approach. This approach can deal with newly available network data and provides evolution model by incorporating key features i.e. history, structure, active members, influential members etc. related to the community. However, it needs community extraction and computing the features belonging to the communities of the corresponding snapshot which will be predicted. In this study, we propose a model which avoids applying community detection algorithm and calculating community features for the relevant snapshot where its evolution will be predicted. So as to do, we have utilized time series analysis model ARIMA to forecast precise community feature values, thereby classify the communities to the related events. To our knowledge, there is no research directly regarding event prediction through the estimation of community features. But, surely several research may be found regarding time series prediction and classification on distinct kind of data typically on link prediction. For instance, Huang *et al.* proposed a model by fitting the occurrence of links between the nodes of the network along time into time series, using ARIMA to project their future values and to measure the probability of new connections [17]. The authors in [18] proposed an approach to perform prediction of new links by addressing the evolution of topological metrics as a time series problem. They used a set of well-known statistical forecasting models to estimate future values. In another study, prediction of global social network measure values in particular by using time series forecasting has been performed instead of individual link forecasting [19]. However, none of them are concerning the community features and events.

Besides, in establishing time series, we employ landmark window approach which relies on the specified amount of the whole past thus allowing us to capture persistent communities. In spite of existing works on the problem of dynamic network analysis, limited number of studies have explicitly explored the impact of selecting different time window lengths. Greene *et al.* [20] states that the size of the time-step window may affects the obtained results, especially in unstable network structure and Kawadia *et al.* [21] point out the importance of determining the granularity of the temporal snapshots for the purpose of detecting temporal communities. However, none of these studies addressed the influence of time window within

[1] http://www.arxiv.org

the scope of time series estimation of community features. Our model handles distinct window granularities in building time series, thus reveals how long the past should be considered in order to provide best estimation results.

III. Proposed Approach

In this paper, we study the community event prediction problem by use of time series analysis of temporal and structural features of the communities. Our model proceeds as follows: an evolving network is represented by a series of static snapshots where each snapshot corresponds to a particular point in time. The communities of a social network are determined using a well known community detection algorithm (Louvain [22]) and a broad range of structural and temporal features are extracted. The extracted community features cover many properties of both the internal link structure and the external interaction of the community with the rest of the network. After identifying matching communities at consecutive time steps, these communities are labeled with the events such as survive, growth, shrink, merge, split and dissolve according to their change rate at consecutive time steps. A window length is defined to generate time series for each feature, thus for each community time series is build as the number of features. Then, ARIMA model is applied to estimate the next values of the features. Using the communities with the forecasted feature values as the test set, we trained several well known classification algorithms on the rest of the window length snapshots as training set. The details of the proposed approach is given in this section.

Definition 1: A graph $G(V, E)$ denotes the sets of nodes V and edges E in a network. Evolving graph G is described over a time period $[0..T]$ and it will be decomposed into a sequence of static snapshots $G^{[0,\epsilon]}, ..., G^{[T-\epsilon,T]} = G^1, ..., G^n$. ϵ is the discretization factor which will be adjusted depending on the granularity of the time stamps and $G^{[t+\epsilon]}$ is the graph containing all nodes and edges involved during the time period $[0, t + \epsilon]$.

A. Community Detection

Definition 2: While G^i representing the ith snapshot of the graph, $C^i = \{C^i_1, C^i_2 ...., C^i_k\}$ represents the set of communities of graph in that snapshot i.

Community evolution analysis starts with community detection. We used Louvain which is a well known community detection algorithm to obtain communities [22]. Louvain is a hierarchical greedy algorithm which is composed of two phase. Initially, each node is assigned to a community on its own. In the first phase, nodes are reassigned to neighboring communities in a local and greedy manner by maximizing the modularity gain. The process repeated until no nodes can be reassigned. In the second phase, each community is considered as a node on its own. Then, the algorithm starts the phase one and so on.

B. Community Feature Extraction

The second stage is the feature extraction in which we extract community features that may be important in tracking community evolution and measure structural and temporal aspects of the communities. We calculated nine different community features within the scope of the model. Each measurement corresponds to a dimension in our feature space and the details are given in Table I.

