Mobile Phone Enabled Social Community Extraction for Controlling of Disease Propagation in Healthcare

Yanzhi Ren*, Jie Yang*, Mooi Choo Chuah†, Yingying Chen*
*Dept. of ECE, Stevens Institute of Technology
Castle Point on Hudson, Hoboken, NJ 07030
{yren2, jyang, yingying.chen}@stevens.edu
† Dept. of CSE, Lehigh University
Bethlehem, PA 18015
chuah@cse.lehigh.edu

***Abstract*—New mobile phones equipped with multiple sensors provide users with the ability to sense the world at a microscopic level. The collected mobile sensing data can be comprehensive enough to be mined not only for the understanding of human behaviors but also for supporting multiple applications ranging from monitoring/tracking, to medical, emergency and military applications. In this work, we investigate the feasibility and effectiveness of using human contact traces collected from mobile phones to derive social community information to control the disease propagation rate in the healthcare domain. Specifically, we design a community-based framework that extracts the dynamic social community information from human contact based traces to make decisions on who will receive disease alert messages and take vaccination. We have experimentally evaluated our framework via a trace-driven approach by using data sets collected from mobile phones. The results confirmed that our approach of utilizing mobile phone enabled dynamic community information is more effective than existing methods, without utilizing social community information or merely using static community information, at reducing the propagation rate of an infectious disease. This strongly indicates the feasibility of exploiting the social community information derived from mobile sensing data for supporting healthcare related applications.**

I. Introduction

The recent years have witnessed an explosion of the usage of mobile wireless devices in our daily lives. In particular, with the rapid deployment of sensing technology in mobile phones, the collected sensing data can be comprehensive enough to be mined not only for the understanding of human behaviors but also for supporting a broad range of applications. For instance, most of the mobile phones support the Bluetooth technology, and the Bluetooth device-discovery software running in a mobile phone allows it to collect information from other nearby Bluetooth devices. It is thus convenient to exploit the mobile phones equipped with Bluetooth technology to discover the encounter events between people such that their social relationships can be derived and analyzed. More importantly, the discovered social relationships can be used to extract social communities [1], [2], which reflect close relationships or similar behavior patterns among people, to assist in the development of applications in various domains, ranging from monitoring/tracking applications, to medical, emergency and military applications.

Group discovery and community detection have been an active research area. In [3], the Kernighan-Lin algorithm was introduced to improve the initial division of a network by optimizing the number of graph edges within and between the partitions using the greedy algorithm. [1] proposed a hierarchical clustering algorithm where communities are merged based on a similarity measure.

The social community structures have been used actively in many areas including online social networks, e.g., community detection in multi-dimensional networks based on online social media [4], and wireless networks, e.g., coping with the propagation of malware on smart phones [5], and facilitating the packet forwarding in Delay Tolerant Networks (DTNs) [6]. However, few studies have been done in exploiting social community structures extracted from mobile phones to control the propagation of infectious diseases in the healthcare domain. [7] studied the relationships between the voluntary vaccination and the transmission of a vaccine-preventable infection. It pointed out that the propagation of the disease is related with the neighborhood size. Besides the traditional random vaccination strategy, recent work used bridge users identified in the human contact networks as distribution points of vaccination [8]. We are not aware of any prior work that exploits social relationships systematically for effective vaccination such that the propagation rate of an infectious disease can be reduced.

Our work is novel in that we extracted dynamic social community information by leveraging the contact traces derived from mobile phones and proposed a community based framework for control of disease propagation.

We experimentally evaluated our framework through a trace-driven approach by using the MIT reality mining trace [9]. The results showed that our strategy is highly effective in reducing the disease propagation rate when compared to methods that do not use social relationships.

