ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

COMMUNITY EVENT PREDICTION IN EVOLVING SOCIAL NETWORKS

Ph.D. THESIS

Nagehan İLHAN

Department of Computer Engineering

Computer Engineering Programme

NOVEMBER 2016

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

COMMUNITY EVENT PREDICTION IN EVOLVING SOCIAL NETWORKS

Ph.D. THESIS

Nagehan İLHAN
(504082501)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

NOVEMBER 2016

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DİNAMİK SOSYAL AĞLARDA TOPLULUK OLAY ÖNGÖRÜSÜ

DOKTORA TEZİ

Nagehan İLHAN
(504082501)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

KASIM 2016

Nagehan İLHAN, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 504082501 successfully defended the thesis entitled "COMMUNITY EVENT PREDICTION IN EVOLVING SOCIAL NETWORKS", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Şule Gündüz Öğüdücü** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Şima Etaner-Uyar** Istanbul Technical University	.....
	Prof. Dr. Yücel Saygın Sabancı University	.....
	Asst. Prof. Dr. Yusuf Yaslan Istanbul Technical University	.....
	Assoc. Prof. Dr Songül Albayrak Yıldız Technical University	.....

Date of Submission : **30 September 2016**
Date of Defense : **03 November 2016**

To my mother,

FOREWORD

First of all, I would like to express my sincere appreciation to my advisor, Assoc. Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ for her helpful advices and encouragement for me to proceed through my PhD study and complete my PhD thesis. Not only she impressed me as a researcher but also she was a role model and I have been very lucky to have such excellent advisor. There are no words to express my gratitude to her. I would like to give very special thanks to my committee members, Assoc. Prof. Dr. Şima ETANER-UYAR and Prof. Dr. Yücel SAYGIN, for their time, interest and helpful recommendations.

I also would like to thank to all my professors at Istanbul Technical University Faculty of Computer and Informatics Engineering, especially Prof. Dr. Eşref ADALI for his academic support and guidance throughout the study. I also thank to colleagues and research assistants of the Computer Engineering Department at ITU, in particular my roommates: Dr. Berk CANBERK, Figen ÖZTÜRK, Selda UYANIK, Dr. Tahir SANDIKKAYA and Atakan ARAL, our valuable faculty members: Dr. Yusuf YASLAN, Dr. Ahmet Cüneyd TANTUĞ and Dr. Tolga OVATMAN for their guidance, friendship and help during my PhD. I wish to express a very special thank to Zuhal YILMAZER for her kindness and helpfulness.

I am indebted to my golden girls, Onan GÜREN, Tuğba AĞAÇAYAK, Gönül ULUDAĞ, Güldem KARTAL ŞİRELİ and Aslıhan VURUŞKAN for their warm friendship and fun-filled environment. I am also thankful to Birkan TUNÇ for his friendship and guidance.

Lastly, I would like to thank my whole family for all their endless support. I flow endless gratitude to my parents for all of the sacrifices that they have made on my behalf and my sisters whose love and support carried me all the way through in my life. I am very sure that if my mother had seen that I have finished, she would have been very happy and proud of me. I dedicate the thesis to her..

November 2016 Nagehan İLHAN

TABLE OF CONTENTS

Page

ABBREVIATIONS

AIC : Akaike Information Criteria
ANN : Artificial Neural Network
ARIMA : Autoregressive Integrated Moving Average
BR : Binary Relevance
CPM : Clique Percolation Method
CAE : Correlation Attribute Evaluation
CFS : Correlation-based Feature Selection
ETS : Exponential Smoothing
FIEP : Feature Identification for Event Prediction
FSE : Feature Subset Extraction
GED : Group Evolution Discovery
GR : Gain Ratio
GRAE : Gain Ratio Attribute Evaluation
hepPh : High Energy Physics Phenomenology
hepTh : High Energy Physics Theory
IGAE : Information Gain Attribute Evaluation
KNN : K-Nearest Neighbor
LEV : Leading Eigen Vector
LPM : Label Propagation Method
MAPE : Mean Absolute Percentage Error
MCC : Multi-Class Classifier
ORAE : OneR Attribute Evaluation
RFAE : Relief-F Attribute Evaluation
SGCI : Stable Group Changes Identification
SNA : Social Network Analysis
SUAE : Symmetrical Uncertainty Attribute Evaluation
WSE : Wrapper Subset Evaluation

LIST OF TABLES

Page

LIST OF FIGURES

Page

COMMUNITY EVENT PREDICTION IN EVOLVING SOCIAL NETWORKS

SUMMARY

The last decade has witnessed the dramatically quick and explosive growth of information available over the Web and Internet. Social networks have become very popular due to increasing proliferation of Internet enabled devices such as personal computers, mobile phones and other recent hardware innovations. Social networking sites enable users to share ideas, post updates and comments, to follow breaking news while keeping up with friends or colleagues. Rapid growth of online social networks has led to a tremendous explosion of network centric data in a wide variety of scenarios. Data from many types of social networks is a graph, where nodes represent individuals and edges represent the relationship and interactions among individuals. In such social networks, communities are constituted as a result of establishing relationships and interactions with each other. Although no single definition has been agreed upon, the most common definition of community is: a collection of nodes who are more densely connected to each other than with the rest of the network. Discovery of communities consists in characterizing the structure of a network at the mesoscopic level, i.e. neither the point of view of a single node or edge (microscopic level) nor the whole graph structure (macroscopic level), but rather an intermediary structure, namely community. Inherently, as time passes, community members interact with each other, but they also interact with others outside the community, resulting in a dynamic or temporal behavior. As a result, communities may stay stable just over a period of time, display a periodic pattern, change member composition abruptly, and perform many other evolutionary events. This phenomenon attracts researchers to study how connections between individuals are established and how they evolve over time.

Capturing the community dynamics and predicting their evolution has been an important subject in Social Network Analysis (SNA). Quantifying community evolution is crucial to identify major changes in the internal organization of the network. Analyzing the evolution of communities over time provides insights in many application domains such as sociology, criminology, advertising and marketing, information diffusion, recommender systems, etc. Latent trends of members' behavior and adoption can be exposed by understanding which communities are growing, shrinking or undergo other events. For instance in marketing strategy, performing the marketing actions on communities and tracking the results of those actions over time is necessary to extract knowledge that can be used to support the redesign of marketing promotions. The user comments and friendship ties within a social networking site can be used to follow emergence and development of new ideas and political views.

Common approaches to tracking the evolution of communities have devised on an event framework that defines a specific behavior of a community like growth, merge and disappear. Many of them propose a two-step event based model to explore

community evolution which firstly detect communities on each snapshot graph and secondly construct relationships between partitions at successive snapshots defining critical events and predicting events by use of community features as attributes. These approaches need community extraction and computing the whole community features of each snapshot including the snapshot which its future to be predicted. Moreover, recent studies predict community evolution by considering the whole historical information. However, such an approach may fail to provide accurate results on current evolution since it considers all the historical data and treat all nodes and links equally, even if they only appear at the early stages of the network life.

In the thesis, evolutionary dynamics of the communities has been studied and an event based approach proposed to predict future events of the communities. The dynamic network is modeled by a series of static snapshots and each snapshot corresponds to a time interval constituted of the interactions during/up to that specific interval. Communities in each snapshot are mined and communities' structural and temporal features are computed. The extracted features cover many properties of both the internal link structure and the external interaction of the community with the rest of the network. A similarity metric is calculated for each pair of communities at successive snapshots and significant events of the communities are identified by applying our proposed event detection algorithm. Then, event prediction modeled as a classification problem where identified events are used as class labels for the classifiers and the structural features as input parameters. Detailed experimental results have proved that the proposed event prediction model can accurately estimate community events.

By utilizing the underlying event based framework, the thesis suggests frameworks to two substantial problems in event prediction of temporal communities. The first one is predicting community events by employing time series analysis models. A time series model predicts how particular community features will change in the following time period thus directly predicts community features at the next time step thus it avoids discovering communities from scratch. 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 and sliding window techniques. Unlike the landmark windows which take into consideration all the historical data, sliding windows focus on the most recent state of the dynamic network thus uncover the most recent changes occurring in the network. Distinct time window intervals are examined in constituting and analyzing time series. Experimental results on two real datasets show that the proposed framework forecasts community feature values with a reasonable error rate and predicted events highly overlap with the actual event labels. Moreover, the effect of window size on the forecast error and event prediction is uncovered.

The second one is identification of community features that perform successful prediction results. A novel framework proposed to examine various structural features of the network and detects the most prominent subset of community features in order to predict the future direction of community evolution without computing the entire feature set. The framework extracts the network's structural properties and use it to determine the subset of community features that leads to accurate community event prediction. Unlike traditional approaches that harvest a large number of features at each time point, the proposed framework suggests the most predictive community features by exploiting the network's topology to effectively determine whether a

community will remain stable or undergo certain events such as shrink, merge or split. Experiments conducted on four real datasets verified the effectiveness of the proposed framework. The experiments indicated that framework produces almost the same prediction results as those produced using the entire feature set, such that there is no statistical difference. Furthermore, the results for the running time and speedup of framework over the use of all features have also presented. Due to the lower number of features that should be calculated, there is a corresponding reduction in computational time and cost.

DİNAMİK SOSYAL AĞLARDA TOPLULUK OLAY ÖNGÖRÜSÜ

ÖZET

Geçtiğimiz son on yıl internet ve web üzerinden elde edilen bilginin hızla ve çarpıcı bir biçimde artmasına tanıklık etmiştir. Sosyal ağlar kişisel bilgisayarlar, cep telefonları ve diğer donanımsal yenilikler gibi internet erişimli cihazların yaygınlaşmasıyla birlikte oldukça popüler hale gelmiştir. Sosyal ağ siteleri kullanıcılarına fikirlerini paylaşma, güncel durum ve yorumları yayınlama, yeni haberleri takip edebilme, arkadaşlar ve meslektaşlar ile iletişimde kalabilme gibi olanaklar sağlamaktadır. Çevrimiçi sosyal ağların hızlı gelişimi çok çeşitli senaryolarda ağ merkezli verilerin muazzam bir biçimde artmasına yol açmıştır. Birçok sosyal ağ çeşidinde veri, düğümlerin bireyleri, ayrıtların ilişki ve etkileşimleri temsil ettiği çizgelerden oluşur. Ağ yapısı içerisindeki bir düğüm kümesi, dışarıya olan bağlantı sayısına kıyasla kendi içinde daha fazla sayıda bağ içeriyor ise bu düğüm kümesi bir topluluk olarak nitelendirilebilir. Bu gibi sosyal ağlarda topluluklar, ağ elemanlarının birbirleriyle ilişki ve etkileşim kurması neticesinde oluşmaktadır. Topluluğun en yaygın tanımı şudur: topluluk içeride yoğun olarak birbirine bağlı ancak dışarıyla daha az yoğunlukta bağlantısı olan düğümler topluluğudur. Toplulukların belirlenmesi, düğüm veya ayrıt bazında bir yaklaşım sergilenen mikroskopik düzey ve bütün çizge yapısını ele alan makroskopik düzeyden farklı olarak, daha ara yapılar olan topluluklar kullanılarak, ağ yapısının mezoskopik düzeyde tanımlanmasıdır. Topluluk üyeleri zaman içinde birbirleriyle ve topluluk dışındaki ağ üyeleriyle etkileşim kurarak dinamik veya zamansal bir davranış sergilerler. Sonuç olarak, topluluklar belirli bir zaman diliminde kararlı bir biçimde durabilir, periyodik örüntü sergileyebilir, üye bileşiminde ansızın değişiklikler olabilir veya diğer başka dinamik davranışlar gösterebilirler. Bu nedenle toplulukların ve ağın zamansal değişimlerinin irdelenmesi önem arz etmektedir. Bu fenomen araştırmacıları bireyler arasındaki bağlantıların nasıl kurulduğu ve zamanla nasıl değiştiğini araştırma konusuna yöneltmiştir.

Ağ yapılarının içerisinde yer alan bireyler/varlıklar arasındaki ilişkilerin çeşitli bilimsel metotlar aracılığı ile detaylı olarak incelenmesi sonucu elde edilen verilerden anlamlı sonuçlar türetilmesi Sosyal Ağ Analizi olarak tanımlanmaktadır. Bu bağlamda, topluluk dinamiklerinin gözlemlenmesi ve değişiminin öngörülmesi Sosyal Ağ Analizi'nin en önemli konularından biridir. Toplulukların değişimlerinin belirlenmesi, ağın içsel organizasyonundaki temek değişiklikleri tanımlamak açısından elzemdir. Toplulukların zaman içindeki değişiminin analiz edilmesi sosyoloji, kriminoloji, reklamcılık ve pazarlama, bilgi difüzyonu, öneri sistemleri gibi pek çok uygulama alanında genişçe yer almaktadır. Üyelerin gizli yönelimleri ve beğenileri, hangi toplulukların büyüyeceği veya küçüleceği gibi olayların anlaşılmasıyla açığa çıkarılabilir. Örnek olarak, topluluklara pazarlama faaliyeti yapmak ve bu faaliyetin zaman içinde sonuçlarını izlemek verilebilir. Buradan yapılan çıkarımlar daha sonra pazarlama teşviklerinin yeniden düzenlenmesinde kullanılabilir. Bir sosyal ağ

sitesindeki kullanıcı yorumları ve arkadaşlık ilişkileri yeni fikir ve politik görüşlerin oluşumunu ve gelişimini izlemekte kullanılabilir.

Toplulukların gelişimini takip eden yöntemler yaygın olarak topluluğun büyüme, birleşme veya yok olma gibi olaylarını tanımlayan çerçeveler üzerine kurulmuştur. Bunların birçoğu, toplulukların evrimini incelemek üzere iki adımlı olay eksenli modeller önermiştir. Bu modeller birincil olarak her bir zaman dilimine ait çizgelerin topluluklarını belirleme ve daha sonra ardışık zaman dilimlerindeki bölümlemeler arasındaki ilişkiyi kritik olaylar olarak tanımlamakta ve toplulukların yapısal özelliklerini öznitelik olarak kullanarak olayları tahmin etmektedir. Netice olarak bu yöntemler, bir sonraki adımdaki olayları bilinmeyen zaman dilimi de dahil olmak üzere tüm zaman dilimlerine ait çizgelerdeki toplulukların belirlenmesi ve bütün topluluksal özelliklerinin hesaplanmasını gerektirmektedir. Buna ek olarak, yakın zamandaki çalışmalar toplulukların evrimini çizgelere ait bütün tarihsel bilgileri kullanarak tahmin etmektedir. Ancak bu tarz bir yaklaşım, bütün geçmiş veriyi kullanması ve sadece ağ oluşumunun ilk aşamalarında var olmuş olan düğümleri diğerlerinden ayırt etmeksizin bütün düğüm ve ayrıtları eşit olarak göz önüne almasından mütevellit, güncel değişimle ilgili doğru sonuçlar sağlamakta başarısız olabilir.

Bu tezde, toplulukların evrimsel dinamikleri irdelenmiş ve toplulukların gelecekteki olaylarını tahmin etmek amacıyla olay eksenli bir yaklaşım önerilmiştir. Dinamik ağ bir dizi statik zaman adımlı çizgeler olarak modellenmiş ve her bir zaman adımlı çizge belirlenen zaman aralığında veya o zamana kadar olan etkileşimlerden oluşturulmuştur. Her bir zaman adımında topluluklar belirlenmiş ve toplulukların yapısal ve zamansal nitelikleri hesaplanmıştır. Ele alınan özellikler toplulukların içsel ayrıt yapıları ve toplulukların ağın geri kalanıyla olan dışsal etkileşimlerini de içerecek şekilde pek çok niteliği kapsamaktadır. Ardışık zaman adımlarındaki her bir topluluk çifti için bir benzerlik ölçütü hesaplanmış ve önerdiğimiz olay belirleme algoritması uygulanarak anlamlı olaylar saptanmıştır. Daha sonra olay öngörüsü, tanımlanmış olayların sınıf etiketleri, yapısal ve zamansal topluluk niteliklerinin giriş parametresi olarak kullanıldığı bir sınıflandırma problemi olarak modellenmiştir. Detaylandırılmış deneysel sonuçlar önerilen olay öngörü modelinin topluluk olaylarını doğru olarak yakınsadığını ispatlamıştır.

Tez kapsamında önerilen olay eksenli çerçeve kullanılarak, zamansal topluluklarda olayların öngörülmesinde karşılaşılan iki temel soruna da çözüm getirilmiştir. Tezin birinci katkısı; topluluk olaylarının zaman serisi analizi modelleri kullanılarak öngörülmesidir. Zaman serisi modelleri topluluksal niteliklerin bir sonraki zaman adımında nasıl değişeceğini ve değerinin ne olacağını tahmin ederek, toplulukların sıfırdan belirlenmesini önlemiş olur. Bir sonraki adımda olayları öngörülecek olan zaman dilimine ait toplulukların her biri için, biriken ve kayan pencereleme tekniklerine göre geçmişten günümüze topluluksal değerlerden müteşekkil bir zaman serisi oluşturulur. Bütün geçmiş veriyi işleyen biriken pencereleme tekniğinin aksine, kayan pencereleme tekniği dinamik ağın güncel durumu üzerine yoğunlaşır ve dolayısıyla ağdaki en son değişiklikleri meydana çıkarır. Zaman serilerinin oluşumu ve analizinde farklı pencere aralıkları test edilmiştir. İki adet gerçek veri üzerinde yapılan deneysel sonuçlar önerilen çerçevenin toplulukların nitelik değerlerinin makul bir hata oranı ile tahmin edildiğini ve öngörülen olayların gerçek olay etiketleriyle

yüksek oranda örtüştüğünü göstermektedir. Buna ek olarak, pencere büyüklüğünün tahmin hatası ve olay öngürüsü üzerindeki etkisi irdelenmiştir.

İkinci olarak, olayların öngörülmesinde başarılı sonuçlar üreten topluluksal nitelikler tespit edilmiştir. Ağın çeşitli yapısal niteliklerini hesaplayan ve topluluk gelişiminin gelecekteki doğrultusunu tüm topluluksal nitelikleri hesaplamadan, önde gelen topluluk nitelik alt kümesini tespit ederek öngören yeni bir çerçeve önerilmiştir. Önerilen çerçeve ağın yapısal niteliklerini kullanarak topluluk olay tahmininde doğru sonuç üretmeyi sağlayan topluluk nitelikleri alt kümesi belirlemektedir. Her bir zaman noktasında çok sayıda topluluksal niteliği hesaplayan yöntemlerin aksine, önerilen çerçeve ağ topolojisinden faydalanarak topluluğun gelecekte başına gelecek olayı en etkin biçimde öngören en kestirimci topluluksal nitelikleri bulmaktadır. Dört farklı gerçek veri seti üzerinde yapılan deneyler önerilen çerçevenin etkinliği doğrulamıştır. Yapılan deneyler, önerilen çerçevenin bütün nitelik kümesi kullanılarak üretilen öngörü sonuçlarıyla hemen hemen aynı sonuçlar ürettiğini, sonuçlar arasında istatistiksel bir farklılık bulunmadığını göstermiştir. Önerilen çerçeve ve bütün topluluk niteliklerinin kullanımı çalışma süresi ölçülerek karşılaştırılmış ve çerçevenin hızlandırma oranı sunulmuştur. Sonuçlar, önerilen çerçeve kapsamında daha az nitelik hesaplanmasından dolayı, aynı nispette hesaplama zaman ve maliyetinde düşüş olduğunu kanıtlamıştır.

1. INTRODUCTION

Complex systems can be described as networks consisting of a set of items which is called vertices or nodes, with connections between them, called edges. There are many types of complex networked systems such as social networks, technological networks, climate networks, protein-protein interaction networks, etc. With the current popularity of social networks, network analysis and modelling of such networks become more popular. In this thesis, studies are performed focusing on social networks. Social networks are made up of actors called nodes that are connected by various social familiarities or relationships represented by edges. A social network dynamically changes since the social ties between network actors change over time. The rapid advent in social networking systems has given rise to a growing need for Social Network Analysis (SNA) in order to investigate the relationships between network actors while being able to follow their evolution. In SNA, the main interest is to infer the structural characteristics of networks. Hence the connections between actors are key elements of the analysis that facilitates the mining of the important behavioral patterns among the actors.

One of the salient feature of the social networks is represented by their mesoscopic structure, characterized by the presence of groups of nodes, called communities, with a high density of connections between nodes of the same community and comparatively sparse connections between nodes of different communities [1] [2]. Exploring network communities is important to reveal the network organization at a coarse level and uncover relationships between the nodes which are not apparent by inspecting the graph as a whole. In 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. 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.

Recently, a great deal of work has been devoted on analyzing dynamic communities and their temporal evolution [3]. A common strategy involves considering a dynamic network divided into series of individual time step graphs, representing successive snapshots of the graph taken at regular intervals and study the structural characteristics of the static networks within each interval. Hopcroft *et al.* proposed a method to track stable clusters over time which utilizes "natural community" that have high stability against to minor perturbations of the graph [4]. Berger-Wolf *et al.* proposed a mathematical and computational framework that enables analysis of dynamic social networks and explicitly makes use of information about the time that social interactions occur [5]. The same team in their later study formulated the community membership detection as a graph coloring problem, using greedy matching heuristic in order to assign individuals to communities in any given time interval [6].