TABLE I. Community Features

No	Feature	Desription	Formula
f_1	Node Number (Nodes)	Number of the nodes in the community number i at time t.	n^t_i
f_2	Edge Number (Edges)	Number of the edges in the community number i at time t.	e^t_i
f_3	Intra Community Edges (Intra)	Ratio of the total number of edges between the nodes inside the community ($e^t_i(in)$) to the number of nodes in the community.	$\frac{e^t_i(in)}{n^t_i}$
f_4	Inter Community Edges (Inter)	Ratio of the total number of edges of the nodes that connected with outside of the community ($e^t_i(out)$) to the number of nodes in the community.	$\frac{e^t_i(out)}{n^t_i}$
f_5	Activeness	Ratio of the total number of connections that has been done in the previous timestamp by the nodes of the community (a^t_i) to the number of nodes in the community.	$\frac{a^t_i}{n^t_i}$
f_6	Aging	Ratio of the total ages of the nodes in the community o^t_i to the number of nodes in the community. Ages of the nodes increased by 1 at each timestamp by starting zero.	$\frac{o^t_i}{n^t_i}$
f_7	Betweenness	Ratio of the total node betweenness in the community b^t_i to the number of nodes in the community.	$\frac{b^t_i}{n^t_i}$
f_8	Degree	Ratio of the sum of the degree's of the nodes in the community d^t_i to the number of nodes in the community.	$\frac{d^t_i}{n^t_i}$
f_9	Conductance	Ratio of the number of edges in the community to the sum of the degrees of the nodes in the community.	$\frac{e^t_i}{d^t_i}$

C. Community Similarity

In order to pursue evolution of a community from one time step to the following time steps and measure the similarity between two communities in consecutive time steps, a matching metric should be used. We consider two communities at consecutive snapshots are similar if the ratio of their mutual members exceeds a threshold [23], formally:

$$Sim(C^t_i, C^{t+1}_j) = min\left(\frac{|C^t_i \cap C^{t+1}_j|}{|C^t_i|}, \frac{|C^t_i \cap C^{t+1}_j|}{|C^{t+1}_j|}\right) \geq \Theta \quad (1)$$

Two communities match up if their similarity value $Sim(C^t_i, C^{t+1}_j)$ exceeds a user set similarity threshold Θ.

D. Event Detection

We consider six events including *growth*, *survive*, *shrink*, *split*, *merge* and *dissolve* to capture the changes of a community where the event types are illustrated in Fig. 1. The output of the matching process between C^t_i and C^{t+1}_j reveals the corresponding community event with the predefined matching thresholds. We defined two distinct thresholds: Similarity Threshold (θ) and Fluctuation Threshold (ϕ). The output of the matching process between C^t_i and one of the communities obtained at time step $t + 1$ will reveal a series of community evolution events which are used as class label of C^t_i in the classification stage of the model. Event detection procedure works as follows: for a given community C^t_i, the similarity between C^t_i and at least one of the successor communities at time step $t + 1$ should be greater than θ in order to be labeled with an event except *dissolve*. If a community has one successor, it may has one of the three possible events $\{survive, growth, shrink\}$. Herein, we propose a metric, namely *fluctuation*, on the purpose of computing the percentage of increase/decrease in the number of community members. Fluctuation rate would give us an accurate result of whether i) the community has been grown (i.e there is a

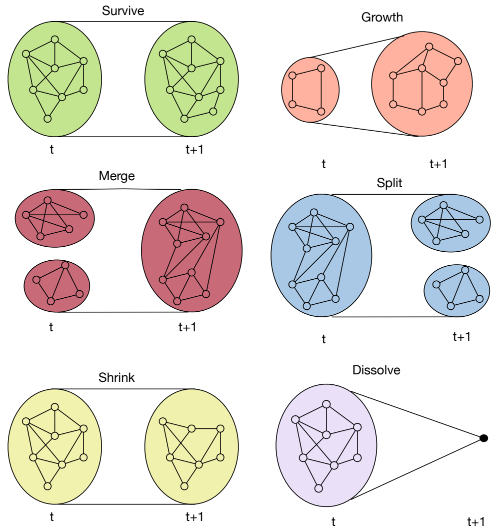

Fig. 1. Event Types

substantial percentage increase in the number of members), or ii) the community has been survived (i.e there is a negligible increase/decrease in the number of members), or iii) the community has been shrunk (i.e there is a substantial percentage decrease in the number of members). More formally, given a community C_i^t has n_i^t members at time snapshot *t* and successor community C_j^{t+1} has n_j^{t+1} members at time snapshot *t+1*, the fluctuation is defined as:

$$fluctuation(C_i^t, C_j^{t+1}) = \frac{n_j^{t+1}}{n_i^t} - 1 \tag{2}$$

We could then label a community as being the survival, grown, or shrunk as follows:

$$label = \begin{cases} \textbf{shrink} & \text{if } fluctuation(C_i^t, C_j^{t+1}) < -\phi \\ \textbf{survive} & \text{if } -\phi \leq fluctuation(C_i^t, C_j^{t+1}) \leq \phi \\ \textbf{growth} & \text{if } fluctuation(C_i^t, C_j^{t+1}) > \phi \end{cases}$$

A community C_i^t at time t may match with a set of communities $C_*^{t+1} = \{C_1^{t+1}...C_j^{t+1}\}$ in a later snapshot in *split* case or a set of communities $C_*^t = \{C_1^t...C_i^t\}$ may match to a community C_j^{t+1} in the subsequent snapshot $t+1$ in *merge* case. In the case where there is no similar community at a later snapshot which means θ is not exceeded, then it is assumed that the community dissolves.

