International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 9, ISSN No. 2455-2143, Pages 22-30
Published Online January 2019 in IJEAST (http://www.ijeast.com)

TO FILTER UNWANTED MESSAGES FROM OSN USER WALL

Monal S. Duragkar
Dept. Computer Technology,
KDK college of Engineering, Nagpur, India

Kartik M. Chandankhede
Dept. Computer Technology,
KDK college of Engineering, Nagpur, India

Sanket T. Gadpayle
Dept. Computer Technology,
KDK college of Engineering, Nagpur, India

Pratiksha S. Bobade
Dept. Computer Technology,
KDK college of Engineering, Nagpur, India

Abstract: **Social media and Social Network Analysis (SNA) acquired a huge popularity and represent one of the most important social and computer science phenomena of recent years. One of the most studied problems in this research area is influence and information propagation. The aim of this paper is to analyze the information diffusion process and predict the influence (represented by the rate of infected nodes at the end of the diffusion process) of an initial set of nodes in two networks: Facebook user's contacts users commenting these post. These networks are dissimilar in their structure (size, type, diameter, density, components), and the type of the relationships (explicit relationship represented by the contacts links, and implicit relationship created by commenting on post), they are extracted using Node XL tool. Three models are used for modeling the dissemination process: Linear Threshold Model (LTM), Independent Cascade Model (ICM) and an extension of this last called Weighted Cascade Model (WCM). Networks metrics and visualization were manipulated by NodeXL as well. Experiments results show that the structure of the network affect the diffusion process directly. Unlike results given in the blog world networks, the information can spread farther through explicit connections than through implicit relations.**

Keywords: ***Information diffusion, influence, social media, social network analysis.***

I. INTRODUCTION

Human social relationships were bounded according to time and space, but the evolution of information and communication technologies tools allowed people to inexpensively and reliably share information anytime and anywhere through social media (YouTube, Flickr, Twitter, Facebook , blogs, emails, etc). These tools are helpful recourses of information, opinions and behaviors regarding different areas of interest. Studying and measuring these social media have attracted considerable interest of many researchers in various domains and led them to create a new field called Social Network Analysis (SNA). In our daily life, there are innumerable situations in which we are influenced in our decision making by what others around us are doing. Simple examples of influence are when academic researchers choose to work on a topic that is currently ″hot″, or when we listen to the same music that our friends listen to In [1], the marketing strategies were enhanced with a word-of-mouth approach using probabilistic models of interactions to choose the best viral marketing plan. Some other researchers focused on information diffusion in certain special cases. Given an example, the study of Sadikov et.al [2], where they addressed the problem of missing data in information cascades, and evaluated their methodology using information propagation cascades in Twitter network, by involving a K-tree model to estimate properties of the cascade of information, such as size and depth. Moreover, a study on the Blog worlds proposed a special model of information diffusion based on explicit and implicit links [3].

Explicit links are the relations formed between blogs directly to obtain information or to maintain a relationship, whereas when information is diffused between blogs not through an explicit relationship, it is called implicit link. Other researchers have studied and modeled social media epidemics (like viruses and rumors) especially on Twitter [5]. Several mathematical and physical based diffusion models have been suggested to formally the spread of information in a network [7]. The literature offers four basic approaches to modeling influence propagation in social networks: cascade models, threshold models, epidemic models and game theory models.

In this paper, we focus on analyzing the information propagation process for anticipating the capability of nodes in spreading the information throughout the network. We also aim to understand how the structure of the network and the type of its relationships can influence the propagation process.

Likewise, this analysis is done on two different networks: an explicit network created from Facebook user's contacts, and an implicit network created from users' comments on Facebook chat. The discussion of networks treats them as static structures: we take a snapshot of the nodes and edges at a particular moment in time and then analyze their structure and the diffusion of information process. These networks were extracted using NodeXL tool.

The primary communication method of social networking services is posts. By writing posts and comments, people share their thoughts, opinions, and real time status. People tend to write messages and replies to people they have close friendships with or are arguing with.