In the course of network evolution, different events may occur such that 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 [7]. 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. Many approaches to characterize the evolution of communities have focused on identifying critical events that a community can encounter, then investigating the occurrences of these events within the network. Palla *et al.* identify six basic events such as birth, growth, and merging by applying Clique Percolation Method (CPM) [8]. Asur *et al.* define critical events between detected communities at two consecutive snapshots which are implemented in the form of bit operations [9]. However, these events do not cover all of the transitions that may occur for a particular community. 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 [10]. Takaffoli *et al.* proposed an event-based framework incorporating substantial features related to a community such as its structure, history and influential members. Their framework allows to track the changes of communities, not only between two consecutive snapshots but also encompassing multiple snapshots. Gliwa *et al.* proposed a method, namely SCGI, for modeling group evolution and event prediction by utilizing leadership, density, cohesion and group size measures [11]. In our paper [12], a model is proposed 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.

In spite of the tremendous amount of work that has been done so far, there are still problems. In the large part of community evolution studies, community features are utilized 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 by utilizing the community features. However, the proposed approaches require the community extraction and computing the community features relevant to the time point to be predicted. Besides, in these approaches, communities are extracted by applying an appropriate community detection algorithm to each snapshot of the network that has been accumulated over the time span (aka landmark windows). Thus, these approaches evaluate the whole historical data in the analysis and designate equal weight to nodes and edges even if they were not active for a while. However, such an approach may fail to provide accurate results on current evolution. The first hypothesis of the thesis is as follows: community features of the next time step can be estimated using time series analysis models applied on time series which constituted using a specified length of snapshot history. Another thing is; most of the existing work on community evolution focuses on a model which predicts a set of events by computing a range of features that span different categories. However, extracting a wide range of features is computationally expensive, especially when working with large datasets. In such cases, it is crucial to discard redundant and ineffective features. Even the studies which incorporate feature selection stage do not remedy the issue of feature calculation cost. The second hypothesis is that: exploring the prominent subset of community

features will reduce the feature calculation cost and these features can be identified by exploiting topological network properties. Therefore, a model should be suggested to identify a minimal number of community features to effectively determine community events.

1.1 Contributions of the Thesis

The main purpose of the thesis is tracking the evolutionary dynamics of the communities in the social network and predicting the future event of the communities over time. Experiments on different data sets prove that a high rate of community evolution prediction has been achieved. On this basis, we also seek solutions to the related problems. In the basis of the proposed approaches, the process proceeds as follows: the evolving network is handled by series of static snapshots where each snapshot corresponds to a particular point in time. In the first step, a community detection algorithm is applied to discover communities. In the second step, the structural features are extracted by measuring on a large scale of the properties of the communities. The extracted features cover many properties of both the internal link structure and the external interaction of the community with the rest of the network. The third step involves matching communities found at consecutive time steps in the individual snapshot graphs and identification of significant events of the communities, such as survive, growth, merge, split and dissolve. The fourth step is the event prediction step in which identified events are used as class labels for the classifiers with the structural features found in the second stage as input parameters. The thesis can be handle into two phases: 1) Community evolution prediction based on time series modeling and 2) Feature identification for predicting community evolution.

1) Community evolution prediction based on time series modeling: In the first phase, an approach to accurately predict the next event of a community with employing various time series models namely the Autoregressive Integrated Moving Average (ARIMA), Artificial Neural Networks (ANN) and Exponential Smoothing (ETS) has been proposed. Our first contribution is effectively predicting community events through community feature forecasting. The community feature values of the next snapshot are directly forecasted thus demand for the community extraction of the snapshot to be predicted is removed. Our second contribution is examining the

dynamics of the community structure in the networks make use of two different time window models: a landmark and a sliding window. Landmark window keeps the data from the beginning to the present time where sliding window is focusing on the specified amount of the recent past thus allowing to capture current events. We then investigate the influence of the windowing approaches and various window lengths on the framework results.

2) Feature identification for predicting community evolution: In the second phase, the problem of feature calculation cost in the study of community event prediction has been addressed. A novel framework named Feature Identification for Event Prediction (FIEP) to identify a proper subset of features for a given network that achieves good prediction results without the need for calculating all features at the beginning. The suggested framework utilizes various structural network measures including clustering coefficient, average path length, embeddedness and betweenness in order to determine the accurate subset of features. The contribution of this stage is threefold. First, the proposed generic methodology for predicting the community evolution facilitates the identification of the predictive community features based on the structure of networks. The community event prediction is then modeled as a classification problem. Second, our methodology is capable of determining a useful subset of community features at the first observation moment of the network without observing the dynamic behavior of the network at different time periods. Third, we have empirically tested different factors related to the network structure and community features that may contribute positively to the community event prediction performance.

The rest of the thesis chapters are organized as follows:

- In Chapter 2, the related literature review has been given.
- Chapter 3 includes the brief descriptions about the community detection algorithms, feature selection and time series analysis methods that are used throughout the thesis.
- In Chapter 4, community evolution prediction based on time series modeling approach is presented and experimental results are given.

- Chapter 5 includes the feature identification for predicting community evolution stage and the details of proposed framework. Related experimental results are also provided.
- Chapter 6 concludes the thesis by discussing the outcomes and the possible future directions for the work.

2. RELATED WORK

2.1 Tracking Community Evolution

Recently, many researchers have been interested in mining the temporal evolution of social networks. [8, 9, 13–19]. A common way to study temporal network behavior is taking static snapshots and analyzing the structural characteristics of the static networks within each snapshot.

Several studies have been carried out on constructing an event prediction framework that characterizes the evolution of communities in dynamic networks. Palla *et al.* proposed an extension of the Clique Percolation Method (CPM) [20] to identify events such as birth, growth and merging in the evolution of dynamic graphs [8]. This extension involved applying CPM on a graph formed by the communities discovered at pairs of consecutive snapshots. The resulting clique based communities were subsequently matched to communities and events pertaining to the communities specified. Asur *et al.* define critical events between detected communities at two consecutive snapshots which are implemented in the form of bit operations [9]. However, these events do not cover all of the transitions that may occur for a particular community. Wang proposed an intuitive method to compare two communities of the consecutive timestamps with rules based on tracking specific core nodes that are more representative of their community than others [21]. Greene *et al.* allowed for tracking of similar communities in different snapshots [22]. They proposed a model for tracking the evolution of communities over time in a dynamic network, where each community is characterised by a series of significant evolutionary events. Their model introduces an effective community-matching strategy for efficiently identifying and tracking dynamic communities in multiple snapshots of a dynamic network. The authors in [10], presented an approach to discover all possible types of community-based anomalies in evolutionary networks characterized by overlapping communities. Tajeuna *et al.* proposed a novel approach for modeling and detecting the

evolution of communities. Their model comprises a new similarity measure, named mutual transition, for tracking the communities and rules for capturing significant transition events a community can undergo [23]. Leskovec *et al.* studied the patterns of graph evolution based on the various properties of the large social networks such as the degree distribution and the small-world phenomena [14]. They also propose Forest Fire model to produce networks satisfying the discovered patterns. Ahn *et al.* analyzed different behavior scaling in degree distribution on online social networks, extracting the main characteristics of online social networks and performing an analysis of the evolution of Cyworld network [24]. However, in these studies the influence of structural properties is examined at individual level and the prediction is lacking. A prediction which discards structural properties of communities may be insufficient when predicting several different events.

2.2 Community Event Prediction

Some approaches focus on the event prediction problem to determine the future behavior of a community. Goldberg *et al.* developed an algorithmic framework for studying the evolution of communities by proposing axioms which imply that an evolution is at most as strong as its weakest link [25]. They also studied the predictability of evolution, in particular lifespan, by identifying a consistent set of structural features including density, intersection, size, growth and core of the early stages of a community that indicate whether a community is going to be short-lived or not. They found that density, intersection, and core size are quite significant and have strong positive correlation with lifespan. In their subsequent study, they proposed a two-step process for the identification of evolution within a network. However, both studies are only useful for predicting the lifetime of the community. Kairam *et al.* proposed a predictive model and investigated the relationsip between a group's network features and its future growth and longevity using online social network community data [26]. Patil *et al.* build a model to predict if a group is going to remain stable or is likely to shrink over a period of time [27]. They successfully predicted group stability with high accuracy using a range of features that describe the group composition, activities within the group and structural aspects of a group. Diakidis *et al.* presented a study to predict the evolution of communities by focusing on the continuation,

growth, shrink and dissolution events computing the structural, content and contextual community features of Twitter [28]. However, their method is unable to identify the merge and split events of a community which are the essential events that a community may encounter in its life cycle. Bródka *et al.* proposed GED (Group Evolution Discovery) method to discover group evolution and the sequences of group sizes, while events between time steps were extracted from the GED results [29]. The sequence consists only of several preceding group sizes and events as an input for the classifier. Similar authors then proposed a new method for future event prediction based on stable group changes identification algorithm (SGCI) [11]. They used leadership, density, cohesion and group size measures to describe the group profile. They show that using many measures to describe the group profile, and in consequence as a classifier input, can improve predictions. The same group, in their later study, compared SCGI and GED with different lengths of evolution chains by extending the variety of community features [30].

2.2.1 Time series analysis

Implementing time series analysis on social networks also a topic of research. The authors in [31] 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. In [32], an approach presented 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. Time series analysis has also been applied in tweet analysis. The authors in [33] analyzed several surveys on consumer confidence and political opinion of textual sentiment in microblog messages through time and they correlate to sentiment word frequencies in contemporaneous Twitter messages. However, none of them are concerning the community features and events. In our previous work [34], we proposed a model which avoids applying community detection algorithm and calculating community features for the relevant snapshot where its evolution will be predicted. The model utilized time series analysis model ARIMA to forecast precise community feature values, thereby classify the communities to the related events. 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.

2.2.2 Feature selection

Feature selection has been an active and widely recognized method for improving data quality in the field of machine learning and data mining communities [35]. Feature selection methods involve selecting the optimal subset of features from the original set of features that show the best performance in terms of well-defined criteria, e.g. classification accuracy [36–38]. Through feature selection, the cost of learning, the amount of data needed to achieve learning and overall execution time is reduced while classification accuracy is possibly improved. In general, feature selection can be classified into three main evaluation models: filter model [39], wrapper model [40] and the embedded model [35]. Filter models extract features from the data without any learning involved Gain Ratio Attribute Evaluation (GRAE) [41], Information Gain Attribute Evaluation (IGAE) [42], OneR Attribute Evaluation (ORAE) [43], Relief-F Attribute Evaluation (RFAE) [44], Symmetrical Uncertainty Attribute Evaluation (SUAE) [45], Correlation-based Feature Selection (CFS) [46]. The wrapper models use learning techniques to evaluate which features are useful Wrapper Subset Evaluation (WSE) [47], Genetic Algorithms [48]. The embedded models combine the feature selection step and the classifier construction (Random Forest [49]). We also note that there are several works which study feature selection in community evolution [16, 30, 50]. These methods select the features to reduce the overfitting effects thereby improving prediction accuracy, but do not eliminate calculation cost of the whole feature set.

3. BACKGROUND

In this chapter, community detection algorithms, feature selection methods and time series analysis models that used within the scope of the thesis are briefly described.

3.1 Community Detection Algorithms

Identifying meaningful community structure in social networks is a hard problem and a number of methods to address this problem have been proposed. In this section very partial descriptions only about the methods which are implemented in the study has been provided.

3.1.1 Fast Greedy

Fast Greedy is a greedy optimization algorithm and tries to optimize a quality function called modularity in a greedy manner [51]. Initially, each node in its own community, and then in every step two communities are merged iteratively in order to gain largest increase in the current value of modularity such that each merge is locally optimal. The algorithm continues when it is not possible to increase the modularity any more. This results a grouping as well as a dendrogram. The hierarchical merging tree is cut at the point where maximum modularity is achieved. The method is fast and generally tried as a first approximation because it has no parameters to tune. However, it is known to suffer from a resolution limit and will always be merged with neighboring communities.

3.1.2 Leading Eigen Vector (LEV)

Leading Eigenvector is a top-down hierarchical graph partitioning approach using a so-called modularity matrix [52]. The method finds communities by calculating the leading non-negative eigenvector of the modularity matrix. In each step, the graph is split into two parts in a way that the separation itself yields a significant increase in the modularity. The split is determined by calculating the eigenvector of the modularity

matrix for the largest positive eigenvalue and then separation occurs on vertices into two community based on the sign of the corresponding element in the eigenvector. There also exist a stopping criteria which prevents tightly connected groups to be split further. If all elements in the eigenvector have the same sign means that the network can not be split anymore.

3.1.3 Infomap

Infomap is a two-level method based on Huffmann coding. First level is to distinguish communities in the network and second to distinguish nodes in a community [53]. It tries to build a community structure which provides the shortest description length for a random walk on the graph. The goal is to find the community structure that minimizes the expected length of the description.

3.1.4 Label Propagation Method (LPM):

Label Propagation Method is based on the simple rule that at each iteration a given node takes the most frequent label in its neighborhood. The starting configuration is chosen such that every node is given a different label and the procedure is iterated until convergence [54].

3.1.5 Louvain:

Louvain Method is a hierarchical greedy algorithm which is composed of two phase [55]. Initially, each node is assigned to a community on its own. In the first stage, 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 stage, each community is considered as a node on its own. Then, the algorithm starts the phase 1 and so on.

3.2 Feature Selection Methods

In the thesis, eight common feature subset selection methods proposed in the literature including both filter and wrapper models are considered. These methods are: Information Gain [42], CFS [46], Correlation Attribute Evaluation [56], Gain Ratio

[41], OneR, Relief-F [44], Symmetrical Uncertainty [45],Wrapper Subset Evaluation [47].

3.2.1 Information Gain Attribute Evaluation (IGAE):

Information Gain measures information obtained for class prediction by evaluating presence or absence of a feature using entropy [42]. Information Gain is calculated by the feature's contribution on decreasing overall entropy. Let D be set consisting of d data samples with k distinct classes, the expected information needed to classify a given instance is given by:

$$I(D) = -\sum_{i=1}^{k} p_i log2(p_i) \tag{3.1}$$

where p_i is the probability that an arbitrary data sample belongs to class C_i estimated as $\mid C_iD \mid / \mid D \mid$. If we want to classify the instance in D on some attribute A, D will split into w partitions set $\{D_1, D_2, ..., D_w\}$. The entropy, or expected information based on the partitioning into subset by A, s given by:

$$E(A) = -\sum_{j=1}^{w} \frac{\mid D_j \mid}{\mid D \mid} xI(D_j) \tag{3.2}$$

where $\mid D_j \mid$ is the weight of the j^{th} partition and $I(D_j)$ ia the entropy of partition D_j. Then, Information Gain (IG) by partitioning on A is:

$$IG(A) = I(D) - I_A(D) \tag{3.3}$$

3.2.2 Correlation-Based Feature Selection (CFS):

CFS evaluates the subsets of features by considering degree of redundancy among them. The methods aims to find subsets of features that are individually highly correlated with the class while having low inter-correlation. Equation of CFS is given in Equation 3.4.

$$\rho_{zc} = \frac{s\overline{\rho_{zi}}}{\sqrt{s + (s-1)\overline{\rho_{ii}}}} \tag{3.4}$$

where ρ_{zc} is the correlation between the summed feature subsets and the class variable, s is the number of subset features, ρ_{zi} is the average of the correlations between the subset features an the class variable, and ρ_{ii} is the average inter-correlation between subset features [46].

3.2.3 Correlation Attribute Evaluation (CAE):

Correlation attribute evaluation measures the worth of an attribute by calculating the Pearson's correlation between it and the class [56]. For two quantitative attribute X and Y, Pearson's correlation coefficient is defined as:

$$r_{XY} = \frac{\sum(x_i - \bar{x}y_i - \bar{y})}{(n-1)S_X S_Y} \tag{3.5}$$

where $\bar{x}$ and $\bar{y}$ are the sample means for X and Y respectively, S_X and S_Y are the sample standard deviations for X and Y and n is the size of the instance used to compute the correlation. The correlation r_{XY} is comprises between -1 and 1. A value of 1 means perfect positive correlation and -1 in the other direction.

3.2.4 Gain Ratio Attribute Evaluation (GRAE):

The information gain measure prefers to select attributes having a large number of values thus it is biased towards tests with many outcomes. C4.5 [41], a successor of the basic decision tree induction algorithm ID3 [57], uses an extension to information gain known as Gain Ratio (GR), which attempts to overcome the bias by introducing an extra term named Split Info (SI) taking into account how the feature splits the data. The split info corresponds to the potential information obtained by partitioning the training data set D into w partitions, resulting to w outcomes on attribute A:

$$SI_A(D) = -\sum_{i=1}^{w}(\mid D_i \mid / \mid D \mid)log_2(\mid D_i \mid / \mid D \mid) \tag{3.6}$$

High SI means partitions have equal size and low SI means few partitions contains most of the tuples. The Gain Ratio (GR) is defined as:

$$GR(A) = IG(A)/SI_A(D) \tag{3.7}$$

The attribute with maximum gain ratio is selected as the splitting attribute.

3.2.5 OneR Attribute Evaluation (ORAE):

OneR approach evaluates each attribute individually by using the 1R classifier. For each attribute and for each value of the attribute, the error produced if only that attribute will be used to classify the corresponding dataset. The attributes ranked based on the error rate obtained and desired number of attributes selected with lowest error rate.

3.2.6 Relief-F Attribute Evaluation (RFAE):

Relief-F method randomly selects feature instances, computes their nearest neighbors to find nearest miss and nearest hit, calculates the weight of a feature and adjusts a feature weighing vector to give more weight to features that discriminate the instance from neighbors of different classes [44].

3.2.7 Symmetrical Uncertainty Attribute Evaluation (SUAE):

Symmetrical Uncertainty method evaluates attributes individually by measuring symmetrical uncertainty with respect to the class [45]. It compensates for the inherent bias of IG by dividing it by the sum of the entropies of X and Y. Symmetrical Uncertainty (SU) is given by:

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{E(X)E(Y)}\right] \tag{3.8}$$

where $IG(X \mid Y)$ is the information gain of independent attribute X and the class attribute Y. $H(X)$ is the entropy of feature X and $H(Y)$ is the entropy of feature Y. SU normalizes the value to the range $[0,1]$. $SU = 0$ indicates that X and Y are uncorrelated and $SU = 1$ indicates that the knowledge of one feature completely predicts. SU is biased toward features with fewer values like GR.

3.2.8 Wrapper Subset Evaluation (WSE):

Wrapper methods evaluate subsets by running a specific classifier on the training data, using only the attributes of the subset [47]. In the wrapper method, the feature subset selection is done by induction algorithm in order to find a good subset. The space of feature subsets searched and the estimated accuracy of a single learning algorithm is calculated for each feature that can be added to or removed from the feature subset. The feature space can be searched with various strategies, e.g., greedy stepwise, best first, random search, etc. In the wrapper approach, the classifier itself is used to determine the attribute subset.

3.3 Time Series Analysis Models

In the thesis, three forecasting techniques were evaluated on the basis of their efficiency to forecast and their ability in producing coherent results with the actual event labels.

3.3.1 Autoregressive Integrated Moving Average (ARIMA)

The Autoregressive Integrated moving average (ARIMA) models are the most popular and effective statistical models for time series forecasting [58]. These models are based on the main principle that the future values of a time series are generated from a linear function of its past values and white noise terms.

The future value of a variable is a linear combination of past values and past errors and can be written as:

$$y_t = \sum_{i=1}^{p} \alpha_i yt - i + \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i}, \tag{3.9}$$

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. Parameter optimization is performed using Box-Jenkins methods [58] and Akaike Information Criterion (AIC) is used for order selection where the model that gives minimum AIC is selected as the best fit model.

3.3.2 Artificial Neural Networks (ANN)

ANN is a computational model implemented in computer science which is inspired by some of the behavioral and adaptive features of biological neural systems [59]. The basic objective of ANN is to build a model for mimicking the intelligence of human brain into machine. It has been suggested as a very successful alternative to the ARIMA models for time series forecasting and it gained enormous popularity in recent years. ANN can estimate any nonlinear continuous function up to any desired degree of accuracy [60]. The most common type of ANNs is a three layer back-propagation network: input, hidden and output node layers which are interconnected with different weights. Each node is called a neuron.

A three layer back-propagation network, which includes i input, j hidden and k output neurons, can be represented by the following equation:

$$y_k = f_k\left(a_k + \sum_{j \to k} w_{jk} f_j\left(a_j + \sum_{i \to j} w_{ij} x_i\right)\right) \tag{3.10}$$

where y_k is the neural output of the k neuron, f_k and f_j are activation functions. w_{jk} or w_{ij} represents weights and a_k and a_j are biases which multiply the signals processing from j to k or from i to j respectively [61].

3.3.3 Exponential Smoothing (ETS)

Exponential smoothing has become very popular as a forecasting model which can be applied to process a wide variety of time series data. Simple exponential smoothing is employed in the study. This model is frequently used short-term analysis and suitable for forecasting data with no trend or seasonal pattern. The simple exponential smoothing method is based on a weighted average of current and past observations assigning exponentially decreasing weights as the observation get older [62]. Given a weight α, the sequence of observed data x_t, the result of exponential smoothing algorithm y_t, the state of a time series is found using the following formula:

$$y_t = \alpha x_t + (1-\alpha) y_{t-1} \tag{3.11}$$

The value of the α is defined by minimizing the sum of squared errors produced by the model.

4. COMMUNITY EVOLUTION PREDICTION BASED ON TIME SERIES MODELING

In this chapter, the methodology for predicting community events using time series models is introduced. Since the next behavior of a community can be quantified and closely related to time factor, its temporal and structural features have been formulated as time series forecasting models. The intuition behind our algorithm is simple. We aim to estimate community feature values belonging to the time step to be predicted, thus discovering the communities of the related snapshot is avoided.