In the following, we provide the formal definitions of these events: The formal definitions of the events are described as follows:

Survive: A community C_i^t at time t is said to be labeled with survive event if there exist a community C_j^{t+1} at time $t+1$ whose similarity is greater than predefined θ and fluctuation falls between $-\phi$ and ϕ. Thus, C_i^t has survived if:

$$Sim(C_i^t, C_j^{t+1}) \geq \theta \wedge -\phi \leq fluctuation(C_i^t, C_j^{t+1}) \leq \phi \tag{3}$$

Growth: A community C_i^t at time t is said to be labeled with growth event if there exist a community C_j^{t+1} at time $t+1$ whose similarity is greater than predefined θ and fluctuation is greater than ϕ. Thus, C_i^t has grown if:

$$Sim(C_i^t, C_j^{t+1}) \geq \theta \wedge fluctuation(C_i^t, C_j^{t+1}) > \phi \tag{4}$$

Shrink: A community C_i^t at time t is said to be labeled with shrink event if there exist a community C_j^{t+1} at time $t+1$ whose similarity is greater than predefined θ and fluctuation is smaller than $-\phi$. Thus, C_i^t has shrunk if:

$$Sim(C_i^t, C_j^{t+1}) \geq \theta \wedge fluctuation(C_i^t, C_j^{t+1}) < -\phi \tag{5}$$

Split: Community C_i^t is said to be split to $C_*^{t+1} = \{C_1^{t+1}...C_j^{t+1}\}$ and has split event if similarity between C_i^t and each C_*^{t+1} and also similarity between C_i^t and the union of two or more communities in the set C_*^{t+1} is greater than $theta$. Thus, C_i^t has splitted if:

$$\forall C_j^{t+1} \in C_*^{t+1}, Sim(C_i^t, C_j^{t+1}) \geq \theta \wedge Sim(C_i^t, \cup\{C_*^{t+1}\}) \geq \theta \tag{6}$$

Merge: A set of community $C_*^t = \{C_1^t...C_i^t\}$ is said to be merged to C_j^{t+1} and have merge event if similarity between each community in C_*^t and C_j^{t+1} and also similarity between the union of the communities in C_*^t and C_j^{t+1} is greater than θ. Thus, a set of community C_*^t has merged if:

$$\forall C_i^t \in C_*^t, Sim(C_i^t, C_j^{t+1}) \geq \theta \wedge Sim(\cup\{C_*^t\}, C_j^{t+1}) \geq \theta \tag{7}$$

Dissolve: A community C_i^t at time t is said to be labeled with dissolve event if there is not matching community at time $t+1$ whose similarity threshold is greater than θ. Thus, C_i^t has dissolved if:

$$Sim(C_i^t, C_j^{t+1} < \theta) \tag{8}$$

E. Time Series Analysis

Let the dynamic social network be an ordered sequence of T graphs $\{G^1, ..., G^T\}$, where $G^i(V^i, E^i)$ represents a static snapshot of the network observed at time point t_i. In our proposed model, we use the landmark window approach where a predefined number of static graphs are analyzed beginning from the one obtained at the initial time step for the temporal analysis. As illustrated in Fig. 2, landmark window model retains the historical information of the network as long as the window length. This model succeeds in finding persistent communities by considering the entire past. Since we use time series analysis in our model, incorporating the whole historical knowledge related to community may improve prediction. Let w be the window length, then our task is to predict the community events of subsequent snapshot t_{w+1} of the window length w by concerning the data of $[t_1, t_w]$. Thus, G^{w+1} is predicted using the knowledge extracted from $[G^1, ..., G^w]$ while $w+1 \leq T$.

To quantitatively characterize the development process of community features, we apply time series analysis to model the changes of the nine features (summarized in Table I) over time. In our proposed approach, we build time series for each community feature of the set $\{f_1, f_2, .., f_9\}$ related to community C_i^{w+1}. Then, as instance, the time series of f_1

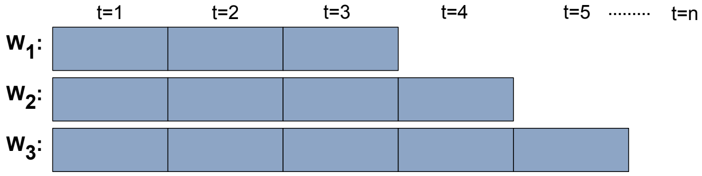

Fig. 2. Landmark Window

of C_i^{w+1} become $TS^1 = \{f_1^1, f_1^2, ..., f_1^w\}$. Being one of the pioneers and mostly used methods in prediction techniques, ARIMA model [24] is used to predict f_1^{w+1}.