Also people write lots of comments one after another when they are having an in-depth conversation. Based on this tendency, we made a model on the latent social relationship among users by examining and analyzing the message threads. In this paper, we propose a method to extract the latent social relationship from a social networking service by analyzing the users' activities. The users' writing patterns are especially considered to examine the intensity of the conversation and the strength of the social relation.

We applied our algorithms to a Facebook dataset and developed an evaluation system to appraise the proposed algorithms. The experimental result shows that the proposed method using a weighted harmonic rule with a root-included sliding window fits best for social relation extraction.

II. LITERATURE SURVEY

The dramatic increase of popularity of social networks has attracted a lot of research. Factors like social interaction, knowledge exchange, knowledge discovery, ability to capture data about various types of social interactions at a very fine granularity with practically no reporting bias, and availability of data mining techniques for building descriptive and predictive models of social interactions have become key drivers for computer science research in SNA.

Discovering knowledge from these networks is a challenging and primary research issue because of their size, reachability and diversity. Several research studies have been conducted on social network analysis and two main approaches have been studied: one is influential user discovery [6] and the other is social network construction. The aim of the first approach is finding most influential users in communities by analyzing their relationships and activities and the second approach concentrates on discovering social network of users [9]. This section presents the work conducted so far in the latter approach. Data from different sources like Web, e-mail communication logs, instant messenger logs, blogs, etc. has been used either individually or in combination for the purpose of social network extraction. In the section below classification of the various techniques used for network extraction has been presented.

a) Co-occurrence of Names on the World Wide Web:

Several studies have been undertaken to use a search engine to extract social networks from the Web. Co-occurrence of names on the web, is obtained by posing a query including two names to a search engine, is commonly used as proof of relational strength. The network obtained is an egocentric network, in that it is focused on a specific person. The input to the system is name of the person (X) whose social network is to be obtained and the system extracts a list of related people (L). Jaccard coefficient [11] is used to measure the significance between X and Y, where YL. The process is repeated for each YL. The goal here is to find series of links i.e. referral chain from the requester node to the expert node (information hub). A path from a person to a person is obtained automatically using the system. With increasing usage of Internet and development of WWW large amount of information about our daily lives is available online, making automatic extraction of social relations more demanding than when Referral Web was developed.. The idea behind this study is that: at academic conferences, a participant registers a brief profile with fields like Name, E-mail, Affiliation, etc. well before the conference which means that there is enough time to gather information about the participants from the Web. The relationships between any two participants are determined using the Web information gathered by posing a query to a search engine in a similar fashion. An edge exists between two nodes if the Jaccard Co-efficient between those two nodes is larger than a threshold value and the weight of that edge is set equal to the Jaccard Co-efficient. To alleviate the problem of ambiguity, it labels the relationships between nodes and uses machine learning to identify them. Social network with 650 conference participants of WWW2002 has been extracted. The Web mining component of Flink similar to that of obtains hit count from a search engine (Google) for both the persons X and Y individually as well as hit count for co-occurrence of these two names with the target being the Semantic Web community. It also performs the additional task of associating a researcher with a given topic of interest. The Web information source in this case being Web pages, e-mail messages, publication archives, and self created profiles (FOAF files). In [10], the strength of relations among individuals is calculated using the Jaccard coefficient $nX∩Y/nX∪Y$, where $nX∩Y$ represents the number of hits yielded by the query X AND Y and $nX∪Y$ represents the number of hits by the query X OR Y. The two researchers are considered to share a relation if the value is greater than a certain threshold. The term “Semantic Web OR Ontology” is added to the query for name disambiguation. Although, [10] has tried to remove the problem of ambiguity in identification of entities with similar names, the system still has certain problem because of data collection (general noise, errors in the extraction of specific cases) in this respect. Matsuo et al. developed POLYPHONET, which also uses a search engine (Google) to measure the co-occurrence of names. In their study, several co-occurrence measures [12] have been compared, including the matching coefficient ($nX∩Y$), mutual information, Dice coefficient, Jaccard coefficient, and overlap coefficient. The overlap coefficient $nX∩Y/ min(nX, nY)$ performs best according to the experiments. In addition, POLYPHONET was operated at several AI conferences in Japan and a couple of international conferences to promote participants communication. For disambiguating personal names, key phrases such as affiliations are added to queries.