4.1 Methodology

The proposed method proceeds as follows: the evolving network is modeled by a series of static snapshots where each snapshot formed by the interactions up to that specific interval. We consider various predefined window lengths (w) of snapshots to constitute the graph for the temporal analysis. The communities of the graph are extracted using Louvain [55] community detection algorithm. Then, a broad range of structural and temporal community 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. Then, community matchings at consecutive time steps are found and these communities are labeled with significant events such as survive, growth, shrink, merge, split and dissolve according to our event detection procedure. Time series of length w is generated for each feature, thus each community has time series as the number of features. Afterwards, time series models are applied to estimate the next values of the features. Finally, model is trained with the several well known classifiers on the w length snapshots as the training data and the forecasted feature values of the communities as the test data. More specifically, the proposed method is performed in six steps: 1) Setting windows 2) Community detection 3) Feature extraction 4) Event tracking procedure 5) Time series analysis and 6) Classification. Each step is described as the following:

Definition 1. A graph $G = (V, E)$ be an undirected and unweighted graph where V is a set of nodes and E is a set of edges. Evolving graph is modeled as an ordered sequence of T graphs $\{G^1, G^2, ..., G^T\}$, where $G^i = (V^i, E^i)$ represents a static snapshot of the network at a given discrete time point t_i where $(i = 1, ..., T)$.

Algorithm 1 Community event prediction using time series analysis.

Input: A sequence of undirected and unweighted graphs: $G^- = G^{h+1}, ..., G^{h+w}$
Output: Prediction Results

for every horizon h where $h \leq (h+w+1)$ **do**
  for every graph G^t in the sequence G **do**
    Apply community detection algorithm
    Extract $C^t = \{C_1^t, C_2^t ...., C_n^t\}$
    for every community $C_i^t \in C^t$ **do**
      Calculate community features $F^i = f_1^i, ..., f_k^i$
      Calculate $Sim(C_i^t, C_j^{t+1})$ with $C_j^{t+1} \in C^{t+1}$
      Apply "Event Detection" algorithm (Algorithm 2)
      Create an instance with the features and label with the corresponding event
    end for
  end for
  for every community C_i^{h+w} **do**
    Constitute time series $\{TS_i^1, ..., TS_i^1\}$
    Apply forecasting models to produce $\{pf_i^1, pf_i^2, ..., pf_i^k\}$
  end for
  Apply classifiers to produce prediction results
end for
Apply classifiers to produce prediction results.

4.1.1 Setting windows

Sliding Window: We propose the use of an overlapping sliding window approach where only a predefined number of static graphs, namely window length (w), are considered for the temporal analysis. A horizon h be constituted by partitioning the time axis into time slots of fixed length w and then time slots are shifted in subsequent horizons. Therefore, a forgetting mechanism is employed by considering only the static graphs falling within each one of these slots. Hence, the past graph G^- and the next graph G^+ is defined as:

$$G^- = G^{[(h+1),(h+w)]}, G^+ = G^{(h+w+1)} \tag{4.1}$$

where $(h+w+1) \leq T$, and $0 \leq h \leq T-(w+1)$.

Such forgetting mechanism allows us to focus only on current events, by considering in the analysis only the most recent connections of the dynamic network. In Figure 4.3 we illustrate seven horizons $(h_k, k = 0, ..7)$ of an overlapping sliding window of length eight time points $(w = 8)$.

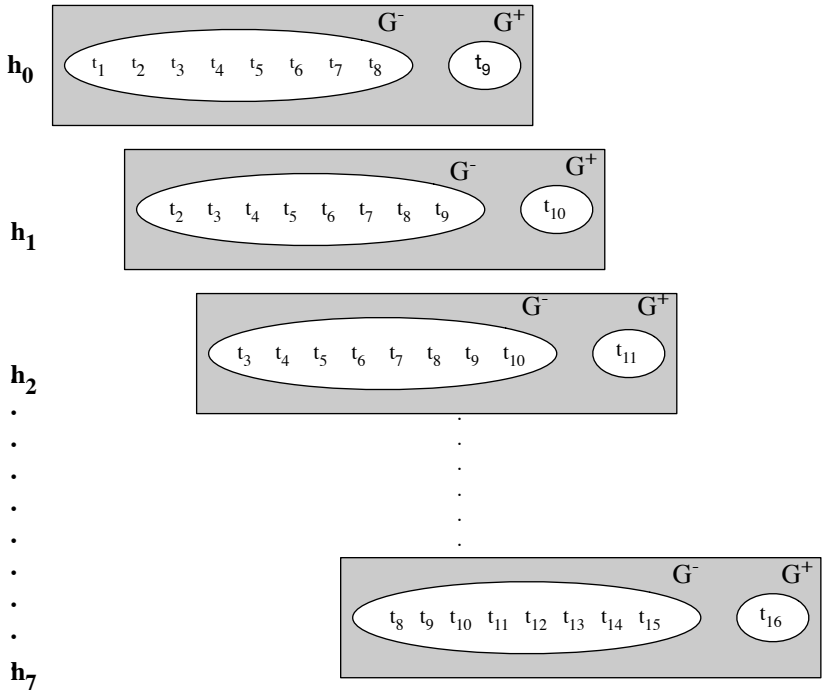

Figure 4.1 : Sliding windows.

Landmark Window

We also employ landmark window approach which relies on the specified amount of the whole past thus allowing us to capture persistent communities. As illustrated in Figure 4.2, 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. The past graph G^- and the next graph G^+ is defined as:

$$G^- = G^{[1,w]}, G^+ = G^{(w+1)} \tag{4.2}$$

where $(w+1) \leq T$.

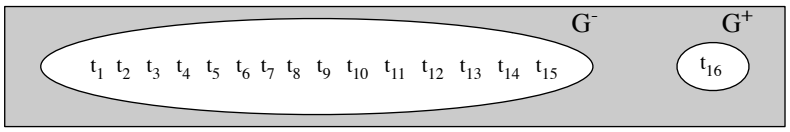

Figure 4.2 : Landmark windows.

Landmark window model corresponds to the h_0 of sliding window model.

4.1.2 Community detection

Definition 2. While G^i representing the ith snapshot of the graph, $C^i = \{C^i_1, C^i_2 ...., C^i_n\}$ 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. Louvain is a two phased hierarchical agglomerative approach proposed by Blondel *et al.*. In beginning each node is placed in its own community. During the first phase, each node is moved to one of its neighbours' community by maximizing the modularity gain or stays in its original community if no gain is possible. This procedure is applied repeatedly and sequentially for all nodes until no further improvement can be achieved. In the second phase, each community is considered as a node on its own. Then, the algorithm starts the phase one and both steps are repeated until stable communities are reached.

4.1.3 Feature extraction

Community features that may be important in tracking community evolution and measure structural and temporal aspects of the communities are extracted. Nine distinct community features are employed within the scope of the model. The features implemented within the algorithm encompass many structural and temporal properties such as node number, edge number, betweenness, degree, activeness and so on.

Structural community features: To gauge the structural properties of a community, we considered its node number, edge number, average of internal links, average of external links, average betweenness, average degree, and conductance.

Temporal community features: The temporal properties of the community, such as aging and activeness, were also quantified. Aging feature assesses the average lifetime of community members. It indicates whether the community is constituted by new arrivals or old members. Activeness feature measures the activity by evaluating the number of established connections of community members in the previous snapshot. Each measurement corresponds to a dimension in our feature space and the details are given in Table 4.1.

Table 4.1 : Community features.

No	Feature	Desription
f_1	Nodes	Number of nodes within the community i at time t.
f_2	Edges	Number of edges within the community i at time t.
f_3	Intra	Ratio of the total number of edges between the nodes inside the community to the number of nodes in the community.
f_4	Inter	Ratio of the total number of edges of nodes connected outside the community to the number of nodes in the community.
f_5	Activeness	Ratio of the total number of connections made in the previous timestamp by the nodes of the community to the number of nodes in the community.
f_6	Aging	Ratio of the total ages of the nodes in the community to the number of nodes in the community. With ages of nodes increasing by 1 at each timestamp, starting from zero.
f_7	Betweenness	Ratio of the total node betweenness in the community to the number of nodes in the community.
f_8	Degree	Ratio of the sum of degrees of the nodes in the community to the number of nodes in the community.
f_9	Conductance	Ratio of the number of edges in the community to the sum of degrees of the nodes in the community.

4.1.4 Event tracking procedure

To capture the changes that are likely to occur for a community, the six frequently occurring events known as *survive*, *growth*, *shrink*, *merge*, *split* and *dissolve* are considered. In order to track the evolution, the set of communities at consecutive snapshots have to be matched with each other. Therefore, the match of a given community at time t among the communities at time $t + 1$ should be found. The output of the matching process between C_i^t and $\{C_1^{t+1}, ..., C_j^{t+1}\}$ reveals a series of community evolution events which are used as the class label of C_i^t in the classification. We defined two distinct thresholds: Similarity Threshold (θ) and Fluctuation Threshold (ϕ).

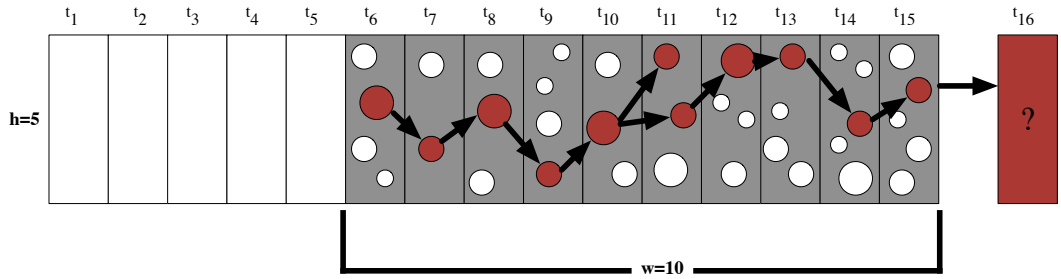

Figure 4.3 : Community evolution.

Two communities at consecutive snapshots are said to be matched to each other if the ratio of their similarity value $Sim(C_i^t, C_j^{t+1})$ exceeds the threshold θ [4], more formally:

$$Sim(C_i^t, C_j^{t+1}) = min\left(\frac{|C_i^t \cap C_j^{t+1}|}{|C_i^t|}, \frac{|C_i^t \cap C_j^{t+1}|}{|C_j^{t+1}|}\right) \geq \theta \tag{4.3}$$

Event detection procedure works as follows: for a given community C_i^t, the similarity between C_i^t 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 other than *dissolve*. If a community has one successor, it may have one of the three possible events $\{survive, growth, shrink\}$. Herein, we propose a metric, namely *fluctuation*, for the purpose of computing the percentage of increase/decrease in the number of community members. An illustration is given in Figure 4.4 where the light-colored object represents the community to be labeled at time step t.

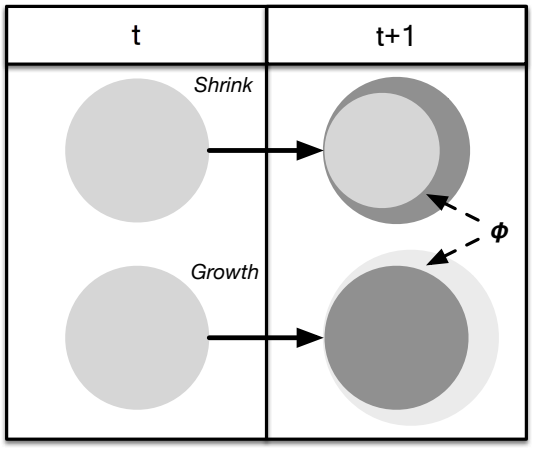

Figure 4.4 : Fluctuation in survive event which results in growth and shrink.

Fluctuation rate would give us an accurate result of whether i) the community has grown (i.e. there is a substantial percentage increase in the number of members), or ii)

the community has survived (i.e. there is a negligible increase/decrease in the number of members), or iii) the community has 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:

$$fluctuation(C_i^t, C_j^{t+1}) = \frac{n_j^{t+1}}{n_i^t} - 1 \tag{4.4}$$

We could then label a community as having survived, 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 the case of *split* 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 the case of *merge*, where $C_*^r \subseteq C^r$. 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.

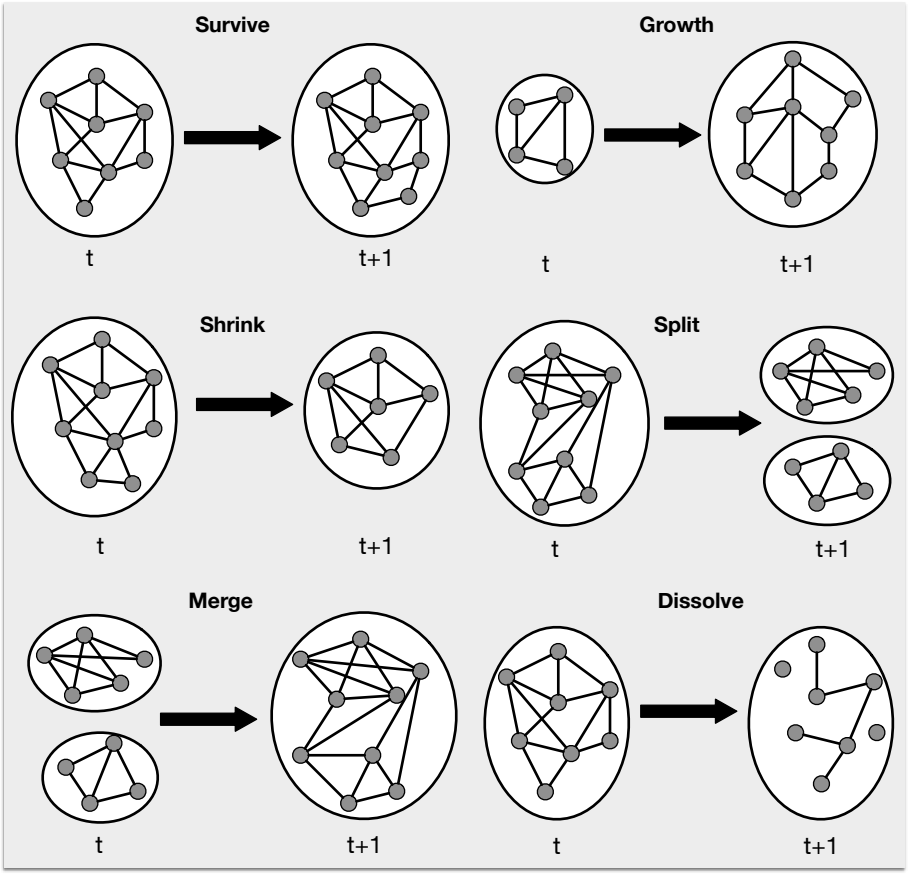

Figure 4.5 : Schematic diagram of types of community events.

The formal definitions of the events are as follows:

Definition 3. A community C_i^t at time t is said to be labeled with ***survive*** event if there exists 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

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

Definition 4. A community C_i^t at time t is said to be labeled with ***growth*** event if there exists 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

$$Sim(C_i^t, C_j^{t+1}) \geq \theta \text{ and } fluctuation(C_i^t, C_j^{t+1}) > \phi$$

Definition 5. A community C_i^t at time t is said to be labeled with ***shrink*** event if there exists 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

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

Definition 6. 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 θ. Thus, C_i^t has split if

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

Definition 7. A set of communities $C_*^t = \{C_1^t ... C_i^t\}$ 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 communities C_*^t has merged if

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

Definition 8. A community C_i^t at time t is said to be labeled with ***dissolve*** event if there is no matching community at time $t+1$ whose similarity threshold is greater than θ. Thus, C_i^t has dissolved if

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

Overall event detection procedure is given in Algorithm 2.

Algorithm 2 Event detection.

Input: $Sim(C_i^t, C_j^{t+1})$
Output: Event label

if $\forall C_i^t \in C_*^t, Sim(C_i^t, C_j^{t+1}) \geq \theta$ and $Sim(\cup\{C_*^t\}, C_j^{t+1}) \geq \theta$ **then**
    label is *Merge*
else if $\forall C_j^{t+1} \in C_*^{t+1}, Sim(C_i^t, C_j^{t+1}) \geq \theta$ and $Sim(C_i^t, \cup\{C_*^{t+1}\}) \geq \theta$ **then**
    label is *Split*
else if $Sim(C_i^t, C_j^{t+1}) \geq \theta$ and $-\phi \leq fluctuation(C_i^t, C_j^{t+1}) \leq \phi$ **then**
    label is *Survive*
else if $Sim(C_i^t, C_j^{t+1}) \geq \theta$ and $fluctuation(C_i^t, C_j^{t+1}) > \phi$ **then**
    label is *Growth*
else if $Sim(C_i^t, C_j^{t+1}) \geq \theta$ and $fluctuation(C_i^t, C_j^{t+1}) < -\phi$ **then**
    label is *Shrink*
else
    label is *Dissolve*
end if

4.1.5 Time series analysis

There are two main objectives of the time series prediction, namely, 1) To predict the change of community features of a given community over time 2) To quantitatively characterize the development process of communities. Let w be the window length and h is the horizon number, our task is to predict the community events of subsequent snapshot t_{h+w+1} by concerning the data of $[t_{h+1}, t_{h+w}]$. In our proposed approach, we build time series for each community feature of the set $\{f_1, f_2, ..., f_k\}$ related to community C_i^{h+w+1}. Thus, as instance, time series related to f_1 of C_i^{h+w+1} become $TS_i^1 = \{f_1^{h+1}, f_1^{h+2}, ..., f_1^{h+w}\}$. As a results of applying time series forecasting methods, the time serie TS_i^1 produce a predicted value of the feature, pf_i^1. Hence, the attributes of community C_i^{h+w+1} which will be used to detect event label by classifiers is $\{pf_i^1, pf_i^2, ..., pf_i^k\}$.

4.1.6 Classification

At the last step, the well-known classifiers are adopted in order to determine corresponding event labels: Adaboost (Base Classifier: Decision Stump), Simple Cart, Bayes Net (Estimator: SimpleEstimator, Search Algorithm: K2) and Bagging (Base Classifier: REPTree) [63]. The instances obtained from G^- are used as the training set for the classifiers. The goal of classification is to predict the next event of a given community at G^+.

4.2 Experimental Study

4.2.1 Datasets

Two citation networks from the High Energy Physics Theory (hepTh) and High Energy Physics Phenomenology (hepPh) sections in e-print arXiv are used in this phase. The hepTh and hepPh are the collaboration networks that covers scientific collaborations between authors' papers. Table 4.2 shows detailed information about the datasets. Four year period in each dataset is experimented by taking three months interval snapshots being sixteen time steps and fifteen evolution transitions in total.

Table 4.2 : hepTh and hepPh datasets.

Name	No. of papers	No. of citations	Time period
hepTh	22623	107857	1993-1997
hepPh	28073	477971	1993-1997

4.2.2 Experimental configuration

The datasets were represented as nodes and relationships that connect nodes were represented as edges. Each entry in the dataset was annotated with the timestamp of the connection. Each dataset was divided into sixteen time frames. A network snapshot at time t contains all nodes and edges existing at the corresponding time or prior to t. Networks were taken to be undirected and unweighted. The communities whose size was smaller than three members were ignored. This diminishment was performed intuitively, since a group of nodes that consist of less than three members does not form a community. Setting the appropriate similarity threshold (θ) and fluctuation threshold (ϕ) is very crucial in the study. A low value of θ may lead to a substantial amount of

matching communities, while high value of θ results in more dissolutions. Similarly, low value of ϕ results in a small amount of survival and an excessive number of grown and shrunken communities. As ϕ increases, the number of survive events increases. To select the optimal thresholds, θ and ϕ values have been trained and event rates are observed. In order to investigate the impact of the similarity threshold (θ) on the community evolution, experiments were conducted where θ was varied from 0.1 to 1 and ϕ fixed at 0.15, yielding the results as depicted in Figure 4.6.

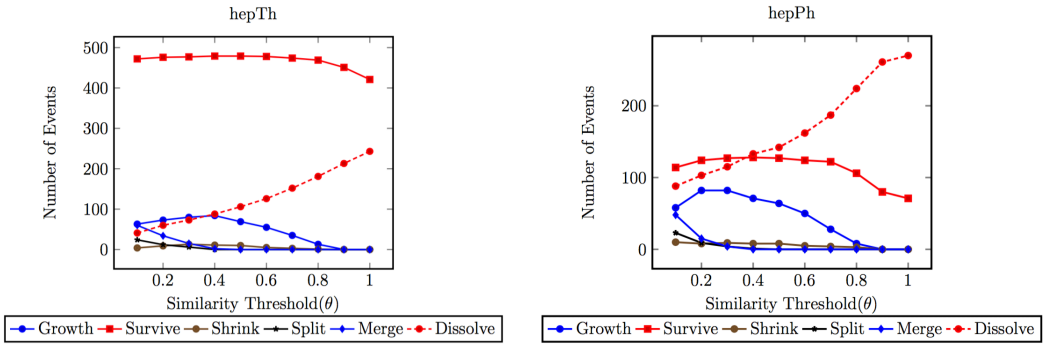

Figure 4.6 : Number of events vs. similarity threshold (θ) of hepTh and hepPh datasets.