The rest of the paper is organized as follows. We present our mobile phone enabled social community based framework in Section II. It describes the system model in our framework and the disease infection model

978-0-7695-4469-4 2011

DOI 10.1109/MASS.2011.68

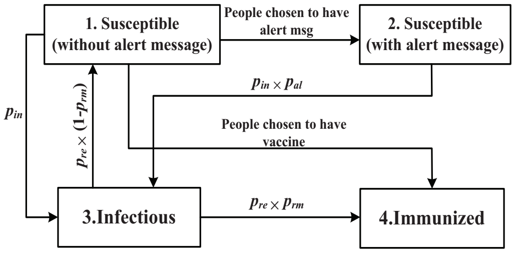


Fig. 1. Epidemic infection model used in our framework.

used in this work. We next present our dynamic social community based scheme in Section III. In Section IV, we validate the feasibility of our framework by using datasets collected from mobile phones and compare with existing methods. Finally, we conclude our work in Section V.

II. Framework Overview

In this section, we first provide the system model for our mobile phone enabled disease control framework, and present descriptions of the architecture in our framework. We envision this framework can be implemented by any State Department of Health through the coordination of the Centers for Disease Control and Prevention (CDC). For example, during the 2009 spreading period of the pandemic influenza A (H1N1) virus, every state in US is required to report the number of infected patients to the CDC. The available vaccines are then allocated appropriately by the CDC to the different states [10].

We then present the infection model used in our work and provide an analysis on the state transitions in our model. Without loss of generality, we do not consider the differences between users and assume that all the users follow the same infection model.

A. System Model

1) Uncovering Human Social Relationships from Contact Traces via Mobile Phones: Instead of random vaccine distribution, targeting vaccination to a group of people with higher risk of infection can provide more effective control of an infectious disease propagation. Traditionally, scientists and doctors have to rely on social relationships derived via manually recorded daily activities from human subjects [9]. However, this approach is tedious, error-prone as the human subjects may forget to perform recording from time to time, and can be out of date. In this work, we consider extracting social community information from human contact traces collected by mobile phones.

The Bluetooth enabled device-discovery process is simple and automatic, and thus is suitable for recording encounter events between people for social relationship analysis. Our framework will utilize the existing infrastructure in cellular networks. We assume the users are subscribed to the cellular data plan and recorded encounter events (which include discovered device IDs and timestamps) will be periodically sent back to a back-end server authorized by the service provider. The dynamic community extraction mechanism is run by the server. The detailed description of our dynamic community extraction approach is presented in Section III. Moreover, the extracted community information will be stored at the server and updated from time to time.

Notation	Description
p_{in}	Disease infection probability
$p_{in} \times p_{al}$	Disease infection probability with alert messages
p_{re}	Recovery probability after the recovery cycle
$p_{re} \times p_{im}$	Probability of recovery with immunization after the recovery cycle
$p_{re} \times (1 - p_{im})$	Probability from infective to susceptible after the recovery cycle
N_p	The length of the disease recovery cycle

TABLE I
Notations used in the infection model.

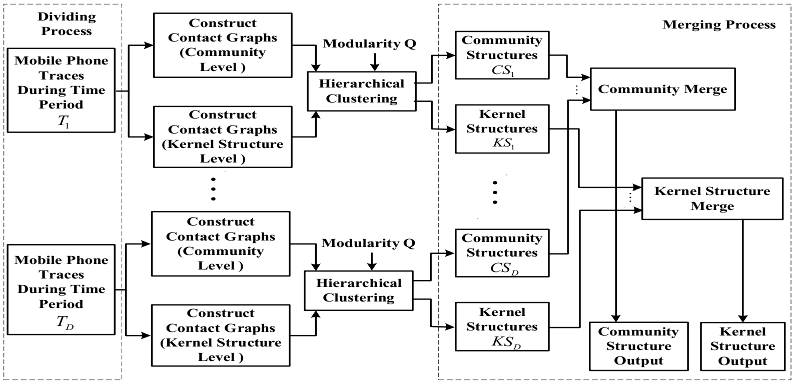
2) Architecture: We design two types of messages that a user may receive: *vaccination* and *alert*. A user who receives a *vaccination* message should go to obtain a vaccine shot, whereas a user receiving a *alert* message should take precautions as directed. We assume that all the users who have been notified will take the necessary recommendated actions. In our framework, vaccine shots of an infectious disease only have limited supplies and are more costly comparing to *alert* messages. The number of *alert* messages for each disease can be either controlled or unlimited.