\The efficiency and accuracy of an extracted social network depends primarily on whether it has been able to address well the problems associated with profile extraction and name disambiguation. Arnetminer [3] focuses primarily on profile extraction and name disambiguation for academic researchers. The system constructs a semantic based social network of academic researchers by extending the Friend-Of-A-Friend (FOAF) ontology [13] as the profile schema, proposes a unified approach based on Conditional Random Fields to extract researcher profiles from the Web using a search engine and integrating the extracted researcher profiles and the crawled publication data from the online digital libraries. A unified probabilistic framework for dealing with the name ambiguity problem has been proposed for integration. It proposes three generative probabilistic models for simultaneously modeling topical aspects of papers, authors, and publication venues. Based on the modeling results, it implements several search services such as expertise search and association search. It proposes a unified approach to profiling consisting of three steps: relevant page identification, preprocessing, and extraction. In relevant page identification, given a researcher name, list of web pages is obtained by a search engine (Google API) and then homepage/introducing page are identified using a binary classifier (SVM).

Preprocessing has two steps (a) separating the text into tokens and (b) assigning possible tags to each token using Conditional Random Fields (CRFs) as the tagging model. After each token is assigned with several possible tags, profiling is performed. For name disambiguation, [8] uses five types of relationships: Co-Author, Citation, Co-PubVenue, Constraints, and -Co-Author, have been used with each type of relation having an impact on F1-score and the relationship of Co-Author having the highest impact (+24.38% by F1). The draw back in this case is that k (actual number of researchers having same name, say ‘a’) has to be provided manually.

b) E-mail Communications:

Email is a valuable and pervasive mean of communication in the information society, is one of the primary ways that people use to communicate and access their widespread social networks, and as such it is a highly relevant area for research on communities and social networks. It is the number one online activity for most users and there are few advanced email technologies that take advantage of the large amount of information present in a user’s inbox . Maintaining and using contacts is an essential and challenging task. Unfortunately, the task of manually maintaining contact information is tedious and error-prone and a system that extracts contact information automatically from email messages itself has limited coverage because of limited to the data present in email. Due to rapid development of electronic communications, email data becomes a powerful information source for studying social networks because of a number of advantages: availability of large amount of data on personal communications in a standard electronic format; ubiquity of email usage; frequency, longevity, and reciprocity of email communications; type (content) of communication; temporal data; and availability on both sender and receiver side. In addition to the advantages, accessing email communications has certain issues as well. Privacy issues like compromising personal privacy and organizational confidentiality concerns are the biggest barriers for email related social research which can be alleviated by accessing only header information but ignoring information carried in the message significantly limits the potential of using email as source of information for analyzing social relationship. Although the format of email messages is relatively standard and it is easy to generate a communication links from email archives, automatic extraction of social network is not easy because of issues like: multiple identities of same person; spam and group aliases; categorization of social relations by email content; weighting ties by different indicators such as reciprocity, frequency, and longevity of discussion.

Several studies have used email communication as data source for social network extraction and tried to leverage the associated benefits and address the issues concerning its usage. Communities of practice are the informal networks of collaboration that naturally grow, collaborate, coexist with the formal structure within organizations, serve many purposes, such as resolving the conflicting goals of the organization to which they belong, solving problems in more efficient ways, and furthering the interests of their members. Any organization that provides opportunities for communication among its members is eventually threaded by communities of people who have similar goals and a shared understanding of their activities.

III. PROPOSED WORK

1.1 OVERVIEW/INTRODUCTION

The publication of social network data entails a privacy threat for their users. Sensitive information about users of the social networks should be protected.