Figure 4.6 shows that the θ value has a noticable effect on the observed events: the number of survive, growth, shrink, merge, split events drop as θ increases, while there are more dissolve events. A low θ value induces a slew of matching communities thus enabling the observation of a significant number of events apart from dissolve. Besides, high values of θ result in a small amount of matching communities and events other than dissolve. Besides, high values of θ result in a small amount of matching communities and events other than dissolve. Therefore, selecting the optimal similarity threshold is crucial and it differs with respect to the structural characteristics of the networks. We intended to select a threshold where the event numbers were as well-proportioned in their distribution as possible. In hepTh, θ is selected as 0.3 due to decrease in the number of split and merge events after that threshold. θ is picked as 0.2 for hepPh dataset, since it is a breakpoint particularly for growth and split events.

Fluctuation threshold (ϕ) is the other metric to be tuned and has significant effect on the survive, growth and shrink events. So as to determine the optimal ϕ value, experiments were conducted where ϕ was varied from 0.1 to 0.5 while fixing θ at 0.3, yielding the results as shown in Figure 4.7.

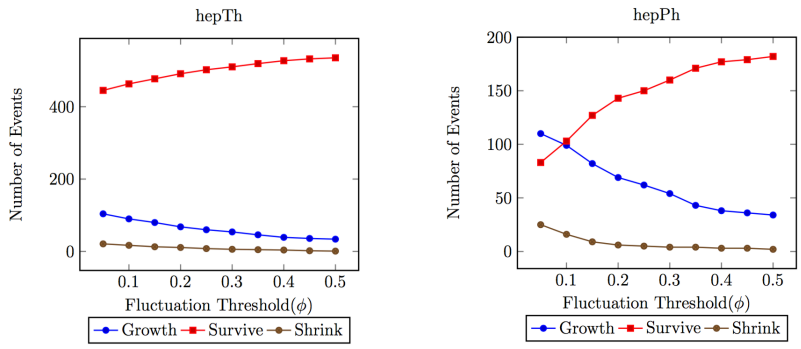

Figure 4.7 : Number of events vs. fluctuation threshold (ϕ) of hepTh and hepPh datasets.

It can easily be observed that the number of survive events increases, whereas the number of growth and shrink events drops gradually as ϕ increases. In hepTh, balanced distribution has been attained with $\phi = 0.15$. In hepPh, the gap between the observed event rates begins to grow after $\phi = 0.1$. Thus, the choice of similarity and fluctuation thresholds is $\theta = 0.3, \phi = 0.15$ in hepTh, and $\theta = 0.2, \phi = 0.1$ in hepPh respectively.

4.2.3 Results

Initially, we give the prediction results of the event detection procedure. The classification results of the event detection procedure for various window lengths $w = \{8, ..., 15\}$ in predicting events of timestamps $t_9, ..., t_{16}$ have been provided in terms of F-measure using classifiers Adaboost, Simple Cart, Bayes Net and Bagging (for simplicity, four classifiers are selected including the best and worst performers). The results of each classifier are the average values of 5 independent runs. Table 4.3 reports the event prediction results of hepTh dataset.

Results in Table 4.3 indicate that event prediction model accurately predicts community events and the accuracy values are in the range **0.54** to **0.96**. It can also be observed that Adaboost performed worst while other classifiers producing higher results. The results are no evidence of overall superiority of one window length over another due to fluctuative performance of both window lengths and classifiers.

Table 4.3 : hepTh-event prediction results.

w	Classifier	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
w=8	Adaboost	0.83	0.71	0.78	0.78	0.66	0.79	0.80	0.80
	Simple Cart	0.76	0.90	0.92	0.88	0.95	0.92	0.88	0.86
	Bayes Net	0.71	0.80	0.78	0.80	0.76	0.85	0.75	0.78
	Bagging	0.80	0.88	0.87	0.88	0.92	0.86	0.82	0.83
w=9	Adaboost		0.72	0.67	0.70	0.66	0.65	0.63	0.62
	Simple Cart		0.86	0.89	0.88	0.95	0.92	0.90	0.93
	Bayes Net		0.82	0.80	0.80	0.70	0.75	0.74	0.80
	Bagging		0.87	0.94	0.88	0.96	0.80	0.84	0.90
w=10	Adaboost			0.68	0.76	0.65	0.69	0.68	0.62
	Simple Cart			0.95	0.97	0.96	0.96	0.97	0.97
	Bayes Net			0.82	0.96	0.87	0.83	0.87	0.80
	Bagging			0.92	0.98	0.94	0.95	0.94	0.92
w=11	Adaboost				0.68	0.64	0.69	0.75	0.54
	Simple Cart				0.90	0.95	0.90	0.96	0.96
	Bayes Net				0.71	0.87	0.78	0.85	0.70
	Bagging				0.88	0.94	0.91	0.94	0.90
w=12	Adaboost					0.65	0.70	0.70	0.73
	Simple Cart					0.90	0.85	0.95	0.94
	Bayes Net					0.81	0.85	0.84	0.82
	Bagging					0.90	0.93	0.92	0.93
w=13	Adaboost						0.62	0.65	0.65
	Simple Cart						0.93	0.90	0.94
	Bayes Net						0.72	0.80	0.70
	Bagging						0.89	0.88	0.91
w=14	Adaboost							0.64	0.60
	Simple Cart							0.96	0.92
	Bayes Net							0.79	0.80
	Bagging							0.94	0.91
w=15	Adaboost								0.61
	Simple Cart								0.86
	Bayes Net								0.73
	Bagging								0.83

The results of the event detection procedure of hepPh have been provided in Table 4.4. Similar to hepTh dataset, the procedure has produced good prediction results. The best accuracy for the hepPh dataset is **0.97** and the worst is **0.52**. Results in Table 4.3 and Table 4.4 quantitatively indicate that good performance is obtained by proposed event detection procedure. So far, conducted experiments have shown the non-necessity of investigating long period of past to achieve better event prediction. For the sake of example, while predicting t_{15} in Table 4.4, window length of eight $(w = 8)$ produce almost same results with $w = 13$ or $w = 14$. In predicting t_{16}, the results of $w = 15$ are relatively better than $w = 8$, but comparable with $w = 9$. Observed differences are negligible when we consider the computation cost of larger window lengths.

Table 4.4 : hepPh-event prediction results.

w	Classifier	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
w=8	Adaboost	0.55	0.53	0.57	0.57	0.67	0.63	0.64	0.68
	Simple Cart	0.97	0.93	0.90	0.93	0.91	0.96	0.97	0.95
	Bayes Net	0.84	0.80	0.88	0.86	0.83	0.91	0.90	0.86
	Bagging	0.93	0.87	0.91	0.84	0.90	0.93	0.96	0.90
w=9	Adaboost		0.53	0.52	0.60	0.60	0.62	0.70	0.64
	Simple Cart		0.89	0.93	0.97	0.97	0.97	0.98	0.97
	Bayes Net		0.92	0.87	0.89	0.88	0.87	0.90	0.93
	Bagging		0.96	0.93	0.96	0.94	0.94	0.96	0.95
w=10	Adaboost			0.60	0.66	0.77	0.74	0.79	0.64
	Simple Cart			0.90	0.92	0.97	0.99	0.97	0.95
	Bayes Net			0.88	0.91	0.90	0.93	0.90	0.95
	Bagging			0.89	0.92	0.96	0.92	0.95	0.95
w=11	Adaboost				0.73	0.62	0.68	0.62	0.67
	Simple Cart				0.96	0.96	0.96	0.95	0.92
	Bayes Net				0.90	0.92	0.92	0.92	0.88
	Bagging				0.96	0.95	0.94	0.92	0.88
w=12	Adaboost					0.68	0.71	0.62	0.65
	Simple Cart					0.97	0.96	0.94	0.89
	Bayes Net					0.78	0.80	0.83	0.81
	Bagging					0.93	0.93	0.88	0.90
w=13	Adaboost						0.72	0.61	0.64
	Simple Cart						0.96	0.96	0.96
	Bayes Net						0.90	0.91	0.90
	Bagging						0.92	0.93	0.95
w=14	Adaboost							0.70	0.68
	Simple Cart							0.97	0.95
	Bayes Net							0.90	0.92
	Bagging							0.96	0.95
w=15	Adaboost								0.70
	Simple Cart								0.94
	Bayes Net								0.91
	Bagging								0.95

After the success of event detection procedure has been shown, we evaluate the performance of the predictive models by comparing their ability to accurately predict the actual events. The ANN, ARIMA and ETS models are tested through various Weka classifiers. The results of each classifier are averaged over 5 independent runs. First, we investigate the matching up values of the actual event labels and the identified event labels by the proposed framework in terms of F-measure. Matching up values represent to what extend the identified event labels are overlap with the actual event labels. The performance of ANN, ARIMA and ETS models are analyzed with various window lengths $w = \{8, ..., 15\}$. Besides, comparative analysis of the landmark window and sliding window has been provided. Afterwards, the global performance of

the forecasting models are evaluated by an accuracy measure, such as Mean Absolute Percentage Error (MAPE). The mathematical formulation of the considered measure is given below:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}|\frac{A_j^i - F_j^i}{A_j^i}| \tag{4.5}$$

where A_j^i represents the actual f_j value of C_i and F_j^i is the forecasted f_j value of C_i. Lower MAPE indicates better forecasts.

Table 4.5 : hepTh-match up results of ANN.

Timestamp	Classifier	Landmark Window	Sliding Window						
			w=8	w=9	w=10	w=11	w=12	w=13	w=14
t_9	Adaboost	1.00							
	Simple Cart	1.00							
	Bayes Net	0.99							
	Bagging	0.97							
t_{10}	Adaboost	1.00	1.00						
	Simple Cart	1.00	1.00						
	Bayes Net	0.97	0.95						
	Bagging	0.98	0.91						
t_{11}	Adaboost	1.00	1.00	1.00					
	Simple Cart	1.00	1.00	1.00					
	Bayes Net	0.94	1.00	0.94					
	Bagging	1.00	1.00	1.00					
t_{12}	Adaboost	1.00	1.00	1.00	1.00				
	Simple Cart	1.00	1.00	0.97	1.00				
	Bayes Net	0.80	0.89	0.93	0.85				
	Bagging	1.00	1.00	1.00	0.95				
t_{13}	Adaboost	1.00	1.00	1.00	1.00	1.00			
	Simple Cart	1.00	0.92	1.00	1.00	1.00			
	Bayes Net	0.88	0.83	0.96	0.83	0.81			
	Bagging	0.82	0.69	0.95	0.90	0.94			
t_{14}	Adaboost	1.00	0.99	1.00	1.00	1.00	1.00		
	Simple Cart	1.00	0.92	0.67	1.00	1.00	1.00		
	Bayes Net	0.84	0.96	1.00	1.00	0.84	0.87		
	Bagging	0.78	0.84	0.69	0.82	0.95	0.88		
t_{15}	Adaboost	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
	Simple Cart	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
	Bayes Net	0.86	0.84	0.78	0.96	0.96	0.89	0.85	
	Bagging	0.99	0.73	0.69	0.83	0.91	0.84	0.99	
t_{16}	Adaboost	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	Simple Cart	1.00	0.84	0.83	0.95	0.86	1.00	0.89	1.00
	Bayes Net	0.90	0.79	0.80	0.90	0.89	0.92	0.81	0.82
	Bagging	0.92	0.72	0.81	0.96	0.82	0.84	0.75	0.92

The time series analysis results of the hepTh dataset using ANN, ARIMA and ETS have shown in Table 4.5, Table 4.6 and Table 4.7 respectively. The left most column

of the tables indicates the timestamp to be predicted. As instance, the sixth row of the $w = 8$ column represents the matching up value of Simple Cart classifier in predicting t_{10} using eight windows.

Our results in Table 4.5 indicate that the predicted event labels of the framework using ANN model in hepTh dataset substantially intersect with the actual event labels as high as **1.00**. We can also observe that the minimum matching up value is **0.67**. Entirely, it is not possible to mention about the superiority of landmark or sliding window approach in this table. Nonetheless, in some horizons either of them out competes, e.g. sliding window approach gave better results a major part in predicting t_{14}.

Table 4.6 : hepTh-match up results of ARIMA.

Timestamp	Classifier	Landmark Window	Sliding Window						
			w=8	w=9	w=10	w=11	w=12	w=13	w=14
t9	Adaboost	1.00							
	Simple Cart	1.00							
	Bayes Net	0.99							
	Bagging	0.97							
t10	Adaboost	1.00	1.00						
	Simple Cart	1.00	1.00						
	Bayes Net	0.97	0.96						
	Bagging	0.99	0.90						
t11	Adaboost	1.00	1.00	1.00					
	Simple Cart	1.00	1.00	1.00					
	Bayes Net	0.94	0.99	0.94					
	Bagging	1.00	1.00	1.00					
t12	Adaboost	1.00	1.00	1.00	1.00				
	Simple Cart	1.00	1.00	1.00	1.00				
	Bayes Net	0.84	0.91	0.92	0.79				
	Bagging	1.00	1.00	1.00	0.95				
t13	Adaboost	1.00	1.00	1.00	1.00	1.00			
	Simple Cart	1.00	0.92	1.00	1.00	1.00			
	Bayes Net	0.88	0.83	0.96	0.88	0.81			
	Bagging	0.83	0.69	0.96	0.91	0.95			
t14	Adaboost	1.00	0.97	1.00	1.00	1.00	1.00		
	Simple Cart	1.00	0.92	0.67	1.00	1.00	1.00		
	Bayes Net	0.84	0.92	0.96	0.95	0.87	0.83		
	Bagging	0.78	0.83	0.74	0.81	0.93	0.88		
t15	Adaboost	1.00	1.00	0.98	1.00	1.00	1.00	1.00	
	Simple Cart	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
	Bayes Net	0.85	0.81	0.77	0.96	0.93	0.84	0.81	
	Bagging	1.00	0.70	0.73	0.77	0.90	0.84	0.99	
t16	Adaboost	1.00	0.99	1.00	0.99	1.00	1.00	1.00	1.00
	Simple Cart	1.00	0.83	0.83	0.95	0.87	1.00	0.84	1.00
	Bayes Net	0.85	0.78	0.79	0.88	0.83	0.92	0.81	0.79
	Bagging	0.86	0.71	0.78	0.82	0.75	0.87	0.75	0.89

Similarly, good results are attained using Arima with **1.00** being the highest and **0.67** being the worst value (Table 4.6). Landmark substantially overcomes smaller window lengths ($w = 8$ and $w = 9$) while predicting t_{15} and t_{16}. There exist more or less equality in predicting other timestamps.

Table 4.7 : hepTh-match up results of ETS.

Timestamp	Classifier	Landmark Window	Sliding Window w=8	w=9	w=10	w=11	w=12	w=13	w=14
t9	Adaboost	1.00							
	Simple Cart	1.00							
	Bayes Net	1.00							
	Bagging	0.97							
t10	Adaboost	1.00	1.00						
	Simple Cart	1.00	1.00						
	Bayes Net	0.97	0.96						
	Bagging	0.98	0.92						
t11	Adaboost	1.00	1.00	1.00					
	Simple Cart	1.00	1.00	1.00					
	Bayes Net	0.94	1.00	0.94					
	Bagging	1.00	1.00	1.00					
t12	Adaboost	1.00	1.00	1.00	1.00				
	Simple Cart	1.00	1.00	0.97	1.00				
	Bayes Net	0.85	0.89	0.93	0.87				
	Bagging	1.00	0.99	1.00	0.95				
t13	Adaboost	1.00	1.00	1.00	1.00	1.00			
	Simple Cart	1.00	0.92	1.00	1.00	1.00			
	Bayes Net	0.87	0.87	0.96	0.83	0.81			
	Bagging	0.81	0.69	0.93	0.90	0.95			
t14	Adaboost	1.00	0.99	1.00	1.00	1.00	1.00		
	Simple Cart	1.00	0.92	0.67	1.00	1.00	1.00		
	Bayes Net	0.85	0.97	1.00	0.99	0.86	0.87		
	Bagging	0.77	0.84	0.74	0.82	0.93	0.87		
t15	Adaboost	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
	Simple Cart	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
	Bayes Net	0.86	0.85	0.78	1.00	0.96	0.89	0.87	
	Bagging	1.00	0.70	0.68	0.83	0.90	0.84	0.99	
t16	Adaboost	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
	Simple Cart	0.84	0.84	0.83	0.95	0.86	1.00	0.88	1.00
	Bayes Net	0.79	0.79	0.80	0.89	0.89	0.92	0.88	0.84
	Bagging	0.72	0.72	0.77	0.87	0.82	0.87	0.76	0.92

As it is presented in Table 4.7, ETS results of hepTh dataset vary between **0.67** and **1.00**. Sliding windows perform better specifically in predicting t_{14} and t_{16} and both approach performed almost identical in the remaining timestamps.

We can easily see that the predicted events using time series analysis models highly overlap with the actual events in hepTh dataset. The classifier results fluctuates as the

window length changes. Despite the fluctuation in the results, sliding window works well in particular with ARIMA and ETS model as against to landmark.

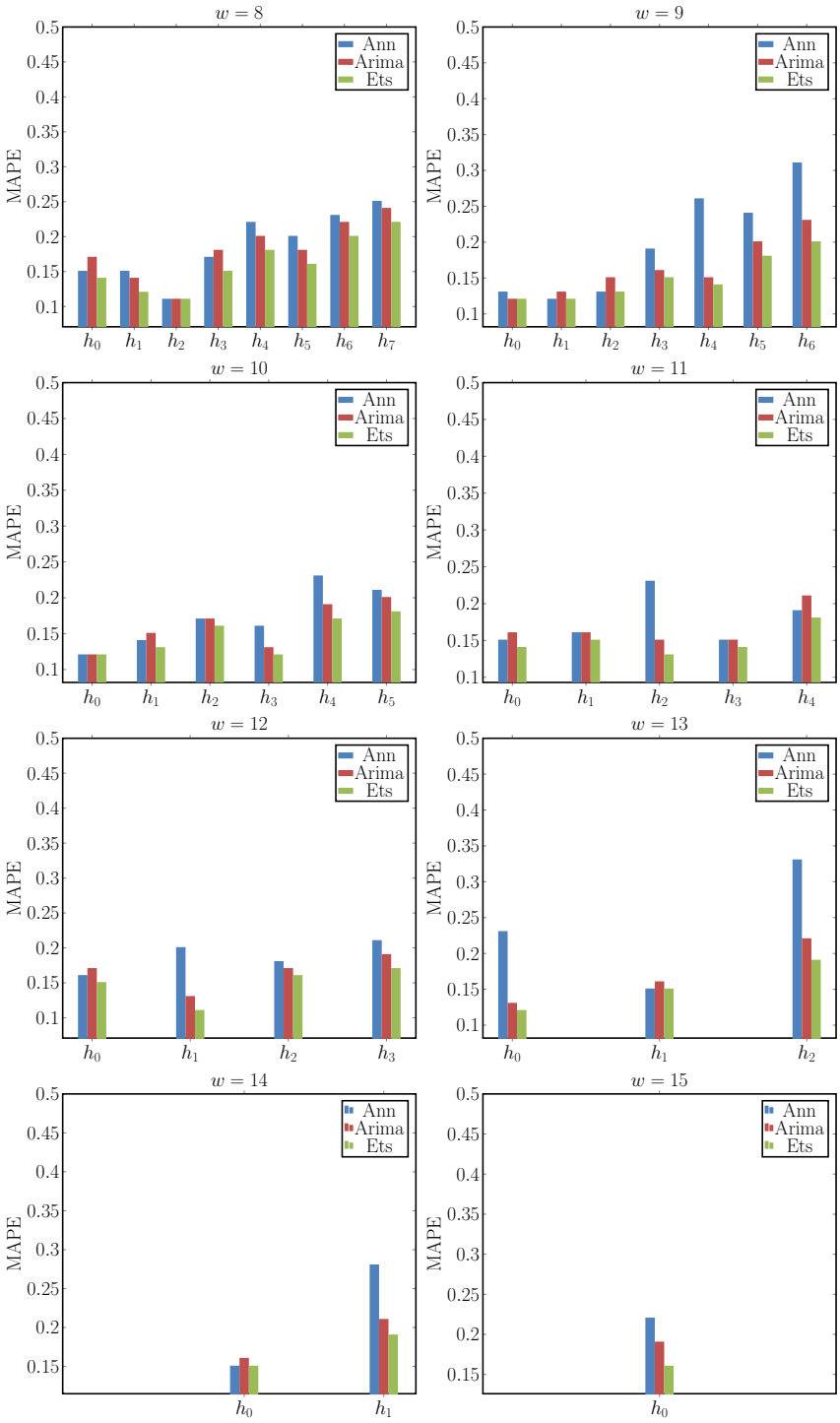

Figure 4.8 : hepTh-MAPE results.

MAPE rates of hepTh dataset are figured demostratively horizon results for each w in Figure 4.8. Note that h_0 corresponds to landmark approaach. As it can be seen, the mean performance of ETS is better than the other models. The ARIMA model stands

in the second place and ANN performed worst. Furthermore, we observe an inverse correlation between error estimates and matching rates. As instance, a lowest error rate for all models are obtained in $w = 8$ and h_2 which corresponds predicting t_{11} with $w = 8$ where higher F-measure results are attained in a considerable extend. The related MAPE and F-measure intervals for the models are as follows: ANN (MAPE:0.11, F-measure:[1]), ARIMA (MAPE:0.11, F-measure:[0.99,1]) and ETS (MAPE:0.11, F-measure:[1]). Another example is the abrupt rising of ANN error rate at $w = 13$ in prediction of second horizon (h_2) which results in decrease on classifiers Bayes Net and Bagging.