In ARIMA model, the future value of a variable is a linear combination of past values and past errors and can be written as:

$$y_t = \sum_{i=1}^{p} \alpha_i yt - i + \epsilon_t + \sum_{i=1}^{q} \beta_i \epsilon_{t-i}, \tag{9}$$

where $y_t = (1 - B)^d x_t$, B is the back shift operator, x_t is the trend, α_i is the auto regressive coefficients, β_i is moving average coefficients and ϵ_t is the residual, uncorrelated white noise with zero mean and constant variance σ^2, and p,d, and q are the order of each parameters. We applied automatic time series ARIMA model [25] of R package [26] in our experimentation.

F. Classification

At the last step, we adopt the well-known classifiers in order to determine corresponding event labels: J48 Decision Tree, Nearest Neighbor (KNN), Random Forest, Random Tree and Kstar classifiers[2]. The sequence of structural features and identified event class label of a community correspond to an instance and used as one of the input parameters for the classifiers. The instances obtained from $[G^1, ..., G^w]$ are used as the training set for the classifiers. The goal of classification is to predict the next event of a given community C_i^{w+1} at time step $w+1$ given the predicted features of that community. The process is repeated for a specified set of w values.

IV. Experimental Study

In this section, we first describe the datasets used in the study and experimental setup. Prediction results of the event detection procedure and results obtained by our proposed community event prediction methodology using time series analysis are presented.

A. Datasets

Two citation networks, hepTh and hepPh are experimented in the study. The hepTh and hepPh are the collaboration networks from e-print arXiv covers scientific collaborations between authors' papers submitted to High Energy Physics Theory and High Energy Physics Phenomenology respectively. Table II shows detailed information about the datasets. We considered a five year period in each dataset by taking three months interval snapshots being twenty time steps and nineteen evolution transitions in total.

TABLE II. Datasets

Name	No. of papers	No. Of citations	Time period
hepTh	22753	128744	1993-1998
hepPh	28074	310495	1992-1997

B. Results

The results are evaluated as follows: first, the classifier is trained with instances obtained from the graphs $[G_1, G_2, ..., G_w]$. Then time series model is used to forecast the feature values of the communities at time step $w + 1$. Given these feature values, the classification model predicts the event labels of the communities at that time step. In order to evaluate the accuracy of the model's predictions, we extracted the communities at time step $w + 1$ using the community detection algorithm. Then, the communities at time step $w+1$ are labeled with the corresponding community events using the communities at time step w. We denote these labels as actual labels of the communities at time $w + 1$. Then, the model's prediction is evaluated in terms of precision, recall, F-Measure and accuracy by comparing the model's prediction with actual class labels. We conducted our experiments with different w values ranging between 10 and 18.

Adjusting the appropriate similarity thresholds is one of the challenges in the study of community evolution. In this paper, we employ two thresholds: similarity threshold (θ) and fluctuation threshold (ϕ) respectively. A low θ may lead to a significant number of matching communities, while high values of θ results in more dissolutions. Similarly, low value of ϕ results in a small amount of survival and excessive number of grown and shrunken communities. The number of survive event increases as ϕ increases. To select the optimal thresholds, we have trained distinct θ and ϕ values and observed event rates. Our setting is ($\theta = 0.3$ and $\phi = 0.15$) for hepTh and ($\theta = 0.25$ and $\phi = 0.10$) for hepPh dataset which yields comparatively balanced event distributions.

The classification results of the event detection model of hepTh dataset are represented in Table III.

Our results in Table III indicate that the event prediction model accurately predicts community events with accuracy as high as **%89**. We can also observe that the minimum prediction accuracy is **%76** which can also be designated as successful. The classifier results of the event detection model of hepPh dataset are reported in Table IV. Similar to hepTh, the model has performed higher prediction results. The best accuracy for the hepPh dataset is **%86** while the worst being **%74**. Besides, we can observe that the prediction results exhibits a small amount of fluctuation with respect to the window size in both datasets. It is hard to say that the accuracy increases/decreases with the windows size. The results fluctuate due to the dynamic nature of the networks.

We display the temporal change in the networks by observing the rate of the number of vertices to edges over time in Fig. 3. The hepPh dataset densifies rapidly while hepTh showing more stationary characteristics over time.