The challenge is to devise methods to publish social network data in a form that affords utility without compromising privacy. Previous research has proposed various privacy models with the corresponding protection mechanisms that prevent both inadvertent private information leakage and attacks by malicious adversaries. These early privacy models are mostly concerned with identity and link disclosure. The social networks are modeled as graphs in which users are nodes and social connections are edges. The threat definitions and protection mechanisms leverage structural properties of the graph. This paper is motivated by the recognition of the need for a finer grain and more personalized privacy.

ALGORITHM DESCRIPTION

• Input: A social network SN, with number of user as node n and user features like images posts etc.

• Output: A social Network SN with user accessing feature of other user .

1) Create number of number of nodes(user) n = 1, . . . ,N having personal information.

2) Each node(user) will have facility add other nodes(user) as a friend by send friend request.

3) Each node(user) will have facility to accept other node(user) friend request and connect to it as a friend.

4) Each node(user) can have n = 1, . . . ,N numbers of friends.

5) In normal condition every other node(user) can write post on other user(friends) wall, read friends post, share friends post, share image.

6) The algorithm starts out with group formation, during which all nodes that have not yet been grouped are taken into consideration, in clustering-like node

7) Here each user (node) will form two group of its connecting friends, one close friend and other friends, this will depend upon user(node) to whom to select as close friend or friend which will be based on maximum similarity both user(nodes) have.

8) Depending upon the group(close friend and friend), nodes features will be decided, which features should be shown to which group.

9) Group (close friend and friend) will be divided into post, view and image authority.

10) Depending on the authority specific user with have access into features of other user.

1.2 WORKING OF ALGORITHM

1) The project is divided into two part ,first is Web Pages were JSP and html pages are kept which are user interface of this project, and second is Source Package were Java code is kept.

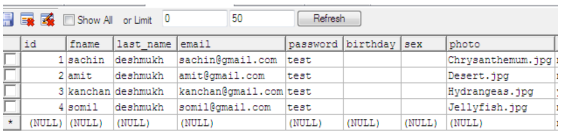
2) The first page that we see when we run the project is LoginForm.jsp, here in html page is design and it contains username and password field to check username and password are correct or not ,when button is click ,it goes to java class “LoginCheck”. This class is in Source Package folder inside org.Servlet .

3) In this class we get data from username and password filed, then get database connection from DBConnectionClass class and then check that username and password exists in database or not.

4)Then we again go to LoginFrom.jsp which is in Web Pages folder ,were if new user is there ,we go to registration in SignUp.jsp.

5) In Signup.jsp registration page is deigns using html and when button is click ,we go to “SignUpServlet” .This class is in Source Package folder inside org.Servlet .

6) In this class we take data from Signup.jsp like username ,email, password ,date of birth, photo etc. , then get database connection from DBConnectionClass class and then store whole record in table “sign_up_tbl”.At the same time here we create node for each user by creating table for each user and giving them name as “n1”,”n2”.(This is our first module i.e Data Collection)



	id	fname	last_name	email	password	birthday	sex	photo
☐	1	sachin	deshmukh	sachin@gmail.com	test			Chrysanthemum.jpg
☐	2	amit	deshmukh	amit@gmail.com	test			Desert.jpg
☐	3	kanchan	deshmukh	kanchan@gmail.com	test			Hydrangeas.jpg
☐	4	somil	deshmukh	somil@gmail.com	test			Jellyfish.jpg
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

7) If we successfully get login ,then we MainPage.jsp which is Web Pages folder, In this page is design in html ,to show login user page and link to add friends, send request of friends, set attribute, add images etc.

8) In MainPage.jsp when we click “Friends” link we go to “FrndPage.jsp”.In FrndPage.jsp we can see all the friends ,In this page we call GetFrndList.class ,This class is in Source Package inside org.DBPacge .IN this class we get database connection from DBConnectionClass class and then get all friends list of specific user from sign_up_tbl table .and then List of friends are shown in “FrndPage.jsp”.