Table 4.8 : hepPh-match up results of ANN.

Timestamp	Classifier	Landmark Window	Sliding Window						
			w=8	w=9	w=10	w=11	w=12	w=13	w=14
t_9	Adaboost	0.92							
	Simple Cart	1.00							
	Bayes Net	0.97							
	Bagging	0.88							
t_{10}	Adaboost	1.00	1.00						
	Simple Cart	0.86	0.81						
	Bayes Net	0.88	0.98						
	Bagging	0.86	0.87						
t_{11}	Adaboost	1.00	1.00	1.00					
	Simple Cart	1.00	0.96	0.81					
	Bayes Net	1.00	1.00	0.86					
	Bagging	0.97	1.00	0.96					
t_{12}	Adaboost	1.00	1.00	1.00	1.00				
	Simple Cart	0.88	1.00	0.95	0.68				
	Bayes Net	1.00	0.97	0.97	0.89				
	Bagging	0.96	0.97	0.91	0.91				
t_{13}	Adaboost	1.00	1.00	0.92	0.98	0.87			
	Simple Cart	1.00	1.00	0.81	0.90	0.91			
	Bayes Net	1.00	1.00	0.90	1.00	0.76			
	Bagging	0.83	0.90	0.88	0.95	0.79			
t_{14}	Adaboost	1.00	1.00	1.00	1.00	1.00	1.00		
	Simple Cart	1.00	1.00	1.00	1.00	0.68	0.92		
	Bayes Net	1.00	0.89	0.97	1.00	1.00	0.98		
	Bagging	0.78	0.79	0.67	0.65	0.68	0.61		
t_{15}	Adaboost	1.00	1.00	1.00	0.92	1.00	1.00	1.00	
	Simple Cart	0.77	0.71	0.95	0.87	1.00	1.00	1.00	
	Bayes Net	0.86	0.89	0.82	0.76	0.92	0.97	0.92	
	Bagging	0.73	0.88	0.82	0.38	0.37	0.62	0.68	
t_{16}	Adaboost	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Simple Cart	1.00	1.00	1.00	1.00	0.98	0.66	1.00	0.98
	Bayes Net	0.97	0.94	1.00	0.88	0.88	0.95	0.99	0.92
	Bagging	0.87	0.98	0.87	0.89	0.88	0.87	0.79	0.71

Matching up results of hepPh dataset using ANN, ARIMA and ETS models are reported in Table 4.8, Table 4.9 and Table 4.10 respectively. Table 4.8 reveals that results of ANN model in hepPh dataset change between **0.37** and **1.00**. We can deduce that the proposed model is more successful in smaller window lengths, especially with $w = 8$ and $w = 9$.

Table 4.9 displays the ARIMA results of hepPh datasets. The lowest matching rate is found as **0.30** and the highest rate is **1.00**. Likewise ANN, ARIMA is successful in smaller window lengths. On the other hand, we observe that the mean value of the landmark window results are slightly lower than ANN.

Table 4.9 : hepPh-match up results of ARIMA.

Timestamp	Classifier	Landmark Window	Sliding Window						
			w=8	w=9	w=10	w=11	w=12	w=13	w=14
t_9	Adaboost	0.76							
	Simple Cart	1.00							
	Bayes Net	0.97							
	Bagging	0.88							
t_{10}	Adaboost	0.88	1.00						
	Simple Cart	0.86	0.74						
	Bayes Net	0.78	0.94						
	Bagging	0.86	0.64						
t_{11}	Adaboost	0.98	0.97	1.00					
	Simple Cart	1.00	0.92	0.84					
	Bayes Net	1.00	1.00	0.86					
	Bagging	0.97	1.00	0.96					
t_{12}	Adaboost	0.90	1.00	1.00	0.90				
	Simple Cart	0.78	1.00	0.95	0.58				
	Bayes Net	0.90	0.97	0.97	0.89				
	Bagging	0.86	0.97	0.91	0.84				
t_{13}	Adaboost	1.00	0.91	0.92	0.98	0.87			
	Simple Cart	1.00	0.98	0.81	0.90	0.83			
	Bayes Net	1.00	1.00	0.90	1.00	0.76			
	Bagging	0.83	0.88	0.88	0.95	0.76			
t_{14}	Adaboost	1.00	1.00	1.00	1.00	1.00	0.98		
	Simple Cart	1.00	1.00	1.00	1.00	0.68	0.90		
	Bayes Net	1.00	0.89	0.97	1.00	1.00	0.96		
	Bagging	0.60	0.80	0.76	0.73	0.68	0.60		
t_{15}	Adaboost	1.00	1.00	1.00	0.92	1.00	1.00	1.00	
	Simple Cart	0.67	0.78	0.95	0.87	1.00	1.00	1.00	
	Bayes Net	0.86	0.89	0.82	0.76	0.92	0.97	0.92	
	Bagging	0.59	0.86	0.84	0.38	0.30	0.59	0.58	
t_{16}	Adaboost	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Simple Cart	1.00	1.00	1.00	1.00	0.98	0.66	1.00	0.98
	Bayes Net	0.97	0.96	1.00	0.88	0.88	0.95	0.99	0.92
	Bagging	0.86	0.98	0.87	0.87	0.88	0.82	0.73	0.71

The ETS results of hepPh dataset is as given in Table 4.10 with **1.00** being the best and **0.37** being the worst match up result as in the other models. Similarly, sliding window approach with smaller window lengths exceeds landmark in most instances.

Table 4.10 : hepPh-match up results of ETS.

Timestamp	Classifier	Landmark Window	Sliding Window						
			w=8	w=9	w=10	w=11	w=12	w=13	w=14
t_9	Adaboost	0.92							
	Simple Cart	1.00							
	Bayes Net	0.97							
	Bagging	0.88							
t_{10}	Adaboost	1.00	1.00						
	Simple Cart	0.86	0.89						
	Bayes Net	0.88	0.98						
	Bagging	0.86	0.64						
t_{11}	Adaboost	1.00	1.00	1.00					
	Simple Cart	1.00	0.96	0.84					
	Bayes Net	1.00	1.00	0.86					
	Bagging	0.97	1.00	0.96					
t_{12}	Adaboost	1.00	1.00	1.00	1.00				
	Simple Cart	0.88	1.00	0.95	0.68				
	Bayes Net	1.00	0.97	0.97	0.89				
	Bagging	0.96	0.97	0.91	0.91				
t_{13}	Adaboost	1.00	1.00	0.92	0.98	0.87			
	Simple Cart	1.00	1.00	0.81	0.90	0.91			
	Bayes Net	1.00	1.00	0.90	1.00	0.78			
	Bagging	0.83	0.90	0.88	0.95	0.76			
t_{14}	Adaboost	1.00	1.00	1.00	1.00	1.00	1.00		
	Simple Cart	1.00	1.00	1.00	1.00	0.68	0.92		
	Bayes Net	1.00	0.89	0.97	1.00	1.00	0.98		
	Bagging	0.58	0.65	0.67	0.65	0.68	0.61		
t_{15}	Adaboost	1.00	1.00	1.00	0.92	1.00	1.00	1.00	
	Simple Cart	0.77	0.78	0.95	0.87	1.00	1.00	1.00	
	Bayes Net	0.86	0.89	0.82	0.76	0.92	0.97	0.92	
	Bagging	0.73	0.88	0.84	0.54	0.37	0.62	0.68	
t_{16}	Adaboost	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Simple Cart	1.00	1.00	1.00	1.00	0.98	0.66	1.00	0.98
	Bayes Net	0.97	0.96	1.00	0.88	0.88	0.95	0.99	0.92
	Bagging	0.87	0.98	0.87	0.87	0.88	0.89	0.79	0.63

Figure 4.9 has shown the MAPEs of hepPh dataset in various window lenghts. It is obvious that the forecasting error made by ANN model is worser than the other models and ETS results are slightly better than the ARIMA. We can also observe the inverse correlation between the MAPEs and matching results. To give an example, the horizon h_0 of $w = 15$ is one of the lowest MAPEs obtained horizon commonly by all models which corresponds predicting t_{16} using landmark window. F-measure values of this

horizon also verifies the correlation with higher results by having values in interval of $[0.80, 1]$ for all models.

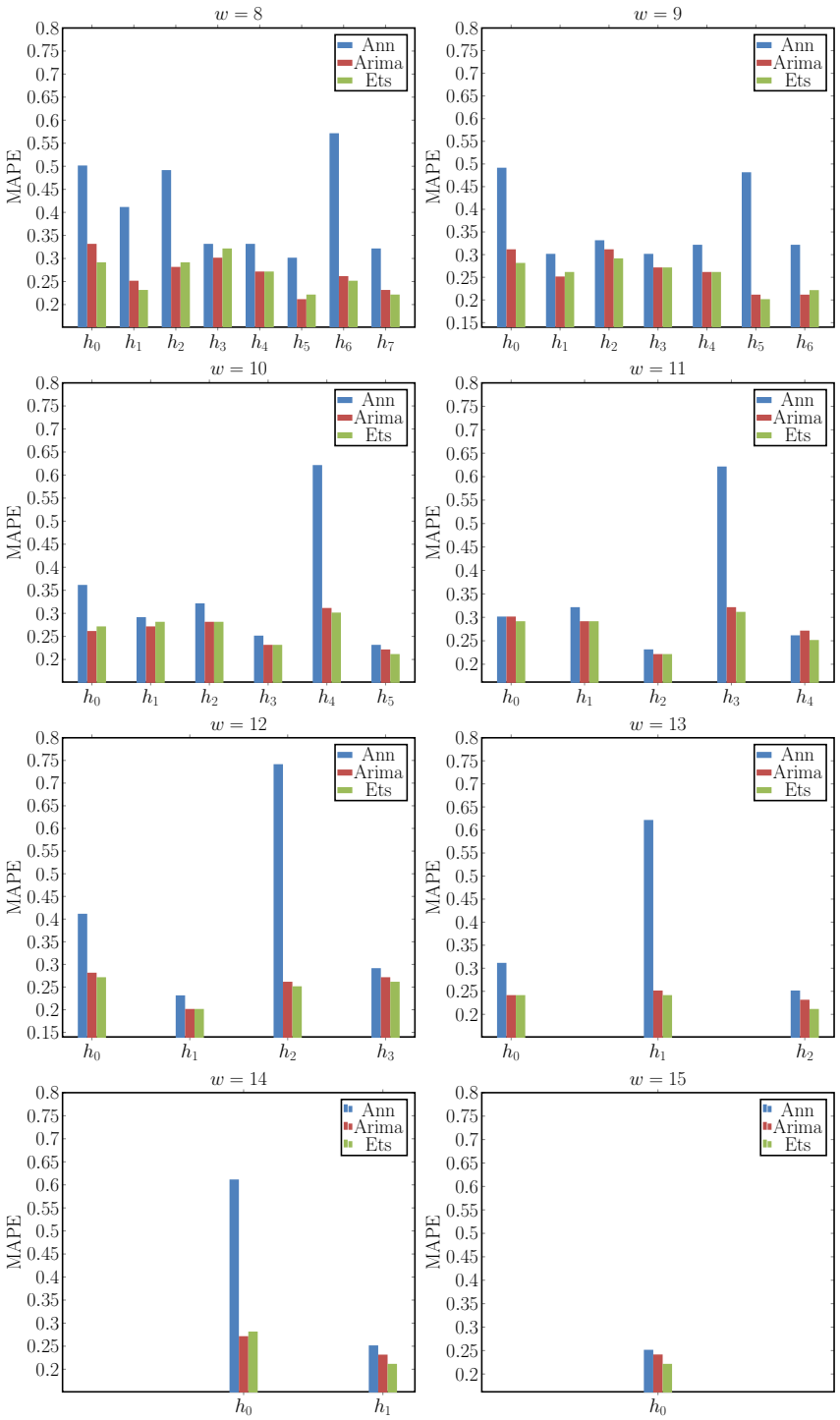

Figure 4.9 : hepPh-MAPE results.

The experimentation on both datasets revealed that the labels predicted by our framework excessively overlaps with the actual event labels. Time series models estimate the community features successfully. The models can be rank by ascending

order with regard to the error rate as follows: ETS, ARIMA, ANN. An inverse correlation between MAPE and F-measure results has been observed. After all, we can proudly say that the proposed framework is robust to the error rate of time series forecasting. More clearly, any abrupt change, rise/fall in time series model estimation does not evenly reflect to event prediction. It is also worth to mention that the results of the current study demonstrated that computing a less amount of past may yield better results than analyzing the whole history in a dynamic network.

5. FEATURE IDENTIFICATION FOR PREDICTING COMMUNITY EVOLUTION

This chapter introduces the Feature Identification for Event Prediction (FIEP) framework which is proposed to detect the most representative set of features for a given network before starting the community evolution process. An experimental system is designed to determine a community feature subset that results in higher (or at least the same) community event prediction accuracies than using all features.

5.1 Problem Formulation

This section briefly introduces the problem definition and formulation as well as the Feature Identification for Event Prediction (FIEP) framework. The framework concerned with identifying the appropriate feature set in predicting the evolutionary dynamics of a sequence of graphs $G = \{G^1, ..., G^T\}$ where $G^t = (V^t, E^t)$ denotes a graph containing the set of vertices and their interactions up to a particular snapshot t. FIEP framework takes the graph snapshots as input and produces event prediction results. The FIEP framework is comprised of two layers: Feature Identification Layer and Event Prediction Layer. Feature Identification Layer gets the first snapshot G^1 of the graph G and outputs the identified community feature subset. Event Prediction Layer obtains the graph G and produces prediction results utilizing the identified feature set generated by the feature identification layer. The notations used throughout the section are listed in Table 5.1.

Overall FIEP framework is represented in Figure 5.1, which is described in Algorithm 3. Lines 1-2 pertain to Feature Identification Layer while Lines 3-12 concern the Event Prediction Layer in Figure 5.1. Structural Network Analysis component corresponds to Line 1 while Community Feature Identifier component corresponds to Line 2. Event Prediction Layer comprises the following components respectively: Community Detection (Line 4), Feature Extraction (Line 6), Community Matching and Event Detection (Line 7-8), and Classification (Lines 9-12).

Table 5.1 : Notations and definitions.

Symbol	Description
$G = G^1, ..., G^T$	The graph sequence
T	The number of timestamps
G^t	The graph at time t
C^t	The communities at time t
C_i^t	The ith community at time t
$NM = \{NM_1, ..., NM_k\}$	The structural network measures
$F = \{f_1, ..., f_k\}$	The feature sequence
$F^i = \{f_1^i, ..., f_k^i\}$	The feature sequence of C^i
$IF = \{if_x, ..., ifz\}$	The identified feature set where $\mid IF \mid \leq r \; IF \subset F$
$SF = \{sf_x, ..., sfz\}$	The selected feature set where $\mid SF \mid \leq r \; SF \subset F$
$Sim(C_i^t, C_j^{t+1})$	Similarity of C_i^t and C_j^{t+1}
θ	*Similarity Threshold*
ϕ	*Fluctuation Threshold*

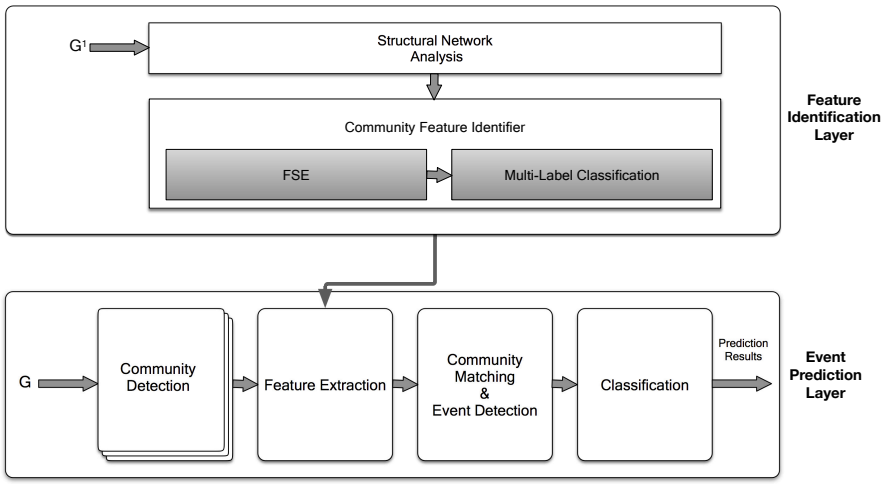

Figure 5.1 : Feature identification for event prediction (FIEP) framework.

In the first line of the Algorithm 3, the structural network measures (*NM*), such as clustering coefficient, average path length, embeddedness and betweenness are computed on the first graph snapshot where the details are given in Section 5.1.1.1. In the second line, community feature identifier process is applied to identify a feature subset (*IF*) by using structural network measures, and the detail of the process is provided in Section 5.1.1.2. The third and fourth lines consist of discovering communities of each graph snapshot using a community detection algorithm. Then for every community $C_i^t \in C^t$, the *IF* produced in the second step is calculated (Line 6). The corresponding community event is detected according to the similarity between

the community and its successor and the instance is created (Lines 7-9). Community detection and the community matching and event detection stage will be explained under the title FSE in Section 5.1.1.2. The last step of the algorithm is applying classifiers on the created instances to produce a prediction result (Line 12).

Algorithm 3 Feature identification for event prediction (FIEP) framework.

Input: A sequence of undirected and unweighted graphs: $G = G^1, ..., G^T$
Output: Prediction results

1: Compute structural network measures NM of G^1
2: Apply "**Community Feature Identifier**" process to extract $IF = \{if_x, ..., if_z\}$
3: **for** every graph G^t in the sequence **do**
4: Apply community detection algorithm to extract $C^t = \{C_1^t, C_2^t ...., C_l^t\}$
5: **for** every community $C_i^t \in C^t$ **do**
6: Calculate identified features $\{if(x)...if(z)\}$
7: Calculate $Sim(C_i^t, C_j^{t+1})$ with $C_j^{t+1} \in C^{t+1}$
8: Apply "**Event Detection**" procedure
9: Create an instance with the calculated community features and event label
10: **end for**
11: **end for**
12: Apply classifiers.

5.1.1 Feature identification layer

Feature identification layer has two main components: structural network analysis and community feature identifier. The objective of feature identifier layer is to build a model using a multi-label classifier on synthetic datasets which makes it possible to determine important subsets of community features of future networks in an automated way. Initially, *NM*, such as clustering coefficient, average path length, embeddedness and betweenness are computed on the first graph snapshot. Then the community feature identifier process is applied to extract *IF*.

5.1.1.1 Structural network analysis

Various measures of the network structure have been used allowing us to understand the structure of a particular network. As an example, Carrington *et al.* propose a list of some measures, including degree distribution, diameter and clustering coefficient [64]. Wasserman and Faust [65] used five properties of social network including betweenness centrality, degree, strength, closeness and clustering coefficient. However, not all of these measures provide information on the underlying structure

of the networks. Experiments were carried out on radius, diameter, closeness centrality, degree distribution coefficient, average path length, clustering coefficient, embeddedness and betweenness centrality measures. Since the experiments show that radius, diameter, closeness centrality and degree distribution coefficient do not yield discriminative and beneficial results in event prediction. The constitutional affinity and correlation between the structural metrics and community features might designate the discriminatory behavior of the structural metrics. The results of non-discriminatory metrics are not included for the sake of simplicity. As is known, different notions of community structures are implemented by numerous community detection algorithms. However, the ultimate aim of all of them is detecting groups with better internal connectivity than external connectivity. Whichever algorithm is used, the resulting communities are actual sub-graphs of the studied network. Therefore, their statistical properties reflect the mesoscopic organization of networks and this organization is similar between the networks with similar characteristics. Moreover, link densities within communities depend strongly on the topology of the network [66].

In this study, some of the connection oriented topological measures that are conducive to characterizing community structures are considered. The following measures have been investigated: clustering coefficient, average path length, embeddedness and betweenness centrality.

First examined measure is *clustering coefficient* (NM_1), which quantifies how densely the neighbourhood of a node is connected. It is a measure of the degree to which the nodes in a graph tend to cluster together. The clustering coefficient of a node i is the fraction of pairs of i's neighbour nodes that are connected to each other by edges [67]. The clustering coefficient of a node also represents how well connected its neighbours are. The local clustering coefficient for undirected graphs can be defined as:

$$CC_i = \frac{2\delta_i}{d_i(d_{i-1})} \tag{5.1}$$

where d_i is the degree of node i and δ_i is the number of triangles containing that node i. The clustering coefficient of a network is the mean clustering coefficient of all nodes:

$$CC_N = \frac{1}{N}\sum_i C_i \tag{5.2}$$

In a tightly connected network, the clustering coefficient approaches 1. Greater clustering coefficient is one of two important characteristics of the "small-world" networks revealed by Watts and Strogatz [67].

The other characteristic of "small-world" networks is smaller path length. *Average path length* (NM_2) of the network is defined as the average minimal distance between all pairs of its nodes. Let $\Delta(i, j)$ denote the shortest distance between node i and node j, the average path length of the network l_G is:

$$l_G = \frac{\sum_{i \neq j} \Delta(i, j)}{N(N-1)} \tag{5.3}$$

Smaller path length is the other characteristic of "small-world" networks [67]. A shorter average path length is an indicator of how close the nodes are one to another and enables the quick transfer of information within the network.