[2] The WEKA Data Mining implementation of the classifiers [27]

TABLE III. RESULTS OF HEPTH

		J48	KNN	Random Forest	Random Tree	KStar
w=10	Precision	0.81	0.87	0.85	0.88	0.82
	Recall	0.83	0.87	0.86	0.88	0.82
	F-measure	0.81	0.86	0.85	0.87	0.81
	Accuracy	0.83	0.87	0.86	0.88	0.82
w=11	Precision	0.82	0.88	0.89	0.89	0.89
	Recall	0.83	0.88	0.89	0.89	0.89
	F-measure	0.82	0.88	0.89	0.89	0.89
	Accuracy	0.82	0.88	0.89	0.89	0.89
W=12	Precision	0.77	0.81	0.84	0.82	0.82
	Recall	0.78	0.82	0.84	0.83	0.83
	F-measure	0.77	0.81	0.83	0.82	0.82
	Accuracy	0.78	0.82	0.84	0.83	0.83
w=13	Precision	0.75	0.80	0.83	0.83	0.82
	Recall	0.77	0.81	0.83	0.83	0.83
	F-measure	0.76	0.81	0.83	0.83	0.82
	Accuracy	0.76	0.81	0.83	0.83	0.83
w=14	Precision	0.75	0.80	0.80	0.82	0.81
	Recall	0.76	0.82	0.82	0.82	0.82
	F-measure	0.76	0.81	0.81	0.82	0.81
	Accuracy	0.76	0.81	0.81	0.82	0.82
w=15	Precision	0.79	0.85	0.84	0.84	0.83
	Recall	0.80	0.85	0.85	0.85	0.84
	F-measure	0.79	0.85	0.84	0.84	0.84
	Accuracy	0.80	0.85	0.85	0.85	0.84
w=16	Precision	0.82	0.82	0.84	0.84	0.82
	Recall	0.82	0.83	0.85	0.84	0.83
	F-measure	0.81	0.82	0.84	0.84	0.82
	Accuracy	0.82	0.83	0.85	0.84	0.83
w=17	Precision	0.80	0.86	0.86	0.86	0.86
	Recall	0.81	0.86	0.86	0.86	0.86
	F-measure	0.80	0.86	0.86	0.86	0.86
	Accuracy	0.81	0.86	0.86	0.86	0.86
w=18	Precision	0.78	0.85	0.83	0.85	0.84
	Recall	0.79	0.86	0.84	0.84	0.85
	F-measure	0.78	0.85	0.83	0.85	0.84
	Accuracy	0.79	0.86	0.84	0.84	0.84

TABLE IV. RESULTS OF HEPPH

		J48	KNN	Random Forest	Random Tree	KStar
w=10	Precision	0.74	0.81	0.81	0.77	0.84
	Recall	0.75	0.82	0.81	0.78	0.85
	F-measure	0.74	0.81	0.80	0.77	0.84
	Accuracy	0.75	0.82	0.81	0.78	0.85
w=11	Precision	0.82	0.82	0.84	0.83	0.82
	Recall	0.82	0.82	0.84	0.84	0.83
	F-measure	0.81	0.81	0.84	0.83	0.82
	Accuracy	0.82	0.82	0.84	0.84	0.83
w=12	Precision	0.79	0.84	0.81	0.83	0.77
	Recall	0.80	0.84	0.82	0.82	0.79
	F-measure	0.79	0.83	0.81	0.82	0.78
	Accuracy	0.80	0.84	0.82	0.82	0.79
w=13	Precision	0.84	0.83	0.86	0.88	0.84
	Recall	0.83	0.83	0.86	0.86	0.85
	F-measure	0.83	0.83	0.85	0.86	0.84
	Accuracy	0.83	0.83	0.86	0.86	0.85
w=14	Precision	0.74	0.83	0.82	0.83	0.82
	Recall	0.75	0.84	0.82	0.83	0.81
	F-measure	0.74	0.83	0.82	0.82	0.80
	Accuracy	0.75	0.84	0.82	0.83	0.81
w=15	Precision	0.74	0.83	0.84	0.81	0.83
	Recall	0.75	0.83	0.83	0.82	0.83
	F-measure	0.74	0.82	0.82	0.81	0.83
	Accuracy	0.75	0.83	0.83	0.82	0.83
w=16	Precision	0.75	0.81	0.84	0.79	0.83
	Recall	0.75	0.82	0.84	0.81	0.84
	F-measure	0.74	0.82	0.83	0.79	0.83
	Accuracy	0.75	0.81	0.84	0.80	0.84
w=17	Precision	0.73	0.78	0.83	0.79	0.80
	Recall	0.74	0.79	0.84	0.80	0.81
	F-measure	0.73	0.77	0.83	0.78	0.79
	Accuracy	0.74	0.79	0.84	0.80	0.81
w=18	Precision	0.75	0.81	0.85	0.83	0.80
	Recall	0.76	0.81	0.85	0.83	0.81
	F-measure	0.75	0.80	0.84	0.82	0.80
	Accuracy	0.76	0.81	0.85	0.83	0.81