When actions need to be taken for an infectious disease, the server will decide on who will receive *vaccination* messages and who will receive *alert* messages respectively based on the extracted social communities stored in its database. Then the server will send out each message to corresponding users.

B. Infection Model

In our framework, we extend the standard epidemic SIR model [11] to four states: *susceptible without alert*, *susceptible with alert*, *infective* and *immunized*. Susceptible means that a user can be infected by the disease. When a user is susceptible, he can be at either *susceptible without alert* or *susceptible with alert*. When a user is infected, he goes to the *infective* state and he can infect other people that he encounters. A user may go to the *immunized* state only when he is either vaccinated or has recovered from the disease with immunity.

The notations used in the infection model and across the paper are summarized in Table I. Figure 1 shows the state transition diagram. The probability of transmission from the *Susceptible without alert message* state to the *Infectious* state is defined as p_{in}, whereas it is $p_{in} \times p_{al}$ from *Susceptible with alert message* to *Infectious*. Further, we define the probability of recovery from the disease as p_{re} after every recovery cycle. We note that the *Infectious* state can transit to either the *Immunized* state with probability $p_{re} \times p_{im}$ or *Susceptible without*



Dividing Process
Mobile Phone Traces During Time Period T_1
Construct Contact Graphs (Community Level)
Construct Contact Graphs (Kernel Structure Level)
Modularity Q
Hierarchical Clustering
Mobile Phone Traces During Time Period T_D
Construct Contact Graphs (Community Level)
Construct Contact Graphs (Kernel Structure Level)
Modularity Q
Hierarchical Clustering
Merging Process
Community Structures CS_1
Kernel Structures KS_1
Community Structures CS_D
Kernel Structures KS_D
Community Merge
Kernel Structure Merge
Community Structure Output
Kernel Structure Output

Fig. 2. Flow overview: components to extract community information.

alert message with probability $p_{re} \times (1 - p_{im})$ after the person recovered from the disease.

III. Dynamic Extraction of Community Information

For the community information, we define two types of social clusters to represent different levels of social relationships: one is refereed to as *community* and the other is referred to as *kernel structure*. The people within the same community meet frequently with one another, while the kernel structure aims to capture a subset of people on top of the community structures that have even higher encounter frequency. Instead of using static community information derived from the whole trace, we propose an approach called *dividing* and *merging*, where dynamic community information is utilized since people may belong to different social communities at various times, and communities may appear or disappear in different time periods.

Flow Overview. Our dynamic community and kernel extraction approach is illustrated in Figure 2. First, mobile phones with Bluetooth capability record user encounter events. The recorded human encounter events are divided into multiple trace files based on each time window. We note that the length of the time window is adjustable (e.g., the length of the time window can be one day).

From each contact trace file, two contact graphs are constructed by the centralized server. One will be used for extracting community and the other for extracting kernel structure. Hierarchical clustering method can be one of the options to extract both community and kernel structures. The extracted community and kernel structures learnt for the current time period are then merged with the existing community and kernel structures that our system maintains. The combined community and kernel structures will be used to make decisions on who to send the *vaccination* and *alert* messages in our framework.

In the following subsections, we first describe how we construct contact graphs and our dividing strategy for the construction of community and kernel structures. Then, we describe how we merge the newly learnt community and kernel structures from different trace files with the existing community information.