9) In MainPage.jsp when we click “Friend Request” link we go to “ AcceptFrnd.jsp”.In AcceptFrnd.jsp we can see all the user which are not friends ,In this page we call MutualFriend.class ,This class is in Source Package inside org.DBPacge .IN this class we get database connection from DBConnectionClass class and then get all non friends list of specific user from sign_up_tbl table .and then List of non friends are shown in “ AcceptFrnd.jsp “.

10) In AcceptFrnd.jsp ,when we click “Accept Request” button ,it go to “AcceptFrnd”, This class is in Source Package folder inside org.Servlet .

In this class we get data i.e from were request has come and to whom friends has to be added, then get database connection from DBConnection class, and then update specific node table with frndReq values i.e If n1 is table ,and n2 is his friend ,then in n1 table there will be row with frndReq column as 2. .(This is Reduce Node Degree Module)

11) In MainPage.jsp when we click “Send Request” link we to “ showUsers.jsp”.In showUsers.jsp.IN this page we get database connection from DBConnectionClass class and then get all friends list of specific user from sign_up_tbl table .and then show list of all user in “ showUsers.jsp “.

12) IN showUsers.jsp when we click “Send Request “ button ,we go to “sendRequst.jsp”,IN sendRequst.jsp we get data i.e from which node(user) request has come, and to which node (user) request has to receive. and then insert row in specific node table for friend request.

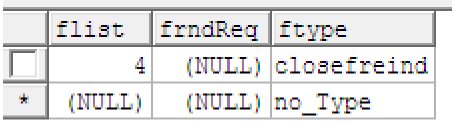
i.e if request is send from node n1,to n2,htne in n1 table one row will be added fro friend request to n2. .(This is Reduce Node Degree Module)

13) In MainPage.jsp when we click “Set Post Attributes” link we go to “ goupAutherity.jsp”.In

goupAutherity.jsp.IN this page we first show all the friends of users who have accepted request from above code. That information is there in node specific table i.e n1 ,n2.

Then we have add name of close friend in text area ,and when we click submit button ,we go to CloseFrndGrp.jsp, in this jsp we get we get database connection from DBConnectionClass class and then get names of users enter in text area as close friend and the update specific node table with close friend relation .

I.e n1 is node table ,and its has enter n4 as close friend ,then in n1 table against n4 columns “ftype “ will be updated as close friend.

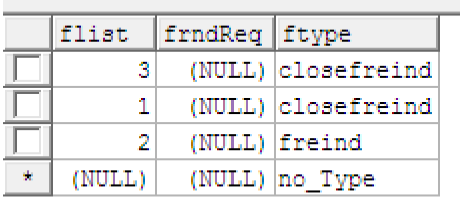


	flist	frndReq	ftype
□	4	(NULL)	closefreind
*	(NULL)	(NULL)	no_Type

(This is Add Node Degree module)

14) Then we have add name of normal friend in below text area ,and when we click submit button ,we go to FriendGroup.jsp, in this jsp we get we get database connection from DBConnectionClass class and then get names of users enter in text area as friend and the update specific node table with close friend relation .

I.e n4 is node table ,and its has enter n2 as friend ,then in n4 table against n2 columns “ftype “ will be updated as friend.

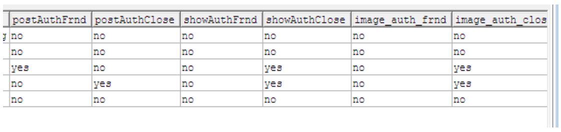


	flist	frndReq	ftype
□	3	(NULL)	closefreind
□	1	(NULL)	closefreind
□	2	(NULL)	freind
*	(NULL)	(NULL)	no_Type

(This is Add Node Degree module)

15) Then in next portion we provide authority to group .i.e whether close friend group and friend group should see post, send post and see image of specific user.