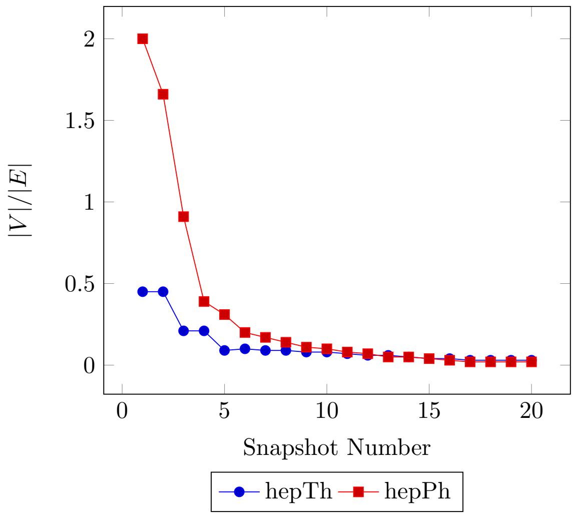

Fig. 3. $|V|/|E|$ Over Time

In order to show the performance of time series analysis approach, we first quantify the Mean Absolute Percentage Error (MAPE) rate of the forecasted feature values and the eventual feature values. We compute MAPE by taking average of the each estimated feature values of the communities related to snapshot $w + 1$ those are forecasted using window size w. Fig. 4 provides the MAPE values of the datasets with respect to the examined window size. The results expose that time series analysis works well with hepTh network with minimum %11 and maximum %24 error rate. In hepPh network, a higher amount of error rate is obtained with minimum being %17 and maximum being %51.

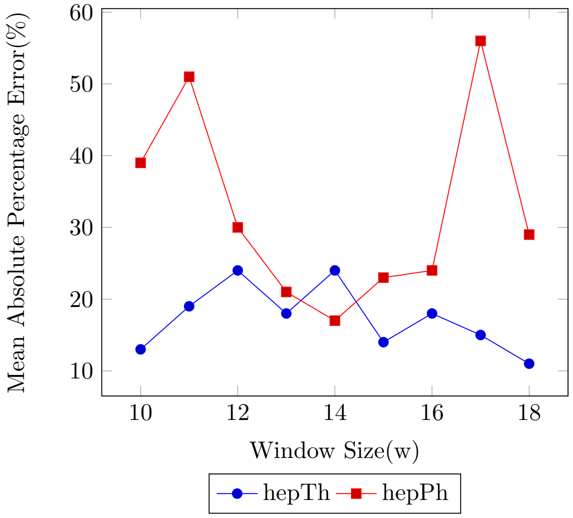

Fig. 4. Mean Absolute Percentage Error Results

The time series analysis results of the hepTh and hepPh have shown in Table V and Table VI respectively. These results represent the matching up values of the actual events determined by the model and the events identified using estimated feature values.

TABLE V. RESULTS OF HEPTH TIME SERIES ANALYSIS

		J48	KNN	Random Forest	Random Tree	KStar
w=10	Precision	0.91	0.90	0.89	0.89	0.89
	Recall	0.89	0.89	0.89	0.89	0.89
	F-measure	0.90	0.89	0.89	0.89	0.89
	Accuracy	0.89	0.89	0.89	0.89	0.89
w=11	Precision	0.85	0.84	0.84	0.84	0.84
	Recall	0.86	0.85	0.85	0.85	0.85
	F-measure	0.85	0.84	0.84	0.84	0.84
	Accuracy	0.86	0.85	0.85	0.85	0.85
w=12	Precision	0.82	0.81	0.81	0.81	0.81
	Recall	0.83	0.82	0.82	0.82	0.82
	F-measure	0.81	0.80	0.80	0.80	0.80
	Accuracy	0.82	0.82	0.82	0.82	0.82
w=13	Precision	0.79	0.79	0.79	0.79	0.79
	Recall	0.80	0.80	0.81	0.81	0.80
	F-measure	0.78	0.78	0.79	0.79	0.78
	Accuracy	0.80	0.80	0.81	0.81	0.80
w=14	Precision	0.79	0.79	0.79	0.78	0.78
	Recall	0.80	0.80	0.80	0.80	0.80
	F-measure	0.78	0.78	0.78	0.78	0.78
	Accuracy	0.80	0.80	0.80	0.80	0.80
w=15	Precision	0.78	0.78	0.78	0.78	0.78
	Recall	0.79	0.79	0.79	0.79	0.79
	F-measure	0.78	0.77	0.78	0.78	0.78
	Accuracy	0.79	0.79	0.79	0.79	0.79
w=16	Precision	0.78	0.78	0.78	0.78	0.78
	Recall	0.79	0.79	0.79	0.79	0.79
	F-measure	0.77	0.77	0.77	0.77	0.78
	Accuracy	0.79	0.79	0.79	0.79	0.79
w=17	Precision	0.78	0.78	0.79	0.79	0.79
	Recall	0.79	0.79	0.79	0.79	0.79
	F-measure	0.77	0.77	0.78	0.78	0.78
	Accuracy	0.79	0.79	0.79	0.79	0.79
w=18	Precision	0.79	0.79	0.79	0.79	0.79
	Recall	0.79	0.79	0.79	0.79	0.79
	F-measure	0.78	0.78	0.78	0.78	0.78
	Accuracy	0.79	0.79	0.79	0.79	0.79