The other measure is *embeddedness* (NM_3), an important measure of social networks [68]. It refers to a node's relative involvement depth in social relations. The embeddedness of an edge $e = (i, j)$ in a network $G(V, E)$, named as $d_G(e)$, is defined to be the number of common neighbors of i and j. For an edge e, the subnetwork consisting of the nodes i, j and their common neighbors $d_G(e)$ is called $d - triangle$ [69]. The embeddedness value of an edge is high if two nodes adjacent on the edge have a high overlap of neighborhoods. Sometimes, embeddedness notion is defined under a different name called tie strength and within highly relationally embedded networks, the more nodes are integrated in dense clusters.

Lastly, *betweenness* (NM_4) centrality is computed, which is the most widely used metric to measure the importance of a node in a network. Betweenness counts the number of shortest paths in a network that passes through a node and takes into account the connectivity of the node's neighbors by giving a higher value for nodes which bridge clusters. It can be represented as:

$$B(i) = \sum_{i \neq j \neq k \in V} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \tag{5.4}$$

where σ_{jk} is total number of shortest paths from node j to node k and $\sigma_{jk}(i)$ is the number of those paths that pass through i. Nodes that occur on many shortest paths between other vertices have higher betweenness and play an important role in communication within the network [70].

5.1.1.2 Community feature identifier

Community feature identifier process aims at producing identified feature set *IF* by using *NM* of the first snapshot of the network. Initially, a correlation should be found between the structural network measures and the community features. It is not accurate to generalize the correlation results using only four datasets. Since there is a limited number of real-world datasets, it makes sense to take advantage of synthetic datasets having different topologies. First, *NM* of the synthetic datasets was calculated. Then, FSE algorithm was applied, as given in Algorithm 4, on synthetic datasets to extract selected community feature set (SF). Each synthetic dataset forms a train instance in the training dataset with its structural measures as attributes and selected community features as class labels. In this dataset each instance may have more than one label, since each network with different characteristics yields a subset of useful community features. For this reason, a multi-label classifier is needed. Thus, synthetic datasets are used as training instances to learn community features *SF* i.e. they are used as multi-labels and build a classification model which is subsequently applied to real networks in extracting their important community features (IF).

- Feature Subset Extraction (FSE): Feature Subset Extraction (FSE) algorithm takes the graph snapshots G as input and outputs the selected subset of features (SF). FSE algorithm aims to explore a feature subset for a given network by utilizing various community detection and feature selection algorithms. An outline of the entire process is provided in Algorithm 4.

Algorithm 4 Feature subset extraction algorithm (FSE).

Input: A sequence of undirected and unweighted graphs: $G = G^1, ..., G^T$
Output: Selected feature set (SF)

for every graph G^t in the sequence **do**
  Apply community detection algorithm to extract $C^t = \{C_1^t, C_2^t ...., C_l^t\}$
  for every community $C_i^t \in C^t$ **do**
    Calculate community features $F^i = f_1^i, ..., f_k^i$
    Calculate $Sim(C_i^t, C_j^{t+1})$ with $C_j^{t+1} \in C^{t+1}$
    Apply "**Event Detection**" procedure
    Create an instance with the features and label with the corresponding event
  end for
end for
Apply feature selection methods to extract SF.

FSE algorithm begins by detecting communities for every graph G^t in the sequence. The set $C^t = \{C_1^t, C_2^t, ..C_l^t\}$ is denoted as the l number of communities detected at the tth snapshot. The community structural features are extracted by measuring the properties of the communities on a large scale in order to better detect future community events. The extracted features cover many properties of both the internal link structure and the external interaction of the community. The six community events of *growth*, *survive*, *shrink*, *merge*, *split* and *dissolve* were considered in order to capture the changes of a community. A community can *survive* if there exists a similar community in the next snapshot. It can *growth* by linking with new members or can *shrink* by unlinking with existing members. Also, it may *split* at a later snapshot if it fractures into multiple communities. In the case where there is no similar community at a later snapshot, then it is assumed that the community *dissolves*. Our event detection process involves two thresholds: θ and ϕ. Two communities that are discovered at consecutive snapshots are similar if their similarity exceeds a given similarity threshold $\theta \in [0,1]$. After a community has been discovered as surviving, it is necessary to detect if the community survives by growing or shrinking, which is done by checking the fluctuation rate ϕ. Since the purpose of FSE is identifying the most predictive subset of features for the networks, we utilized several feature selection methods. To sum up, FSE arises from four stages: (1) Community Detection, (2) Community Feature Extraction, (3) Community Matching and Event Detection, and (4) Feature Selection.

- Multi-label Classification: The experiments were performed in MEKA which is an open source tool for multi-label classification [71]. A common approach to multi-label classification is to perform problem transformation in which a multi-label classification problem is transformed into a single-label classification task. The most popular problem transformation method is the Binary Relevance (BR) method [72]. BR decomposes a multi-label classification problem into several distinct single-label binary classification problems, one problem for each label, such that each binary model is trained to predict the relevance of one of the labels. In this work, BR method with Multi-Class Classifier (MCC) as the base classifier is used to conduct the feature identification experiment and produce an identified feature set (IF).

5.1.2 Event prediction layer

Event prediction layer has four main stages: community detection, feature extraction, community matching and event detection, and classification. The first three stages are as mentioned in Section 4.1.2, Section 4.1.3 and Section 4.1.4. The only difference being that they are computed for *IF* rather than the entire set. At the last stage of the event prediction layer, the identified community features with the assigned event class label of a community constitute an instance and are used as the input parameter for the classifiers. The following well-known classifiers are adopted in order to ascertain corresponding event labels: Bagging (Base Classifier: REPTree), Bayes Net (Estimator: SimpleEstimator, Search Algorithm: K2), J48, Decision Tree, Decision Table (Search Algorithm: BestFirst), Nearest Neighbor (KNN) (K-value:3, Search Algorithm: LinearNNSearch), OneR, Random Forest, Random Tree and Simple CART classifiers [1].

5.2 Experimental Study

Several experimental studies are conducted both on real and synthetic datasets. The details of these datasets are given in Section 5.2.1. In Section 5.2.2, the experimental design is explained, along with the threshold settings obtained through exhaustive experimental analysis. In Section 5.2.3, the results are provided. First, FSE algorithm performed on real datasets by exploiting a number of community detection algorithms to demonstrate that prominent features vary across datasets independently of the used algorithm. Then, the FSE algorithm results on synthetic datasets is given to display the correlation between the selected community features and the network topology. Lastly, prediction and performance results of the FIEP framework are provided.

5.2.1 Datasets

Four distinct real datasets and forty synthetic datasets were subject to experimentation in this study.

5.2.1.1 Real datasets

[1] The WEKA Data Mining implementation of the classifiers [63]

The real-world datasets were Digg, Slashdot, Enron, and Internet Topology.

- Digg: Digg is the reply network of the social news web site which allows users to submit a web page for general consideration and to vote Web content up or down, called digging and burying, respectively. The dataset consisted of 30,398 nodes and 87,627 edges from August to September 2008 [73]. Each node in the network is a user of the website, and each directed edge denotes that one user replied to another.

- Slashdot: Slashdot is a popular web site that frequently publishes short news posts and allows its readers to comment on them. One year of activity on Slashdot and consisting of 140,778 comments about news posts written by 51,083 users has been used [74]. Nodes are users and edges are replies. The edges are started by the responding user and annotated with the timestamp of the reply.

- Enron: Enron is one of the largest public datasets of a corporate e-mail environment. Each node in the network is an e-mail address, and a timestamped edge represents an e-mail sent between two addresses. There exist 1,326,771 edges corresponding to individual e-mails sent between 84,716 e-mail addresses. There were 215,841 unique timestamps covering a period of approximately 4 years [75].

- Internet Topology: Internet Topology is the network of connections between autonomous systems of the Internet. Autonomous systems are the collections of connected IP routing prefixes controlled by independent network operators. Nodes represent the autonomous systems, and edges represent the connections between autonomous systems. The dataset has 22,084 nodes and 122,439 edges collected between January 1, 2004 and October 24, 2004 [76].

5.2.1.2 Synthetic datasets

This study makes use of the tool provided by Greene *et al.* [22] which is adapted from LFR benchmark dataset generator [77] for dynamic graphs. The LFR is a scalable model proposed by Lancichinetti *et al.* for generating static networks with embedded ground truth communities that closely resemble real-world graphs [78]. The studies in [79] [80] have also shown the superiority of LFR benchmark over several other benchmarks with regard to their realism. LFR graphs follow power-law distribution in both degree and community size, having a user-controlled set of parameters such

as node number (N), desired average (δ) and maximum degree (δ_{max}), maximum and minimum community size ($\mathbb{C}_{min}$ and $\mathbb{C}_{max}$), and mixing coefficient (μ). A mixing coefficient governs the fraction of edges that are between communities. The adapted version by Greene *et al.* provides additional parameters to specify the number of time steps (t) and the probability of a node switching community membership between time steps (ρ). Forty synthetic datasets were generated where the number of the nodes (N) varied between 1000 and 20,000 and the number of edges between 4657 and 183,341. Diverse parameter settings were employed in synthetic dataset generation to obtain networks with different structures and with the topological properties consensually considered to be present in real-world networks. The other parameter settings were varied as the following: t=10, δ :[3,25], δ_{max}=[15,3000], $\mathbb{C}_{min}$=[3,1000], $\mathbb{C}_{max}$=[20,2000], μ=[0.1,0.2] and ρ=[0.1,0.5].

5.2.2 Experimental configuration

The datasets were represented as nodes and relationships that connect nodes were represented as edges. Each entry in the dataset was annotated with the timestamp of the connection. Each dataset was divided into ten time frames. A network snapshot at time t contains all nodes and edges existing at the corresponding time or prior to t. Networks were taken to be undirected and unweighted. Within the scope of the framework, various community detection algorithms were utilized. However, for the sake of simplicity, Infomap is selected to present the prediction results of FIEP, since several papers have shown that Infomap is the best algorithm for LFR benchmarks [81] [82]. It also performs well among several community detection algorithms in real-world networks [83]. Furthermore, it was found that the community size distribution obtained by applying Infomap fits power-law in our studied networks, being very similar to those of the real graphs. The communities whose size was smaller than three members were ignored. This diminishment was performed intuitively, since a group of nodes that consist of less than three members does not form a community.

The cardinality of selected and identified feature subset r (Table 5.1) was set at four since it is not efficient to pick and compute more than four features among nine in terms of time and complexity. The other important parameters to be tuned were similarity threshold (θ) and fluctuation threshold (ϕ). The impact of the thresholds on

the evolution of the communities can be understood from the results in Section 4.2.2. which are presented along with the setting procedure for real datasets. The same procedure was also performed on synthetic datasets (results not shown here). In all experiments, the classifiers were evaluated with stratified 10-fold cross-validation approach. For more reliable results, the cross-validation procedure was executed 10 times for each classifier and dataset.

5.2.2.1 Threshold setting

Tuning the appropriate thresholds for the purpose of tracking the matching communities and determining the community events at consecutive time steps is one of the challenges in studying the community evolution. A low similarity threshold (θ) may lead to a significant number of matching communities, while high value of θ results in more dissolutions. Likewise, low value of fluctuation threshold (ϕ) results in a small amount of survival and an excessive number of grown and shrunken communities. The number of survive events increases as ϕ increases. In order to investigate the impact of the similarity threshold (θ) on the community evolution, experiments were conducted where θ was varied from 0.1 to 1 and ϕ fixed at 0.15, yielding the results as depicted in Figure 5.2.

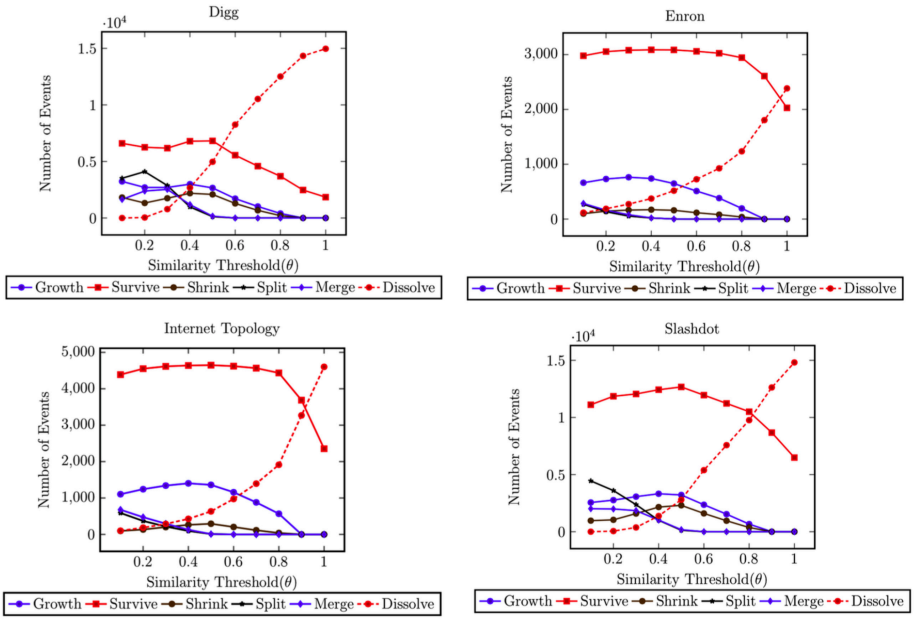

Figure 5.2 : Number of events vs similarity threshold (θ).

Figure 5.2 shows that the θ value has a noticeable effect on the observed events: the number of survive, growth, shrink, merge, split events drop as θ increases, while there

are more dissolve events. A low θ value induces a slew of matching communities thus enabling the observation of a significant number of events apart from dissolve. Besides, high values of θ result in a small amount of matching communities and events other than dissolve. Therefore, selecting the optimal similarity threshold is crucial and it differs with respect to the structural characteristics of the networks. We intended to select a threshold where the event numbers were as well-proportioned in their distribution as possible. The Digg and Slashdot datasets have more stable communities in which members participate over a long period warranting a value of $\theta = 0.5$. In Enron and Internet Topology datasets, communities can be more dynamic, since the number of matching communities begins to decrease after the similarity threshold (θ) reaches 0.4 and split and merge events are not observed. Hence, a rather low threshold of $\theta = 0.4$ is used to analyze the evolution of communities in these networks.

Fluctuation threshold (ϕ) is the other metric to be tuned and has significant effect on the survive, growth and shrink events. So as to determine the optimal ϕ value, experiments were conducted where ϕ was varied from 0.1 to 0.5 while fixing θ at 0.4, yielding the results as shown in Figure 5.3.

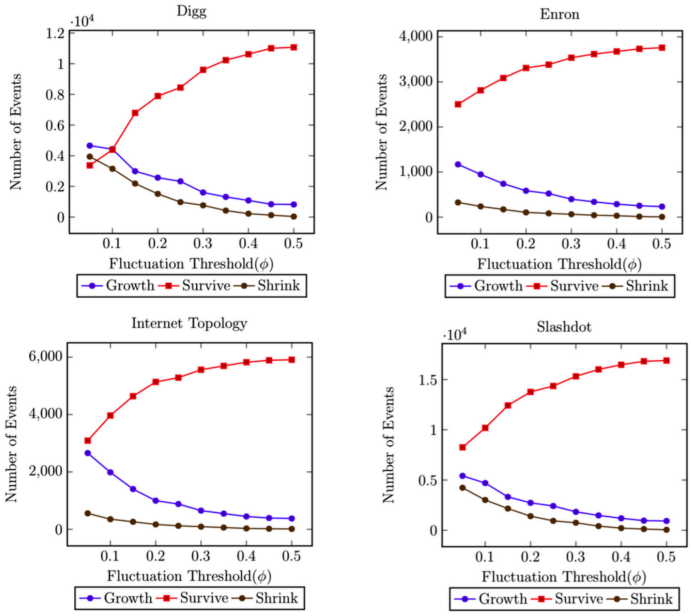

Figure 5.3 : Number of events vs fluctuation threshold (ϕ).

It can easily be observed that the number of survive events increases, whereas the number of growth and shrink events drops gradually as ϕ increases. In Enron and Internet Topology datasets, the gap between the observed event rates begins to grow

after $\phi = 0.1$. In Digg and Slashdot, balanced distribution can be attained with $\phi = 0.15$.

Thus, the choice of similarity and fluctuation thresholds is $\theta = 0.4, \phi = 0.1$ in Enron and Internet Topology, and $\theta = 0.5, \phi = 0.15$ in Digg and Slashdot, respectively. The event numbers regarding the selected thresholds in the networks are presented in Table 5.2.

Table 5.2 : Event numbers.

Dataset	Survive	Growth	Shrink	Merge	Split	Dissolve
Digg	6817	2661	2084	160	118	4967
Enron	2814	947	237	21	20	374
Internet Topology	3966	1986	353	128	96	427
Slashdot	12671	3217	2307	174	130	2804

Despite catching the utmost balanced distribution, a class imbalance exists due to the behavioral characteristics of social networks. Thus, to balance class labels and prevent over fitting, Resampling technique [2] is used to obtain uniform class distribution.

5.2.3 Results

5.2.3.1 Results of FSE on real datasets

In the first part of our experiments, we want to find out whether there exists a prominent subset of community features in real datasets that can be selected by feature selection methods for the purpose of effectively predicting community events. We want to reveal the prominent community feature subset which is independent of the used community detection algorithm, i.e. how common are the groups of features selected by different methods. The following process was applied: for each dataset and community detection algorithm, the entire set of feature selection methods were used. Then for each feature, the frequency of appearance in the selected subsets (SF) (those selected by feature selection methods) is computed. The frequency of selected features with each community detection algorithm is depicted in Figure 5.4. Features are represented in descending order depending on the frequency level. To better visualize the selection of the features, the columns of the heat-map are clustered to create blocks of similarly colored cells.

[2] A filtering approach available in Weka [63]

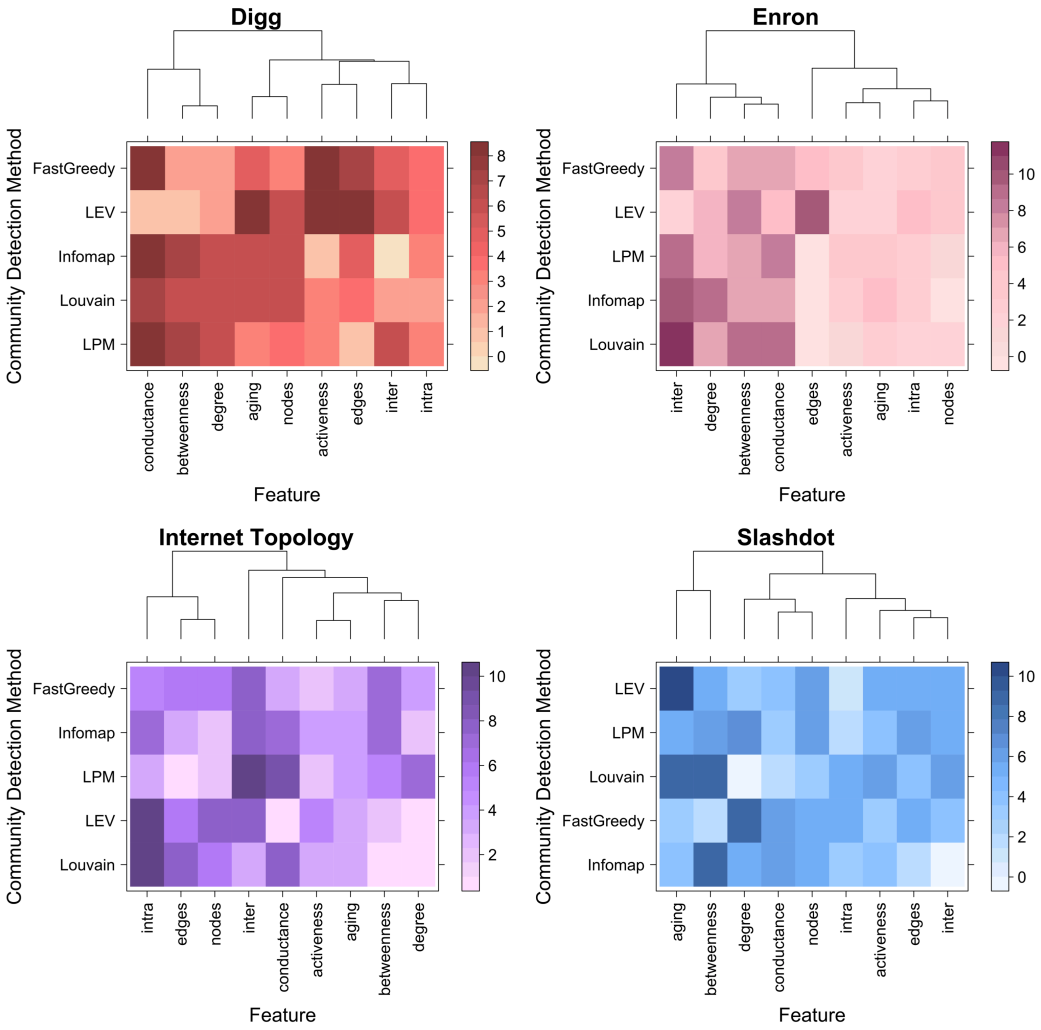

Figure 5.4 : Frequency of selected features with each community detection algorithm for all datasets.