When you click check box and click submit button ,we go to setShowAttribute.jsp,in this jsp we get database connection and then get specific assigned authority and then update sign_up_tbl with authority as “yes” or “no”



postAuthFrnd	postAuthClose	showAuthFrnd	showAuthClose	image_auth_frnd	image_auth_clos
no	no	no	no	no	no
no	no	no	no	no	no
yes	no	no	yes	no	yes
no	yes	no	yes	no	yes
no	no	no	no	no	no

Here in FrienGroup.jsp and groupAutjority.jsp our GINN algorithm code is written i.e

The algorithm starts out with group formation, during which all nodes that have not yet been grouped are taken into consideration, in clustering-like node. In the first run, two nodes with the maximum similarity of their neighborhood labels are close friend. Their neighbor labels are modified to be the same immediately so that nodes in one group always have the same neighbor labels.

Larger value indicates larger similarity of the two close

friend. Then nodes having the maximum similarity with any node in the group are clustered into the group till the group has ` nodes with different sensitive labels.

Thereafter, the algorithm proceeds to create the friend group. If fewer than ` nodes are left after the last group’s formation, these remainder nodes are clustered into friend groups according to the similarities between nodes and groups.

(This is Add Node Degree module)

15) In MainPage .jsp when we click “Post on frnd wall” link, we go to “FreindsPage.jsp”. In FreindsPage.jsp, we first get list of all the friends to specific node table, and then from sign_up_tbl table check whether that specific node have post authority, if it has authority ,then it can post data on that user(node) .

(This is Add Noise Node module)

15) In MainPage .jsp when we click “share image” link, we go to “ shareImages.jsp”. In shareImages.jsp, we first browser to get image path and then go to ShareImage class which is inside Source Package folder inside org.ServletPacage, in this class we get image path from ServletFileUpload class and the get database connection from DBConnectionClass(),and then insert row in image_tbl, with image name and who has posted image.

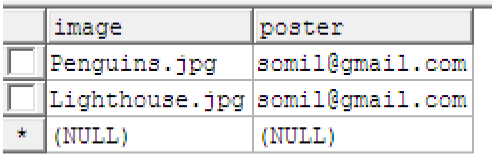


	image	poster
□	Penguins.jpg	somil@gmail.com
□	Lighthouse.jpg	somil@gmail.com
*	(NULL)	(NULL)

(This is Add Noise Node module)

16) In MainPage .jsp when we click “Show Share Image” link, we go to “ ShowImages.jsp”. In ShowImages.jsp, in this jsp we get database connection and then from image_tbl table get image name and who has posted that image and ,the show that image on page.

(This is Add Noise Node module)

17) In MainPage .jsp when we click “Show Friend Share Image” link, we go to “ ViewFrndImages.jsp”. In ViewFrndImages.jsp, we first get list of all the friends to specific node table, and then from sign_up_tbl table check whether that specific node have show image authority, if it has authority ,then it can see image of that user(node) .

(This is Add Noise Node module)

18) To see node graphs we have create class in Source Package folder ,inside NodeCreation package,the class name is CraeteNode.java,in this class we get database connection from DBConnectionClass.java, then from sign_up_tbl table ,get list of all user and then from node table get friends and close friend list.

The we use org.neo4j.graphdb.Node class we create nodes in graphs form, from org.neo4j.graphdb.Relationship class we create relationship between each node, from org.neo4j.graphdb.RelationshipType classwe create which type of relation is is havng i.e close or normal friend and then from org.neo4j.graphdb.factory.GraphDatabaseFactory class we show that in graph format in neo4j url.

IV. METHODOLOGY AND IMPLEMENTATION

4.1. INTRODUCTION

The social networks are modeled as graphs in which users are nodes and features(images ,post) are labels. Labels are denoted either as sensitive or as non-sensitive. We treat node labels both as background knowledge an adversary may possess, and as sensitive information that has to be protected. We present privacy protection algorithms in which users have to create the groups between its connected node (close friend and friend) and decide which labels to be accessible to which group nodes .