TABLE VI. RESULTS OF HEPPH TIME SERIES ANALYSIS

		J48	KNN	Random Forest	Random Tree	Kstar
w=10	Precision	0.74	0.78	0.78	0.76	0.73
	Recall	0.57	0.57	0.59	0.59	0.56
	F-measure	0.56	0.59	0.62	0.63	0.60
	Accuracy	0.57	0.57	0.59	0.59	0.56
w=11	Precision	0.69	0.69	0.71	0.71	0.68
	Recall	0.55	0.55	0.57	0.58	0.55
	F-measure	0.58	0.58	0.60	0.61	0.59
	Accuracy	0.55	0.55	0.57	0.58	0.55
w=12	Precision	0.68	0.70	0.70	0.69	0.65
	Recall	0.55	0.57	0.58	0.58	0.55
	F-measure	0.58	0.60	0.60	0.60	0.57
	Accuracy	0.55	0.57	0.58	0.58	0.55
w=13	Precision	0.69	0.69	0.69	0.69	0.65
	Recall	0.58	0.59	0.60	0.60	0.56
	F-measure	0.60	0.61	0.61	0.62	0.58
	Accuracy	0.58	0.59	0.60	0.60	0.56
w=14	Precision	0.69	0.70	0.71	0.72	0.68
	Recall	0.60	0.61	0.62	0.63	0.59
	F-measure	0.62	0.63	0.64	0.65	0.61
	Accuracy	0.60	0.61	0.62	0.63	0.58
w=15	Precision	0.71	0.72	0.72	0.72	0.68
	Recall	0.62	0.62	0.63	0.63	0.59
	F-measure	0.65	0.65	0.65	0.65	0.62
	Accuracy	0.62	0.62	0.63	0.63	0.59
w=16	Precision	0.72	0.72	0.72	0.72	0.67
	Recall	0.63	0.63	0.63	0.62	0.58
	F-measure	0.65	0.65	0.65	0.65	0.60
	Accuracy	0.63	0.63	0.63	0.62	0.58
w=17	Precision	0.70	0.70	0.70	0.70	0.66
	Recall	0.62	0.62	0.62	0.62	0.57
	F-measure	0.64	0.64	0.65	0.65	0.60
	Accuracy	0.62	0.62	0.62	0.62	0.57
w=18	Precision	0.70	0.70	0.70	0.71	0.67
	Recall	0.62	0.62	0.62	0.62	0.58
	F-measure	0.64	0.64	0.65	0.65	0.60
	Accuracy	0.62	0.62	0.62	0.62	0.58

We can easily see from Table V that the predicted events using time series analysis model highly overlap with the actual events in hepTh dataset. The accuracy values vary between **%89** as the highest and **%79** as the lowest. As it can be seen from Table VI, the hepPh dataset accuracy results are relatively lower ranging between **%63** and **%55**. At the same time, we observe precision is rather high, but the recall is low. Thus, we can infer that some events predicted successfully but as far not in all instances. Also, it can be seen that in hepTh dataset, there exists a small decay in accuracy results as the window size increases. In hepPh dataset, the accuracy results are fluctuating.

To sum up, we have shown that our community event detection procedure successfully identifies the events in various window sizes. The proposed event identification through time series analysis approach performed well in hepTh and nearly good in hepPh dataset. As it is shown in Fig. 3 and it is also evident from the results that the datasets are highly dynamic and especially hepPh network performs fluctuative behavior. It could be better to reduce the snapshot interval (i.e one month instead of three months) so as to capture the changes and attain reliable results in highly dynamic networks. Sliding window technique which is embedded with forgetting mechanism that drop older data may also be an alternative in investigation of highly dynamic networks.

V. CONCLUSION

We introduce a methodology to predict community dynamics in evolving networks, based on time series ARIMA model. We first describe our event-based model to characterize community behavioral patterns. Within the scope of the model, we define and extract a set of features covering structural and temporal aspects of the communities. In event prediction, we propose using forecasted community feature values by ARIMA instead of computing features on extracted communities.