A. Interval-Based Contact Graph Construction

The whole contact trace is divided into multiple trace files which cover different non-overlapping time intervals. We assume that each trace file consists of recorded encounter events that happened during a time period $[T_i, T_{i+1}]$. Each entry in such a trace is a record of one encounter event between two mobile phones: including the starting and ending time of the contact as well as unique IDs of the mobile phones. We also assume that the same person carries the mobile phone for the duration of the trace. Based on this information, a contact graph $G = (V, E)$ can be derived, which consists of a vertex set V and an edge set E. Each vertex $u \in V$ denotes a person, while each edge $e(u, v)$ denotes that person u has contacted person v for at least W times. The weight $t(u, v)$ denotes how frequent the two persons u and v meet during $[T_i, T_{i+1}]$. We use the number of times that the two persons have encountered with each other as the weight because the people who encounter with each other frequently tend to have closer relationships or similar social behaviors (e.g. riding on the same train to go to work each morning).

B. Community & Kernel Structures Extraction From Contact Graphs

For each trace file, we construct two contact graphs: one with $W = w_1$ and the other with $W = w_2$ where $w_2 > w_1$. Clusters extracted using the first contact graph are referred to as communities, whereas clusters extracted from the second one are referred to as kernel structures.

For scalability, it is important that an efficient algorithm is used to partition the contact graph $G = (V, E)$ into separate clusters. In this paper, we use a simple, yet effective partition algorithm called hierarchical clustering [12]. Further, to verify whether a particular division is meaningful or not, we use the modularity metric, Q [12]. This metric has often been used by researchers in previous studies to measure how good a partition is. A larger Q value indicates a better partition of the users.

C. Merging Community Information Extracted over Different Time Periods

Recall that the social community information may change with time: some communities may merge, some may disappear, and others may be divided into smaller ones. We next describe our community merging technique. We note that the same technique is applied to merge kernel structures.

We assume that we have D time windows. We have constructed one contact graph from each time period and we assume these are non-overlapping time periods: $[T_0, T_1]$, $[T_1, T_2]$,..., $[T_{D-1}, T_D]$ with $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$,..., $G_D = (V_D, E_D)$. The communities are extracted from each contact graph $G_i = (V_i, E_i)$ by using the hierarchical clustering algorithm and the modularity Q. Let S_i represents the set of communities found for time window i. Thus, we have $S_1, S_2, ..., S_D$. Each S_i contains a set of vertices A_i. Each A_i has been divided into k_i communities, which are represented as follows:

$$A_i = A_i^1 \cup A_i^2 \cup ... \cup A_i^{k_i} \tag{1}$$

We compare each community in S_i with all the communities discovered in S_{i+1} to see if a community in S_i satisfies one of the following conditions:

- It is part of a bigger community in S_{i+1} and hence can be removed.
- It can be merged with one community in S_{i+1} using the community merge operation for two communities A_i^j and A_{i+1}^l under an adjustable threshold τ:

$$\frac{|A_i^j \cap A_{i+1}^l|}{Max(|A_i^j|, |A_{i+1}^l|)} > \tau \tag{2}$$

- It is a superset of a community A_{i+1}^j in S_{i+1}, then A_{i+1}^j is removed from set S_{i+1} .

At the end of this operation, the two sets S_i and S_{i+1} are unioned to form a new $S_{i+2}^{'}$, which will merged with S_{i+2} in the next round of comparison. The merging process iterates through D time windows.

We have applied our community extraction approach using RFID traces that we collected in real-world environments. Our preliminary results indicate that our approach produces good detection rate. We are in the process of building our prototype using smartphones.

D. Using Extracted Community Information in Disease Propagation Control

Based on our community information extraction strategy, people that belong to the same kernel structures have a higher encounter frequency. Thus, an infectious disease has a higher probability to spread among these group of people if one person is infected already. Similarly, those in the same community as a sick person are also more susceptible to be infected by the disease. However, the probability for the disease to spread across two disjoint communities is low because people in such communities contact less frequently. We note that one person can belong to multiple communities or kernels. Let V_s represent the set of sick persons for a particular infectious disease. We define the susceptible persons who are in the same kernels as the sick people as V_k, while those susceptible persons who are in the same communities but are not in the same kernels as those sick people as V_c.