To this aim, the algorithms transform the original graph into a graph in which nodes are sufficiently indistinguishable. The algorithms are designed to do so while losing as little information and while preserving as much utility as possible. We evaluate empirically the extent to which the algorithms preserve the original graph's structure and properties. We show that our solution is effective, efficient and scalable while offering stronger privacy guarantees than those in previous algorithm.

4.2 BLOCK DIAGRAM

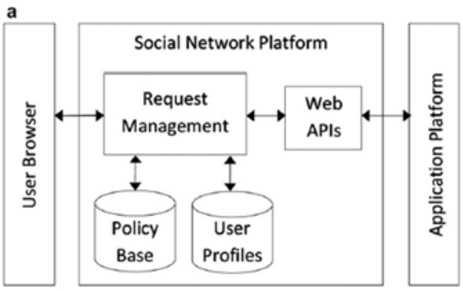


Fig.1 Social Networking Block Diagram

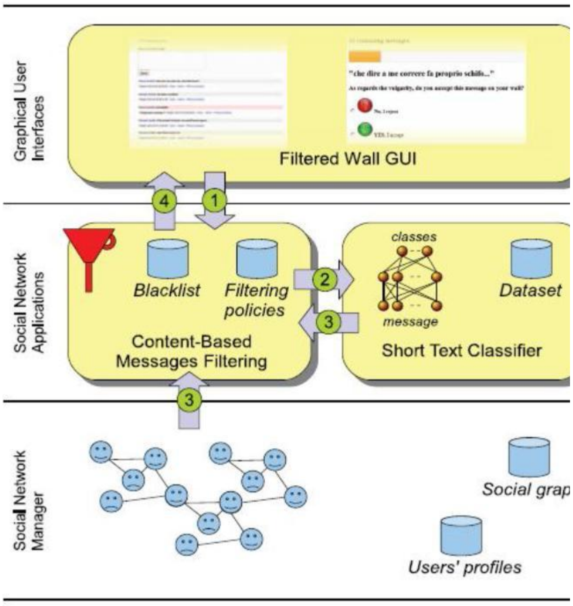

Graphical User Interfaces
Filtered Wall GUI
"che dire a me correre fa proprio schifo..."
As regards the vulgarity, do you accept this message on your wall?
4
1
Social Network Applications
Blacklist
Filtering policies
Content-Based Messages Filtering
2
3
classes
message
Dataset
Short Text Classifier
3
Social Network Manager
Social grap
Users' profiles

Fig. 2 Application Interaction Block Diagram

4.3 IMPLEMENTATION

Applications require permission to access user's profile data to provide a service customized to the user's profile data. In this section we present our approach to enable fine grain access control for third party applications, to limit applications' access only to relevant user's profile data. We first provide some preliminary definitions related to applications and API set, and then we discuss our proposed fine grain access control framework for API based applications.

Our social network web sites will released APIs (Application Programming) that allow developers to leverage and aggregate information stored in user profiles and provide extended social network services. The exposed APIs are basically a set of web services that provide a limited and controlled view for the application to interface with the social network site. The social network application architecture includes three interacting parties namely the user, social network server, and the third party application server. Fig. (a), shows the different blocks used in the social networks architecture.

Note that the application server is able to connect to social network through the exported web APIs. Furthermore, these requests are filtered through the request management module .

For example, consider an application that recommends user to see features of other user(friends). In this case, the application requires access to retrieve user(friend) information like posts, images etc. Then with the help of our application it will decided that which feature user is authenticate to see the user(friend) feature.

Some other applications would not only require to see all features from your profile but would also require features from your close friends' profiles. For example, consider an application that projects your friends as a close friend and want to see all your feature and his all features you should be see. This application requires all your features to be open for close friend profile and viceversa.