Interesting results can be inferred from Figure 5.4. For example, in Digg: Infomap, Louvain and LPM give similar results, and conductance, aging, edges and betweenness features are stated as frequent. In Enron dataset: the entire set of the community detection algorithm results are similar except that LEV and inter, degree, betweenness and conductance are prominent features. In Internet Topology: Fast Greedy, LEV and Louvain algorithms yield similar results, and inter, edges and nodes are frequently observed. In Slashdot: LEV, LPM and Infomap algorithms are similar and aging, betweenness, degree and conductance emerged as the foremost features. As can be seen in Figure 5.4, for a given dataset the features selected by various community detection algorithms and feature selection methods are substantially matching, while there is considerable variation between the datasets themselves. Digg-Slashdot and Enron-Internet Topology pair results are fairly similar to each other.

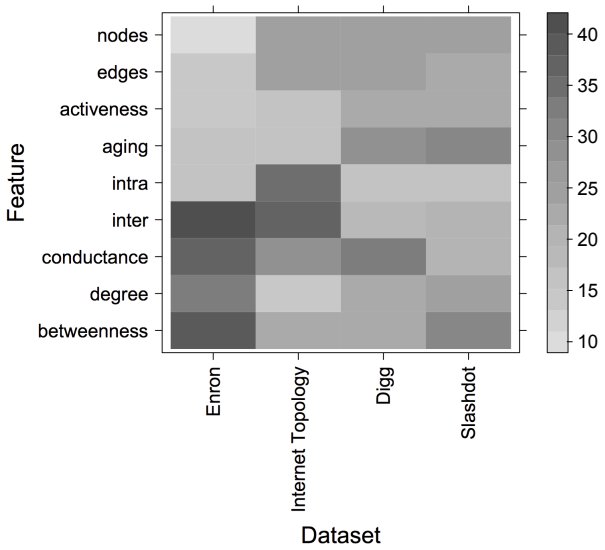

Figure 5.5 : Generalized frequency results.

Figure 5.5 shows the number of times that each feature is selected by each of the community detection algorithms for each dataset. The overall frequently selected features are as follows; Digg: conductance, aging, edges and betweenness; Enron: inter, betweenness, conductance and degree; Internet Topology: inter, intra, conductance and edges; Slashdot: aging, betweenness, nodes and degree. The generalized frequency results also verify that a different set of features comes to the forefront in each examined dataset. This raises some questions. Why has such diversity occurred? What are the distinguishing properties and underlying structures of these networks? So as to understand the ground of the difference, we need to analyze the structure of the networks and extract topological properties.

5.2.3.2 Results of FSE on synthetic datasets

As deduced in previous subsection (Section 5.2.3.1), the selected features are not identical in all networks. That being the case, how can we decide the feature set that will give us better/adequate community event prediction results in advance? With this intent, we make use of synthetic datasets with different characteristics. The motivation behind this is to investigate the discrepancy of the prominent features in distinct networks. Thus, to establish training data for multi-label classification and to investigate the correlation between the network topology and the prominent feature subsets, FSE algorithm (Algorithm 4) was performed on our synthetically generated networks possessing various topologies. The *NM* of the networks was calculated, all feature selection methods were applied, and the top four frequent features (*SF*) were revealed for each synthetic dataset.

Figure 5.6 shows the box-plots of structural measure values for the synthetic datasets covering all community features. To set an example, degree feature is frequently selected when the median value of the average path length is 3.85, while the upper quartile is 4.62 and lower quartile is 3.44.

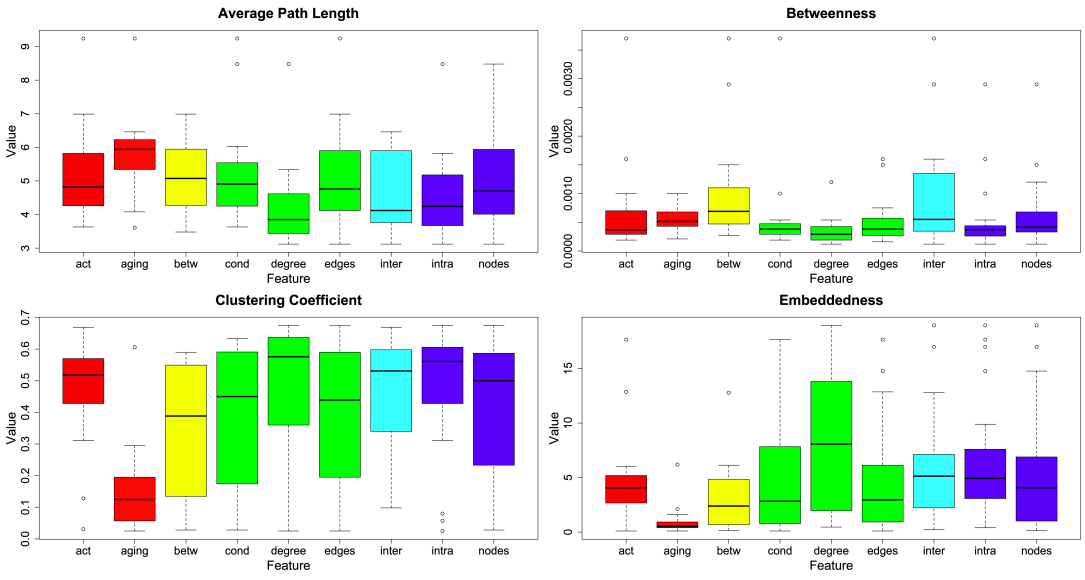

Figure 5.6 : Box-plots of structural networks measures over community features.

The results indicate that *inter*, *intra* and *degree* features are frequently selected when the synthetic datasets have a lower median value for average path length while *aging* feature is selected for higher average path length value. Betweenness centrality results

differ in the order of small decimal fractions. The greatest betweenness centrality rate was produced in selection of *betweenness* feature with 0.00069 median. Inversely, in case of low betweenness centrality rate, *conductance* and *intra* features are stated as prominent. Clustering coefficient value varies between 0.03 and 0.68 over the whole experiment. Higher clustering coefficient value yields the prominence of *activeness*, *degree*, *inter* and *intra* features, while smaller clustering coefficient results in the prominence of *aging*. *Degree*, *inter* and *intra* features are designated as prominent with the greater median embeddedness value. Smaller embeddedness reveals the selection of *aging* feature.

5.2.3.3 Performance and evaluation of FIEP

In order to analyze the performance of the FIEP framework, Algorithm 3 is conducted as indicated below:

Step 1. The real networks are converted into a sequence of network snapshots evolving over time (a total of 10 snapshots for each network).

Step 2. Structural measures belonging to the first snapshot of the networks have been calculated (Table 5.3).

Table 5.3 : Structural measures of the networks.

	Clustering Coef.	**Avg.Path Length**	**Embeddedness**	**Betweenness Centrality**
Digg	0.01	5.51	0.15	0.0006
Slashdot	0.05	5.50	0.48	0.0010
Enron	0.35	4.36	2.87	0.0004
Internet Topology	0.34	3.86	2.35	0.0002

Step 3. The model is trained using synthetic datasets with BR multi-label classification. MCC is applied as base classifier on the first snapshot of real networks. For each network, the subset of community features (IF) is identified.

Step 4. Infomap community detection algorithm is applied on each snapshot of the networks.

Step 5. IF found in Step 3 is calculated.

Step 6. Community matching and event detection processes are applied.

Step 7. Classifiers are used to testify the performance of the framework in terms of community event prediction.

In Step 3, *IF* for each dataset has been designated as follows: **Digg** (*conductance, betweenness, aging, nodes*), **Enron** (*inter, degree, betweenness*), **Internet Topology** (*inter, edges, degree*) and **Slashdot** (*aging, betweenness, degree, nodes*).

At this point, three basic questions arise. Why not use just one significant feature of a community? What are the benefits of using a set of community features? Do identified features really make a difference in community event prediction? In the approach proposed by Huang *et al.* only the activity features are used in measuring the influence of member activities to predict the network evolution [50]. Likewise, the GED method by Bródka *et al.* utilizes the community size feature alone to discover community evolution [29]. Hence, we also want to investigate whether a good event prediction accuracy can be obtained using only nodes [29] and activeness [50] features. The prediction performance of utilizing all features, FIEP framework, nodes [29] and activeness [50] in terms of accuracy, precision, recall, and F-measure using the top five accurate classifiers is shown in Table 5.4.

Table 5.4 reveals that overall FIEP framework F-measure results lie between 0.70 and 0.88, and accuracy results lie between 71.27% and 88.59% which can be considered as a successful event prediction rate. It has shown that the FIEP framework performs pretty much the same as using the full feature set and outperforms using only nodes [29] and activeness [50] by a wide margin in all datasets. Also, it is obvious from the table that usage of only nodes and activeness yield very poor prediction results. Also One-way Anova and Tukey HSD tests are performed with a significance level of $\alpha = 0.05$ to test whether the differences between the approaches are statistically significant or not. The test was performed on all the classifier results i.e. not just using the represented top five. It was found that there is not a significant difference between our FIEP framework and utilizing all features with $p = 0.99$. Besides, it is very clear that there is a significant difference between the FIEP and nodes/activeness with $p = 0.00$.

Table 5.4 : Prediction results of real datasets.

	Digg					Enron				
	Classifiers	**Accuracy**	**Precision**	**Recall**	**F-measure**	**Classifiers**	**Accuracy**	**Precision**	**Recall**	**F-measure**
All Features	J48	73.31	0.72	0.73	0.72	J48	84.17	0.83	0.84	0.83
	Random Forest	80.37	0.80	0.80	0.80	Random Forest	89.34	0.89	0.89	0.89
	Random Tree	78.40	0.77	0.78	0.78	Random Tree	87.27	0.87	0.87	0.87
	Bagging	72.49	0.71	0.72	0.71	Bagging	83.92	0.83	0.83	0.83
	Simple CART	72.62	0.71	0.72	0.72	Simple CART	82.92	0.82	0.82	0.82
FIEP	J48	72.53	0.72	0.73	0.72	J48	83.44	0.83	0.83	0.83
	Random Forest	79.79	0.79	0.80	0.79	Random Forest	88.59	0.88	0.89	0.88
	Random Tree	78.59	0.78	0.79	0.78	Random Tree	87.41	0.87	0.87	0.87
	Bagging	71.27	0.70	0.71	0.70	Bagging	81.70	0.81	0.82	0.81
	Simple CART	72.58	0.72	0.73	0.72	Simple CART	82.15	0.82	0.82	0.82
Nodes	J48	35.86	0.29	0.35	0.31	J48	40.81	0.39	0.40	0.36
	OneR	35.85	0.29	0.35	0.31	Random Forest	41.97	0.41	0.42	0.38
	KNN	35.83	0.29	0.35	0.31	Random Tree	41.88	0.41	0.41	0.38
	Random Forest	35.86	0.29	0.35	0.31	Bagging	41.42	0.42	0.41	0.35
	Random Tree	35.85	0.29	0.35	0.31	Simple CART	41.36	0.40	0.41	0.38
Activeness	J48	38.40	0.37	0.38	0.36	J48	59.02	0.57	0.59	0.57
	Random Forest	38.95	0.38	0.39	0.37	Random Forest	63.51	0.62	0.63	0.62
	Random Tree	38.81	0.38	0.38	0.36	Random Tree	63.53	0.62	0.63	0.62
	Bagging	38.10	0.36	0.38	0.36	Bagging	58.95	0.57	0.59	0.57
	Simple CART	38.31	0.37	0.38	0.36	Simple CART	60.86	0.60	0.60	0.59
	Internet Topology					**Slashdot**				
	Classifiers	**Accuracy**	**Precision**	**Recall**	**F-measure**	**Classifiers**	**Accuracy**	**Precision**	**Recall**	**F-measure**
All Features	J48	80.27	0.79	0.80	0.79	J48	76.79	0.76	0.76	0.76
	Random Forest	86.20	0.85	0.86	0.85	Random Forest	83.81	0.83	0.83	0.83
	Random Tree	84.71	0.83	0.84	0.84	Random Tree	82.17	0.81	0.82	0.81
	Bagging	79.55	0.78	0.79	0.78	Bagging	75.96	0.75	0.76	0.75
	Simple CART	78.18	0.76	0.78	0.77	Simple CART	75.47	0.74	0.75	0.75
FIEP	J48	78.47	0.77	0.78	0.77	J48	74.99	0.74	0.75	0.75
	Random Forest	84.57	0.84	0.85	0.84	Random Forest	83.05	0.83	0.83	0.83
	Random Tree	83.72	0.83	0.84	0.83	Random Tree	81.74	0.81	0.82	0.81
	Bagging	77.22	0.76	0.77	0.76	Bagging	74.11	0.73	0.74	0.73
	Simple CART	78.24	0.77	0.78	0.77	Simple CART	74.72	0.74	0.75	0.74
Nodes	J48	35.50	0.36	0.35	0.32	OneR	36.09	0.35	0.36	0.32
	Random Forest	35.30	0.35	0.35	0.32	Bayes Net	36.21	0.36	0.36	0.32
	Random Tree	35.18	0.35	0.35	0.32	Decision Table	36.18	0.35	0.36	0.32
	Bagging	34.85	0.35	0.35	0.31	Random Tree	36.09	0.35	0.36	0.32
	Simple CART	35.08	0.35	0.35	0.31	Simple CART	36.09	0.35	0.36	0.32
Activeness	J48	42.14	0.41	0.42	0.41	J48	40.67	0.39	0.40	0.38
	Random Forest	43.96	0.44	0.44	0.43	Random Forest	41.86	0.41	0.41	0.40
	Random Tree	43.65	0.43	0.43	0.42	Random Tree	41.60	0.40	0.41	0.39
	Bagging	41.65	0.41	0.41	0.41	Bagging	40.68	0.39	0.40	0.38
	Simple CART	42.19	0.42	0.42	0.41	Simple CART	41.08	0.40	0.41	0.39

The prediction results of FIEP versus the feature selection methods are presented in Table 5.5. J48 and Random Forest classifiers are selected to represent results in terms of F-measure. It is clear from Table 5.5 that FIEP is superior to the feature selection methods particularly in Digg and Internet Topology datasets. Only in Enron do some feature selection methods give better results than FIEP and even then only by a narrow margin (0.02 in J48, 0.03 in Random Forest). This difference is negligible when the increase in speedup is taken into account (see Figure 5.7). In Slashdot, none of the feature selection methods could overcome FIEP. When we compare the feature selection methods among themselves, despite the fact that there is not being any dramatically difference, WSE gave the worst and IGAE, ORAE and RFAE gave the best results in general.

Table 5.5 : Prediction results: FIEP vs feature selection methods.

	Digg		**Enron**		**Internet Topology**		**Slashdot**	
	J48	**Random Forest**	**J48**	**Random Forest**	**J48**	**Random Forest**	**J48**	**Random Forest**
FIEP	0.72	0.79	0.83	0.89	0.77	0.84	0.75	0.83
IGAE	0.70	0.78	0.84	0.89	0.69	0.79	0.75	0.82
CFS	0.66	0.76	0.85	0.92	0.70	0.83	0.69	0.80
CSAE	0.70	0.78	0.84	0.89	0.66	0.78	0.75	0.82
CSE	0.70	0.78	0.83	0.90	0.65	0.74	0.74	0.82
CAE	0.53	0.54	0.85	0.91	0.66	0.75	0.74	0.81
CVAE	0.67	0.73	0.84	0.91	0.67	0.82	0.71	0.79
GRAE	0.67	0.76	0.84	0.89	0.68	0.77	0.74	0.81
ORAE	0.70	0.78	0.84	0.89	0.69	0.82	0.74	0.82
RFAE	0.67	0.75	0.84	0.91	0.70	0.82	0.74	0.82
SUAE	0.70	0.78	0.84	0.89	0.69	0.79	0.74	0.81
WSE	0.50	0.53	0.84	0.89	0.65	0.74	0.74	0.82

Time spent predicting the community events is also registered. All experiments are performed on a PC with 2xIntel Xeon 4C CPU (2.4GHz) and 80GB of RAM. The programs are coded in Java without any code optimization. The run time of FIEP and use of all features is measured in seconds. Also, the improvement in the run time of FIEP framework versus using all features in terms of speedup is computed by dividing the run time of all features by the run time of FIEP framework. From Figure 5.7, it can be seen that the FIEP framework is faster, in comparison to using all features in all datasets, and achieves a speedup of 2.05, 4.54, 2.65, 1.83 times in Digg, Enron, Internet Topology and Slashot respectively.

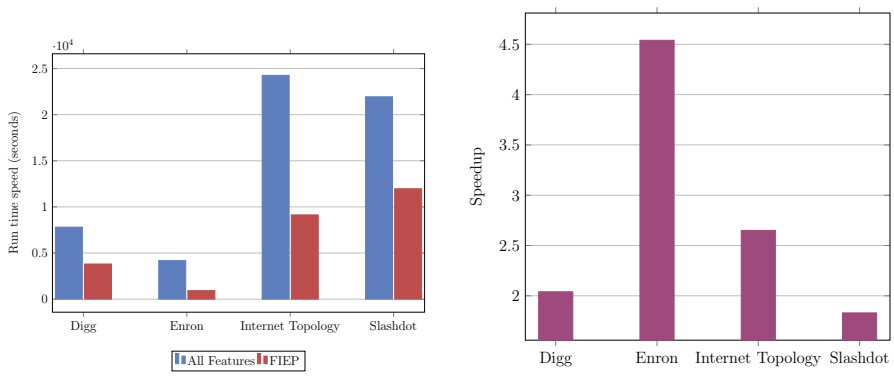

Figure 5.7 : (a) Run time of FIEP and all features in seconds (b) Run time speedup of FIEP over using all features

5.2.4 Discussions

In this subsection, the correlation between the prominent community features and the network topology is examined in perspective. It has been discovered that identified features of FIEP match up in the findings with Section 5.2.3.1. Thus, we can conclude that FIEP identifies a generic feature subset that produce successful event prediction results with reduced computation time. The first examined structural measure is clustering coefficient which quantifies how densely the neighborhood of a node is connected. A high clustering coefficient indicates the presence of triads in the network. High density of triads can be related to the existence of community structures. The clustering coefficients of Enron and Internet Topology networks are higher than for Slashdot and Digg networks, thus they constituting modular community structures when compared with Slashdot and Digg. Recall that *inter*, *intra*, *degree* and *activeness* features are prominent at higher clustering coefficient value while *aging* is selected when clustering coefficient value is low (Figure 5.6). As expected, FIEP identifies *inter* and *degree* features in Enron and Internet Topology and *aging* feature in Digg and Slashdot.

The networks were also analyzed to quantify the strength of community structure. A network has a clear-cut community structure if it is divided naturally into groups of nodes with dense connections within the groups and sparser connections between the different groups. To measure this γ_i is defined as the ratio of total number of inter

$(e_i^t(out))$ and intra $(e_i^t(in))$ links of a community i:

$$\gamma(i) = \frac{e_i^t(out)}{e_i^t(in)} \tag{5.5}$$

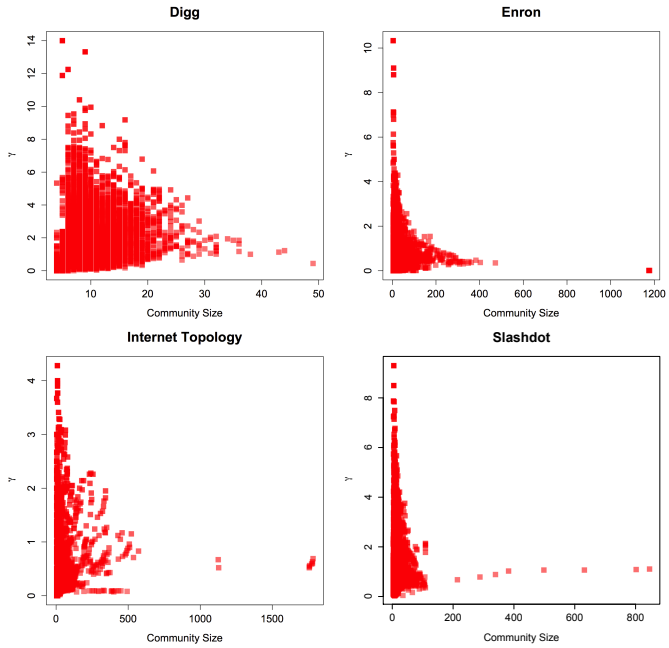

Figure 5.8 : The distribution of γ_i over community size

Figure 5.8 shows the distribution of γ_i as a function of community size. With respect to this measure, Enron and Internet Topology datasets possess clear-cut community structure: the majority of communities have low γ values. Predictably, if a node is in a dense community, its clustering coefficient is expected to be high. It is also clear from the figure that Enron and Internet Topology datasets have a more coarse-grained community structure where the graph is partitioned into larger communities, while Digg and Slashdot have fine-grained communities. It can be deduced that as the networks get closer to "small-world" effect and represent modular community structure they might have better community event prediction results with *inter*, *intra* and *degree* features.

The other measure to be examined is average path length which is simply the average path of all pairs of shortest paths in a social network. Enron and Internet Topology networks have smaller average path length (Table 5.3). Bearing in mind that *inter*, *intra degree* features are prominent in smaller average path length, as in Enron and Internet Topology, and *aging* is the prominent feature in greater average path length, as in Digg and Slashdot (See Figure 5.6). The results indicate that Enron and Internet

Topology networks fit the two patterns of "small-world" phenomena better than Digg and Slashdot, with a larger clustering coefficient and a smaller average shortest path length.

The third measure tested was embeddedness, a notion used to capture the strength of an edge with regard to the number of common neighbors. The link densities are higher in networks with higher embeddedness values. From Figure 5.6, we can easily see that embeddedness value produces results correlatively with clustering coefficient and average path length; in other words *inter*, *intra* and *degree* features are selected in case of greater embeddedness value and *aging* feature is selected in case of lower embeddedness value.