Because of the limited supply and relatively high cost of vaccines, an appropriate decision on efficient vaccination is that the vaccine shots and the alert messages should be given to those people who have higher risk of being infected by the disease. Thus, by utilizing the community information, the people in V_k should have higher priority to receive *vaccination* or *alert* messages than those in V_c. We further define the importance of a person by the weight when there are total M number of extracted communities (or kernel structures):

Definition 1. *The weight $W(v, S)$ of a person v in the community (or kernel structure) set $S = V_1, V_2, ..., V_M$ is defined as the total number of people in the community (or kernel structure) that v belongs to: $W(v, S) = \sum(|V_j| - 1)$ for all V_j which satisfies $v \in V_j$.*

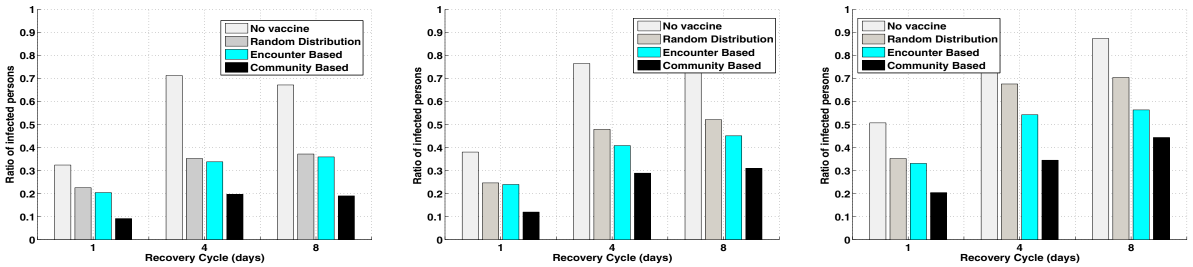
We further return the top K user list based on the following function:

Definition 2. *The $TOP(V, K)$ is defined as the function which can return a Top-K ranked list of the persons in V based on their weights W.*

Our goal is to find two optimum sets of people, one for receiving *vaccination* messages for vaccine shots, and the other for receiving *alert* messages, such that we can keep the infection rate low and effectively control the propagation of the disease. We next describe how these two sets of users are selected in our community-based framework.

Community Based Algorithm. As described in Section II, the server will decide who should receive *vaccination* or *alert* messages. The flow of the community based algorithm is as follows:

- The kernel structures V_k are considered first and the weight of each person in V_k reflects the priority. The function $TOP(V_k, K)$ is called to produce the top K user list L_k, where K is determined by the number of available *vaccination* or *alert* messages.
- If there are remaining *vaccination* and *alert* messages after considering all the people in V_k, then the community structures V_c are considered and the weight of each person in V_c reflects the priority. The $TOP(V_c, K)$ function is called to return the top K user list L_c, where K will be set to the remaining value of *vaccination* or *alert* messages.
- In the case that the number of these messages is larger than the number of total susceptible persons



(a) Scenario 1: Initial sick ratio: 0.03

(b) Scenario 2: Initial sick ratio: 0.15

(c)Scenario 3: Initial sick ratio: 0.3

Fig. 3. MIT traces: Performance comparison under different recovery cycle N_p when there are 15 vaccines and 30 alert messages with $p_{in} = 0.5$, $p_{al} = 0.7$, $p_{re} = 0.1$, $p_{im} = 0.2$, $p_{va} = 0.3$.

in $V_c \bigcup V_k$, the remainder messages are held till the next update of the community information as new persons may appear.

We note that for every round of calculation, choosing the candidates to send out the *vaccination* messages will take higher priority than *alert* messages.

IV. Performance Evaluation

In this section, we first describe our simulation methodology and present three existing methods for vaccination distribution. We then present the performance of our social community based methods by comparing to the existing techniques.