We first give the performance of our event-based model and the results indicate that our model accurately predicts community events with various window sizes. We then present the results obtained using time series based methodology. Time series based approach identifies events accurately in particular hepTh dataset. Consistency of the results with various window sizes differ highly depending on the underlying network.

Our future research may be experimenting the proposed methodology with distinct time sliced snapshots since it could be better to observe changes in narrow intervals. Implementing sliding window technique may also perform better results.

REFERENCES

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] A. Clauset, M. E. J. Newman, , and C. Moore, "Finding community structure in very large networks," *Physical Review E*, pp. 1– 6, 2004. [Online]. Available: www.ece.unm.edu/ifis/papers/community-moore.pdf

[3] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, June 2005. [Online]. Available: http://dx.doi.org/10.1038/nature03607

[4] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101,

no. Suppl 1, pp. 5249–5253, Apr. 2004. [Online]. Available: http://www.pnas.org/content/101/suppl.1/5249.abstract

[5] 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*, ser. KDD ’07. New York, NY, USA: ACM, 2007, pp. 717–726. [Online]. Available: http://doi.acm.org/10.1145/1281192.1281269

[6] Z. Chen, W. Hendrix, and N. F. Samatova, “Community-based anomaly detection in evolutionary networks,” *J. Intell. Inf. Syst.*, vol. 39, no. 1, pp. 59–85, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1007/s10844-011-0183-2

[7] B. Gliwa, P. Bródka, A. Zygmunt, S. Saganowski, P. Kazienko, and J. Kozlak, “Different approaches to community evolution prediction in blogosphere.” *CoRR*, vol. abs/1306.3517, 2013.

[8] N. İlhan and Şule Gündüz Öğüdücü, “Community event prediction in dynamic social networks,” *Machine Learning and Applications*, vol. 2, pp. 269–274, 2013.

[9] S. Asur, S. Parthasarathy, and D. Ucar, “An event-based framework for characterizing the evolutionary behavior of interaction graphs.” in *KDD*, P. Berkhin, R. Caruana, and X. Wu, Eds. ACM, 2007, pp. 913–921.

[10] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, “Group formation in large social networks: Membership, growth, and evolution,” in *Proceedings of KDD’06*, 2006.

[11] R. Kumar, J. Novak, and A. Tomkins, “Structure and evolution of online social networks,” in *KDD ’06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2006, pp. 611–617.

[12] G. Palla, A.-L. Barabasi, and T. Vicsek, “Quantifying social group evolution,” *Nature*, vol. 446, no. 7136, pp. 664–667, April 2007.

[13] M. Takaffoli, F. Sangi, J. Fagnan, and O. R. Zaïane, “Community evolution mining in dynamic social networks,” *Procedia - Social and Behavioral Sciences*, vol. 22, no. 0, pp. 49 – 58, 2011.

[14] ——, “A framework for analyzing dynamic social networks,” 2010.

[15] P. Bródka, P. Kazienko, and B. Koloszczyk, “Predicting group evolution in the social network.” in *SocInfo*, ser. Lecture Notes in Computer Science, vol. 7710. Springer, 2012, pp. 54–67.

[16] S. Huang and D. Lee, “Exploring structural features in predicting social network evolution,” *Machine Learning and Applications*, vol. 2, pp. 269–274, 2011.

[17] Z. Huang and D. K. J. Lin, “The time-series link prediction problem with applications in communication surveillance,” *INFORMS Journal on Computing*, pp. 286–303, 2009.

[18] P. R. da Silva Soares and R. B. C. Prudêncio, “Time series based link prediction.” in *IJCNN*. IEEE, 2012, pp. 1–7.

[19] R. Michalski, P. Kazienko, and D. Krol, “Predicting social network measures using machine learning approach.” in *ASONAM*. IEEE Computer Society, 2012, pp. 1056–1059.

[20] D. Greene, D. Doyle, and P. Cunningham, “Tracking the evolution of communities in dynamic social networks,” in *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 176–183.

[21] V. Kawadia and S. Sreenivasan, “Online detection of temporal communities in evolving networks by estrangement confinement,” *CoRR*, vol. abs/1203.5126, 2012.

[22] V. Blondel, J. Guillaume, R. Lambiotte, and E. Mech, “Fast unfolding of communities in large networks,” *J. Stat. Mech*, p. P10008, 2008.

[23] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, “Tracking evolving communities in large linked networks,” *Proceedings of the National Academy of Sciences*, vol. 101, pp. 5249–5253, April 2004.

[24] G. E. Box and G. M. Jenkins, “Some recent advances in forecasting and control,” *Applied Statistics*, pp. 91–109, 1968.

[25] R. J. Hyndman and Y. Kh, “Automatic time series forecasting: The forecast package for r,” *Journal of Statistical Software*, 2008.

[26] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278