Not all nodes are equally important in a network. Centrality analysis is performed to find out the most important nodes. In particular, betweenness centrality of nodes was studied, being one of the most frequently considered centrality indices. Node betweenness quantifies the extent to which a particular node is positioned between communities. Nodes connecting distinct communities have very high betweenness. As with the correlation of degree and betweenness centrality measures in social networks, *degree* and *betweenness* features of our framework were correlated and selected together (Figure 5.4). In Figure 5.6, it can be observed that *betweenness* and *inter* features are prominent with greater betweenness value. We have already seen that, *betweenness* feature is one of the identified features of Digg and Slashdot which are the datasets having the greatest betweenness centrality measure.

In brief, the observations can be summed up in the list below:

- The networks which ensure "small-world" property and have strong community structure, produce better event prediction results with *inter*, *intra* and *degree* features.
- Similar patterns are observed with embeddedness, i.e. a higher embeddedness corresponds to the prominence of *inter*, *intra* and *degree*.
- *Aging* feature is prominent in more stable and less embedded networks.
- *Betweenness* feature is frequently selected in the networks that have higher node betweenness value.

- *Betweenness* and *degree* features are strongly correlated and observed together.
- *Nodes* and *edges* features do not present discriminative behavior within the framework.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

In the first phase of the thesis an approach has been proposed to predict next event of a community through time series analysis on the structural and temporal feature history of the community. Differently from the classical event prediction approaches that takes the network prior to time t in order to predict events at future time t', our method takes a pre-defined length of temporal information into account by monitoring how the network evolved along time. An empirical comparative evaluation of the performance of forecasting models ANN, ARIMA and ETS was employed for predicting the next values of the features and attendantly the events of the communities. Experiments are conducted on two different windowing approach, namely landmark and sliding. Various window lengths are experimented to testify framework performance in terms of error rate (MAPE) of community feature forecasting and rate of overlap in terms of F-measure between the predicted and actual events.

The obtained results suggest that proposed event detection procedure accurately predicts next event of the communities. Moreover, usage of time series analysis model results highly match up with the actual event labels. Additionally, for the forecasting of community features, acceptable error rate is produced specifically by ARIMA and ETS models. The ANN model was often worse in forecasting accuracy among all the models we examined. We have also investigated that there exist remarkable correlation between error rates and match up values. However, the framework is not that much sensitive to the MAPE values. Precisely, the poor results in MAPE values do not trigger a sharp fall in matching accuracy. Moreover, we have shown the effect of processed window length on the results. One may think that longer window lengths may improve forecasting accuracy. Contrarily, even though the existence of the fluctuative results, experiments revealed that sliding window approach performed better than landmark, in particular with shorter window lengths.

In the second phase, a novel framework, namely FIEP has been proposed to identify a feature set that provides better results in community event prediction among a broad range of community features by exploiting the network's topological properties. Initially, various community detection algorithms with feature selection methods were performed on real datasets to obtain a generalized projection on the results, showing that a diverse set of community features is frequent in distinct networks. Our findings uncovered that distinct features are extracted as prominent in networks with distinct topologies. To investigate the ground of the difference, detailed structural network analysis was performed. Subsequently, the correlation between the network topology and the prominent community features was investigated on synthetically generated datasets. By multi-label classification, where the synthetic dataset results were used as the training data, representative community features of the real datasets were extracted. In this way, the most representative community features (which are initially unknown) were identified so that grown, shrunken, survived, merged, split, and dissolved communities could be predicted with greater accuracy. The experiments indicated that FIEP produces almost the same prediction results as those produced using the entire feature set, such that there is no statistical difference. Due to the lower number of features that should be calculated, there is a corresponding reduction in computational time and cost. We have also presented results for the running time and speedup of FIEP framework over the use of all features.

6.2 Future Work

For the first phase, an open problem is how many window lengths that users should employ in forecasting. A solution may be using validation set to identify the required window length that perform good results. In its current state, our study can predict the values of the community events at a future time step. In future research, we intend to extend the prediction horizon from one timestamp to more by preserving the accuracy. Furthermore, we will deeply investigate the underlying structural properties that yield fluctuation of the results as the window length changes.

Another future direction is to predict the time of change and estimate a possible break point time in community structure by using the general structural properties of the network and evolving community metrics. Routinely, the community detection

algorithm is applied without examining whether any considerable change occurs in the network. By predicting the time of change, the community detection algorithm will not be executed unnecessarily and the time and space complexity will be reduced.

REFERENCES

[1] **Girvan, M. and Newman, M.E.J.** (2002). Community structure in social and biological networks, *Proceedings of the National Academy of Sciences*, *99*(12), 7821–7826.

[2] **Wasserman, S. and Faust, K.** (1994). *Social network analysis: Methods and applications*, volume 8, Cambridge university press.

[3] **Fortunato, S.** (2010). Community detection in graphs, *Physics Reports*, *486*(3-5), 75 – 174.

[4] **Hopcroft, J.**, **Khan, O.**, **Kulis, B. and Selman, B.** (2004). Tracking evolving communities in large linked networks, *Proceedings of the National Academy of Sciences*, *101*, 5249–5253.

[5] **Berger-Wolf, T.Y. and Saia, J.** (2006). A Framework for Analysis of Dynamic Social Networks, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, ACM, New York, NY, USA, pp.523–528.

[6] **Tantipathananandh, C.**, **Berger-Wolf, T. and Kempe, D.** (2007). A Framework for Community Identification in Dynamic Social Networks, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, ACM, New York, NY, USA, pp.717–726.

[7] **Newman, M.E.J. and Park, J.** (2003). Why social networks are different from other types of networks, *Phys. Rev. E*, *68*(3), 036122.

[8] **Palla, G.**, **Barabasi, A.L. and Vicsek, T.** (2007). Quantifying social group evolution, *Nature*, *446*(7136), 664–667.

[9] **Asur, S.**, **Parthasarathy, S. and Ucar, D.** (2009). An event-based framework for characterizing the evolutionary behavior of interaction graphs, *ACM Trans. Knowl. Discov. Data*, *3*(4), 16:1–16:36.

[10] **Chen, Z.**, **Hendrix, W. and Samatova, N.F.** (2012). Community-based Anomaly Detection in Evolutionary Networks, *J. Intell. Inf. Syst.*, *39*(1), 59–85.

[11] **Gliwa, B.**, **Bródka, P.**, **Zygmunt, A.**, **Saganowski, S.**, **Kazienko, P. and Kozlak, J.** (2013). Different approaches to community evolution prediction in blogosphere., *ASONAM*, ACM, pp.1291–1298.

[12] **İlhan, N. and Şule Gündüz Öğüdücü** (2013). Community Event Prediction in Dynamic Social Networks, *Machine Learning and Applications*, *2*, 269–274.

[13] **Backstrom, L.**, **Huttenlocher, D.**, **Kleinberg, J. and Lan, X.** (2006). Group Formation in Large Social Networks: Membership, Growth, and Evolution, *Proceedings of KDD'06*.

[14] **Leskovec, J.**, **Kleinberg, J. and Faloutsos, C.** (2005). Graphs over time: densification laws, shrinking diameters and possible explanations, *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, ACM, New York, NY, USA, pp.177–187.

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

[16] **Takaffoli, M.**, **Rabbany, R. and Zaïane, O.R.** (2014). Community evolution prediction in dynamic social networks., ***X. Wu, M. Ester and G. Xu,*** *editors, ASONAM*, IEEE, pp.9–16.

[17] **Berger-wolf, T.Y.** (2006). A framework for analysis of dynamic social networks, *DIMACS Technical Report*, ACM Press, pp.523–528.

[18] **Tantipathananandh, C.**, **Berger-Wolf, T. and Kempe, D.** (2007). A framework for community identification in dynamic social networks, *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, ACM, New York, NY, USA, pp.717–726.

[19] **Fagnan, J.**, **Rabbany, R.**, **Takaffoli, M.**, **Verbeek, E. and Zaïane, O.R.** (2014). Community Dynamics: Event and Role Analysis in Social Network Analysis, *Advanced Data Mining and Applications - 10th International Conference, ADMA 2014, Guilin, China, December 19-21, 2014. Proceedings*, pp.85–97.

[20] **Palla, G.**, **Derenyi, I.**, **Farkas, I. and Vicsek, T.** (2005). Uncovering the overlapping community structure of complex networks in nature and society, *Nature*, *435*, 814–818.

[21] **Wang, Y.** (2008). Community Evolution of Social Network: Feature, Algorithm and Model, *Preprint-arxiv*, *0804.4356*.

[22] **Greene, D.**, **Doyle, D. and Cunningham, P.** (2010). Tracking the Evolution of Communities in Dynamic Social Networks., *ASONAM*, IEEE Computer Society, pp.176–183.

[23] **Tajeuna, E.G.**, **Bouguessa, M. and Wang, S.** (2015). Tracking the evolution of community structures in time-evolving social networks, *2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, October 19-21, 2015*, IEEE, pp.1–10.

[24] **Ahn, Y.Y.**, **Han, S.**, **Kwak, H.**, **Moon, S. and Jeong, H.** (2007). Analysis of Topological Characteristics of Huge Online Social Networking Services,

Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, New York, NY, USA, pp.835–844.

[25] **Goldberg, M.K.**, **Magdon-Ismail, M.**, **Nambirajan, S. and Thompson, J.** (2011). Tracking and Predicting Evolution of Social Communities., IEEE, pp.780–783.

[26] **Kairam, S.R.**, **Wang, D.J. and Leskovec, J.** (2012). The Life and Death of Online Groups: Predicting Group Growth and Longevity, *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, ACM, New York, NY, USA, pp.673–682.

[27] **Patil, A.**, **Liu, J. and Gao, J.** (2013). Predicting Group Stability in Online Social Networks, *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, ACM, New York, NY, USA, pp.1021–1030.

[28] **Diakidis, G.**, **Karna, D.**, **Fasarakis-Hilliard, D.**, **Vogiatzis, D. and Paliouras, G.** (2015). Predicting the Evolution of Communities in Social Networks, *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS '15, ACM, New York, NY, USA, pp.1:1–1:6.

[29] **Bródka, P.**, **Kazienko, P. and Koloszczyk, B.** (2012). Predicting Group Evolution in the Social Network., *SocInfo*, volume7710 of *Lecture Notes in Computer Science*, Springer, pp.54–67.

[30] **Saganowski, S.**, **Gliwa, B.**, **Bródka, P.**, **Zygmunt, A.**, **Kazienko, P. and Kozlak, J.** (2015). Predicting Community Evolution in Social Networks, *Entropy*, *17*(5), 3053.

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

[32] **da Silva Soares, P.R. and Prudêncio, R.B.C.** (2012). Time Series Based Link Prediction., *IJCNN*, IEEE, pp.1–7.

[33] **OĆonnor, B.**, **Balasubramanyan, R.**, **Routledge, B.R. and Smith, N.A.** (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series., *ICWSM*, The AAAI Press.

[34] **İlhan, N. and Öğüdücü, c.G.** (2015). Predicting Community Evolution Based on Time Series Modeling, *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, ACM, New York, NY, USA, pp.1509–1516.

[35] **Guyon, I. and Elisseeff, A.** (2003). An Introduction to Variable and Feature Selection, *The Journal of Machine Learning Research*, *3*, 1157–1182.

[36] **Jain, A.K. and Zongker, D.E.** (1997). Feature Selection: Evaluation, Application, and Small Sample Performance., *IEEE Trans. Pattern Anal. Mach. Intell.*, *19*(2), 153–158.

[37] **Siedlecki, W. and Sklansky, J.** (1993). *Handbook of Pattern Recognition and Computer Vision*, World Scientific Publishing Co., Inc., River Edge, NJ, USA.

[38] **Tabakhi, S.**, **Moradi, P. and Akhlaghian, F.** (2014). An unsupervised feature selection algorithm based on ant colony optimization, *Engineering Applications of Artificial Intelligence*, *32*, 112 – 123.

[39] **Yu, L. and Liu, H.** (2003). Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution., *ICML*, AAAI Press, pp.856–863.

[40] **Dy, J.G. and Brodley, C.E.** (2000). Feature Subset Selection and Order Identification for Unsupervised Learning, *In Proc. 17th International Conf. on Machine Learning*, Morgan Kaufmann, pp.247–254.

[41] **Quinlan, J.R.** (1993). *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[42] **Mitchell, T.M.** (1997). *Machine learning*, McGraw-Hill.

[43] **Holte, R.** (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning*, *11*(1), 63–90.

[44] **Kononenko, I.** (1994). Estimating Attributes: Analysis and Extensions of RELIEF., ***F. Bergadano and L.D. Raedt****, editors, ECML*, volume784 of *Lecture Notes in Computer Science*, Springer, pp.171–182.

[45] **Press, W.**, **Teukolsky, S.**, **Vetterling, W. and Flannery, B.** (1995). *Numerical Recipes in C – the Art of Scientific Computing*, Cambridge University Press, U.K.

[46] **Hall, M.A.** (2000). Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning., *ICML*, Morgan Kaufmann, pp.359–366.

[47] **Kohavi, R. and John, G.H.** (1997). Wrappers for Feature Subset Selection, *Artificial Intelligence*, *97*(1-2), 273–324.

[48] **Jirapech-Umpai, T. and Aitken, J.S.** (2005). Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes., *BMC Bioinformatics*, *6*, 148.

[49] **Díaz-Uriarte, R. and Alvarez de Andrés, S.** (2006). Gene selection and classification of microarray data using random forest, *BMC bioinformatics*, *7*(1), 1–13.

[50] **Huang, S. and Lee, D.** (2011). Exploring Activity Features in Predicting Social Network Evolution, *Machine Learning and Applications*, *2*, 269–274.

[51] **Clauset, A.**, **Newman, M.E.J.**, **and Moore, C.** (2004). Finding community structure in very large networks, *Physical Review E*, 1– 6.

[52] **Newman, M.E.J.** (2006). Finding community structure in networks using the eigenvectors of matrices, *Physical review E*, *74*(3).

[53] **Rosvall, M. and Bergstrom, C.T.** (2008). Maps of random walks on complex networks reveal community structure, *Proceedings of the National Academy of Sciences*, *105*(4), 1118–1123.

[54] **Raghavan, U.N.**, **Albert, R. and Kumara, S.** (2007). Near linear time algorithm to detect community structures in large-scale networks, *Physical Review E*, *76*(3).

[55] **Blondel, V.D.**, **Guillaume, J.L.**, **Lambiotte, R. and Lefebvre, E.** (2008). Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008.

[56] **Pearson, K.** (1895). Note on regression and inheritance in the case of two parents, *Proceedings of the Royal Society*, *58*, 240–242.

[57] **Quinlan, J.** (1986). Induction of decision trees, *Journal of Machine Learning*, *1*, 81–106.

[58] **Box, G.E. and Jenkins, G.M.** (1968). Some recent advances in forecasting and control, *Applied Statistics*, 91–109.

[59] **Hill, T.**, **O'Connor, M. and Remus, W.** (1996). Neural Network Models for Time Series Forecasts, *Management Science*, *42*(7), 1082–1092.

[60] **Zhang, G.**, **Patuwo, B.E. and Hu, M.Y.** (1998). Forecasting with artificial neural networks:The state of the art, *International Journal of Forecasting*, *14*(1), 35–62.

[61] **Georgakarakos, S.**, **Koutsoubas, D. and Valavanis, V.** (2006). Time series analysis and forecasting techniques applied on loliginid and ommastrephid landings in Greek waters, *Fisheries Research*, *78*(1), 55 – 71.

[62] **Wheelwright, S. and Makridakis, S.** (1973). *Forecasting methods for management*, Wiley series on systems and controls for financial management, Wiley, New York u.a.

[63] **Hall, M.**, **Frank, E.**, **Holmes, G.**, **Pfahringer, B.**, **Reutemann, P. and Witten, I.H.** (2009). The WEKA data mining software: an update, *SIGKDD Explor. Newsl.*, *11*(1), 10–18.

[64] **Carrington, P.J.**, **Scott, J. and Wasserman, S.** (2005). *Models and methods in social network analysis*, Cambridge Univ Pr.

[65] **Wasserman, S. and Faust, K.** (1994). *Social Network Analysis: Methods and Applications*, Cambridge University Press.

[66] **Lancichinetti, A.**, **Kivelä, M.**, **Saramäki, J. and Fortunato, S.** (2010). Characterizing the Community Structure of Complex Networks, *PLoS ONE*, *5*(8), 1–8.

[67] **Watts, D.J. and Strogatz, S.H.** (1998). Collective dynamics of'small-world'networks., *Nature*, *393*(6684), 409–10.

[68] **Moody, J. and White, D.R.** (2000). Structural Cohesion and Embeddedness: A hierarchical conception of social groups., *American Sociological Review*, *68*, 103–127.

[69] **Sridharan, A.**, **Gao, Y.**, **Wu, K. and Nastos, J.** (2010). Statistical Behavior of Embeddedness and Communities of Overlapping Cliques in Online Social Networks, *CoRR*, *abs/1009.1686*.

[70] **Freeman, L.C.** (1978). Centrality in social networks conceptual clarification, *Social Networks*, 215.

[71] **Read, J.**, **Bifet, A.**, **Holmes, G. and Pfahringer, B.** (2012). Scalable and efficient multi-label classification for evolving data streams, *Machine Learning*, *88*(1-2), 243–272.

[72] **Tsoumakas, G. and Katakis, I.** (2007). Multi-label classification: An overview, *International Journal of Data Warehousing and Mining*, *3*, 1–13.

[73] **Choudhury, M.D.**, **Sundaram, H.**, **John, A. and Seligmann, D.D.** (2009). Social Synchrony: Predicting Mimicry of User Actions in Online Social Media, *Proc. Int. Conf. on Computational Science and Engineering*, pp.151–158.

[74] **Gómez, V.**, **Kaltenbrunner, A. and López, V.** (2008). Statistical analysis of the social network and discussion threads in slashdot, *WWW '08: Proceeding of the 17th international conference on World Wide Web*, ACM, New York, NY, USA, pp.645–654.

[75] **Lahiri, M. and Cebrian, M.** (2010). The genetic algorithm as a general diffusion model for social networks, *Proc. of the 24th AAAI Conference on Artificial Intelligence*, AAAI Press.

[76] **Zhang, B.**, **Liu, R.A. and Massey, Daniel, L.Z.** (2004). Collecting the internet AS-level topology., *Computer Communication Review*, *35*(1), 53–61.

[77] **Lancichinetti, A.**, **Fortunato, S. and Radicchi, F.** (2008). Benchmark graphs for testing community detection algorithms, *Phys. Rev. E*, *78*(4), 046110.

[78] **Li, Y.**, **He, K.**, **Bindel, D. and Hopcroft, J.E.** (2015). Uncovering the Small Community Structure in Large Networks: A Local Spectral Approach, *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, ACM, Republic and Canton of Geneva, Switzerland, pp.658–668.

[79] **Zhang, Z.**, **Sun, K. and Wang, S.** (2013). Enhanced Community Structure Detection in Social Networks, *Scientific Reports*, *3*, 3241.

[80] **Aldecoa, R. and Marín, I.** (2013). Surprise maximization reveals the community structure of complex networks, *Scientific Reports*, *3*, 1060.

[81] **Fortunato, S. and Lancichinetti, A.** (2009). Community Detection Algorithms: A Comparative Analysis: Invited Presentation, Extended Abstract, *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, ICST (Institute

for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, pp.27:1–27:2.

[82] **Aldecoa, R. and Marín, I.** (2013). Exploring the limits of community detection strategies in complex networks., *Scientific Reports*, *3*, 2216.

[83] **Moradi, F.**, **Olovsson, T. and Tsigas, P.** (2012). An Evaluation of Community Detection Algorithms on Large-scale Email Traffic, *Proceedings of the 11th International Conference on Experimental Algorithms*, SEA'12, Springer-Verlag, Berlin, Heidelberg, pp.283–294.

CURRICULUM VITAE

Name Surname: Nagehan Ilhan

Place and Date of Birth: Şanlıurfa,1982

E-Mail: nagehan.ilhan@gmail.com

EDUCATION:

- **B.Sc.:** 2004, EMU, Computer Engineering
- **M.Sc.:** 2007, EMU, Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2004-2007 Research Assistant, Eastern Mediterranean University
- 2007-2008 Research Assistant, Harran University
- 2008-2016 Research Assistant, Istanbul Technical University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Ilhan N.**, Gündüz Öğüdücü Ş., 2016. Community Evolution Prediction Framework based on Time Series Modeling in Dynamic Networks. *Information Science*, *Submitted*.
- **Ilhan N.**, Gündüz Öğüdücü Ş., 2016. Feature Identification for Predicting Community Evolution in Dynamic Social Networks. *Engineering Applications of Artificial Intelligence*, 55(2016), 202-218.
- **Ilhan N.**, Gündüz Öğüdücü Ş., 2015. Predicting Community Evolution Based on Time Series Modeling. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '15)*, August 25-28, 2015 Paris, France.
- **Ilhan N.**, Gündüz Öğüdücü Ş., 2013. A Study on Generation of Synthetic Evolving Social Graph. *International Conference on Agents and Artificial Intelligence (ICAART '13)*, February 15-18, 2013 Barcelona, Spain.
- **Ilhan N.**, Gündüz Öğüdücü Ş., 2013. Community Event Prediction in Dynamic Social Networks. *IEEE International Conference on Machine Learning and Applications (ICMLA '13)*, December 4-7, 2013 Miami, Florida, USA.