A. Simulation Methodology

We implemented our framework in a home-grown trace-driven simulator. We used a human contact-based trace, namely the MIT reality [9] trace which was collected using smart phones equipped with bluetooth devices. Each trace contains information about the IDs of the Bluetooth devices which are within the transmission range of each other, and the starting and ending times of their encounter. The MIT traces were collected from smart phones carried by 97 participants in an university environment. We used the first 20 days of the MIT traces which contains encounter events from 71 people. In particular, we used the first half of the trace (i.e., 10 days) as training data to extract the communities and kernel structures, and the second half trace as the testing data to evaluate our approach. We conducted extensive experiments on MIT trace by varying different parameters in our epidemiology infection model, including varying parameters of p_{in}, p_{al}, p_{re}, p_{im}, p_{va} and N_p from the infection model described in Section II. Due to the space limit, we only present a subset of the results in the following subsections.

B. Existing Methods

We compare our social-community based approach to the following three existing methods for efficient vaccine distribution to achieve effective disease propagation control.

Random Distribution Method. This is the most straight forward method. In this method, the server will randomly choose the users to receive the *vaccination* and *alert* messages.

Encounter-based Method. This method involves message distribution based on the encounter of mobile phones. We apply the scheme in [13] and let the sick user to send out messages when it encounters a susceptible person. In our simulation, once the sick person encounters with a susceptible person, the *vaccination* message is sent with the probability p_{va}, while the *alert* message is sent with the probability $1 - p_{va}$.

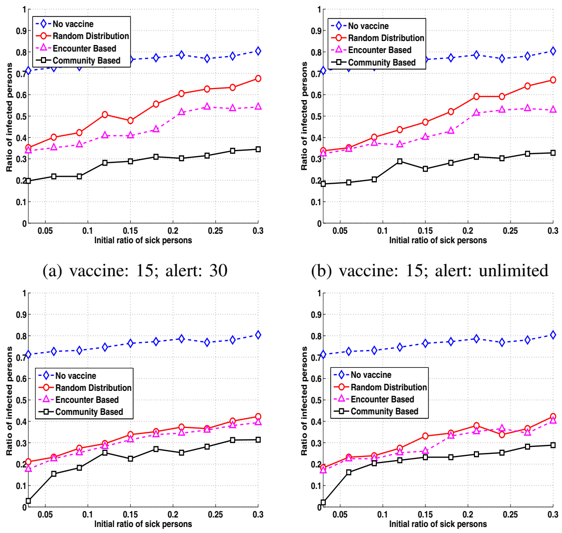
C. Effectiveness of Disease Propagation Control

In the first set of experiments, we evaluate the effectiveness of our social community based methods in terms of the final ratio of infected persons at the end of our test by comparing to existing methods of *Random Distribution* and *Encounter-based*. We use the MIT traces and vary both the recovery cycle N_p and the initial sick ratio. Figure 3 presents the final ratio of the infected persons versus the recovery cycle. The *No vaccine* is plotted as a baseline case.

The key observation is that our proposed community based method achieves a lower infection ratio than *Random Distribution* and *Encounter-based* methods for each initial sick ratio and each recovery cycle. This is very encouraging since the persons chosen by our community-based methods to have *vaccination* or *alert* messages interact more frequently with each other. Consequently, the proposed community based methods can control the disease propagation more effectively than other methods. We also found that the final infection ratio increases when the initial infection ratio or recover cycle increases for all the methods.

D. Impact of the Number of vaccination and alert Messages

Next, we change the available number of the *vaccination* and *alert* messages under different initial ratios of infected persons. The results from MIT traces are presented. Figure 4(a) and (b) depicted the results of 30 alert messages and unlimited alert messages respectively, when the available number of vaccines is 15, which is about 20% of the total number of people in the experiment. While Figure 4(c) and (d) presented the



(a) vaccine: 15; alert: 30

(b) vaccine: 15; alert: unlimited

(c) vaccine: 35; alert: 30

(d) vaccine: 35; alert: unlimited

Fig. 4. MIT traces: Performance comparison under different number of *vaccination* and *alert* messages with $p_{in} = 0.5$, $p_{al} = 0.7$, $p_{re} = 0.1$, $p_{im} = 0.2$, $p_{va} = 0.3$, $N_p = 4$ days.

results of 30 alert messages and unlimited alert messages respectively, when the available number of vaccines is 35, which is about 50% of the total number of people in the experiment. Again, we observed that our proposed community based method can achieve a much lower final infection ratio than the *Random Distribution* and *Encounter-based* methods under different number of *vaccination* and *alert* messages.