Social networks provide mechanisms for users to customize their profiles and to add applications developed by external developers. The application provides the customized services by accessing the exported APIs. Fig. 2(b), depicts the interaction stages between the user browser, social network and the third party application. The interaction starts when a user requests an application APP (Steps 1e2). The application server interacts with the social network server by instantiating API calls (Step 3). Upon receiving the responses of the API calls, the application server compiles and sends a response to the social network which is forwarded to the requesting user (Steps 4e5).

V. EXPERIMENTAL RESULTS

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

For our project we have taken facebook data from various sites like

http://snap.stanford.edu/data/egonets-Facebook.html,
http://konect.uni-koblenz.de/networks/facebook-wosn-links

http://socialnetworks.mpi-sws.org/data-wosn2009.html.

We have downloaded the data and added that to the excel sheet graph, to provide comparison between various techniques available for privacy protection in social networking based on Scalability, Time and Security.

VI. CONCLUSION

In this paper we have investigated the protection of private label information in social network data publication. We consider node graphs with rich label information (data and images), which are categorized to be either sensitive or non-sensitive.
We assume that user will divides its nodes into two group close friend and friend such that a model for attaining privacy while publishing the data, in which node labels are both part of adversaries' background knowledge and sensitive information that has to be protected. We accompany our model with algorithms that transform a node network graph before publication, so as to limit adversaries' confidence about sensitive label data. Our experiments on both real and synthetic data sets confirm the effectiveness, efficiency and scalability of our approach in maintaining critical graph properties while providing a comprehensible privacy guarantee.

VII. REFERENCES

[1] Richardson M. and Domingos P., (2013) "Mining knowledge-sharing sites for viral marketing," in Proc. of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining ACM New York, NY, USA ©2002. International Journal of Future Computer and Communication, Vol. 2, No. 3.

[2] Sadikov E., Medina M., Leskovec J., and Molina H. G., (2011) "Correcting for missing data in information cascades," WSDM'11, Hong Kong, China, February 9–12.

[3] Kwon Y. S., Kim S.W., Park S. J., Lim S. H., and Lee J. B., (2009) "The information diffusion model in the blog world," The 3rd SNA-KDD Workshop '09 (SNA-KDD'09), Paris, France, June 28, 2009.

[4] Jafa K., McElroy P., Fitzpatrick L., Borkowf C. B., MacGowan R., Margolis A., Robbins K., Youngpairoj A. S., Stratford D., Greenberg A., Taussig J., Shouse R. L., LaMarre M., Lemal E. M., Heneine W., and Sullivan P. S., (2009) "HIV transmission in a state prison system 1988–2005," PLoS ONE, vol. 4, no. 5, May 2009.

[5] Qazvinian V., Rosengren E., Radev D. R., and Mei Q. Z., (2011) "Rumor has it: Identifying misinformation in microblogs," in Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1589–1599, Edinburgh, Scotland, UK, July 27–31, 2011.

[6] McClurg S. D., Wade M. L., and Phillips M. V. W., (2012) "Gender, social networks, and voting behavior, Political networks paper archive," Working Papers, (pp. 64), 2012

[7] Easley D. and Kleinberg J., (2010) "Networks, crowds, and markets: Reasoning about a highly connected world," Cambridge University Press, 2010.

.[8]NodeXL Graph Gallery. [Online]. Available: http://nodexl.codeplex.com/

[9] Shneiderman D. L. H. B and Smith M. A., (2011) "analyzing social media networks with nodexl," Insights from a Connected World, Elsevier, 2011.

[11] Babutsidze Z., Lomitashvili T. and Turmanidze K., (2011) "The structure of georgian blogosphere and implications for information diffusion," August 5, 2011.

[12] Demaine E. D. and Zadimoghaddam M., (2010) "Minimizing the diameter of a network using shortcut edges," Lecture Notes in Computer Science, vol. 6139, Algorithm Theory - SWAT 2010, (pp. 420-431).

[13] Paolillo J. C., (2008) "Structure and network in the YouTube core," in Proc. Of the 41st Annual Hawaii International Conference on System Sciences, pp. 156, IEEE Computer Society, Washington DC, USA ©2008.