Furthermore, we found that there is an increasing trend of the infection ratio as we increase the initial ratio of sick persons. However, the final infection ratio decreases as the number of *alert* messages increases from 30 to unlimited. This is consistent with our expectation: more alert messages allow more people to take the necessary precautions, which reduce their chances of being infected, and hence reducing the number of total infected people.

In addition, comparing the results in Figure 4 under different vaccine numbers, we found that the performance difference between our proposed community based methods and other methods is smaller when increasing the vaccine number from 15 to 35. This further indicates that our proposed approach is more effective when the supply of vaccine is limited.

V. Conclusion

In this paper, we proposed a mobile phone enabled community based disease control framework, which utilizes human social relationship information to reduce the rate at which an infectious disease spreads in the healthcare domain. The extracted social community information is used for efficient vaccine distribution as opposed to the traditional random vaccine distribution. Our framework first partitions the set of encountered people into multiple communities and kernel structures based on their social relationships, where the people encountering information can be derived from traces collected by mobile phones. We believe people who are in the same kernel structure and community as a sick person have higher risks of being infected since they frequently interact with each other. Hence, these people will be chosen by our framework to receive vaccination or alert messages. We further developed a merging technique that helps to capture the dynamic community information so as to control the disease propagation more effectively. We compared our community based disease control method with existing techniques such as Random Distribution and Encounter-based methods using real contact-based traces such as the MIT reality trace. Our results showed that the propagation rate of an infectious disease can be significantly reduced by utilizing the social community information. Our study demonstrated more opportunities for utilizing social relationships information to support healthcare related applications.

VI. Acknowledgment

This work is supported in part by National Science Foundation Grants CNS1016303, CNS 1016296 and CCF1018270.

References

[1] J.Scott, *Social Network Analysis: A Handbook*. Sage Publication Ltd, 2000.

[2] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in *Proceedings of the National Academy of Sciences of the United States of America*, June 2002.

[3] B.W.Kernighan and S.Lin, "An efficient heuristic procedure for partitioning graphs," *in Bell System Technical Journal*, vol. 49, pp. 291–307, 1970.

[4] L. Tang, X. Wang, and H. Liu, "Uncovering groups via heterogeneous interaction analysis," in *Proceedings of IEEE International Conference on Data Mining(ICDM)*, 2009.

[5] F. Li, Y. Yang, and J.Wu, "Cpmc: An efficient proximity malware coping scheme in smartphone-based mobile networks," in *Proceedings of IEEE Infocom*, 2010.

[6] F. Li and J. Wu, "Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks," in *Proceedings of of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks(SECON)*, 2009.

[7] A. Perisic and C. T. Bauch, "Social contact networks and disease eradicability under voluntary vaccination," *In PLoS Computational Biology*, vol. 5, p. e1000280, 2009.

[8] S. Huang, "Probabilistic model checking of disease spread and prevention," in *Scholarly Paper for the Degree of Masters in University of Maryland*, 2009.

[9] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," *In Personal and Ubiquitous Computing*, vol. 10, no. 4, 2005.

[10] , "Morbidity and mortality weekly report," Mar. 2010. [Online]. Available: http://www.cdc.gov/mmwr/preview/mmwrhtml/mm5914a3.htm?s_cid=mm5914a3_e

[11] H. Yuan, G. Chen, J. Wu, and H. Xiong, "Towards controlling virus propagation in information system with point-to-group information sharing," *Decision Support Systems*, vol. 48, no. 1, pp. 57–68, 2009.

[12] M.E.J.Newman, "Detecting community structure in networks," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 38, pp. 321–330, March 2004.

[13] S. Tanachaiwiwat and A. Helmy, "Encounter-based worms: Analysis and defense," in *IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON) Poster/Demo Session*, 2006.