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 † Dept. of CSE, Lehigh University
Castle Point on Hudson, Hoboken, NJ 07030 Bethlehem, PA 18015
{yren2, jyang, yingying.chen}@stevens.edu *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. Our approach can be deployed using a centralized or distributed architecture. 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.

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 [3], and wireless networks, e.g., coping with the propagation of malware on smart phones [4], and facilitating the packet forwarding in Delay Tolerant Networks (DTNs) [5]. 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. In this work, we focus on building a mobile phone enabled social community based framework to reduce the rate at which an infectious disease spreads.

In the healthcare domain, the infectious disease is a clinically illness resulting from the presence of pathogenic microbial agents [6], [7]. Transmission of the infectious diseases such as SARS, bird flu and swine flu [8] can occur when people are in close proximity. For example, the air around a person with swine flu may contain H1N1 virus and infect the other people close-by [9]. The transmission of infectious diseases in public is a serious problem related to life or death and can cause panic in the whole society if not controlled effectively. Due to the typical characteristic of a slow start and then exponential propagation of the disease [10], mitigating an infectious disease at its early stage is critical and vaccination is a typical strategy. Because of the limited supply of vaccines and its relatively high cost when applying to a large population, how to efficiently distribute the vaccine and in the meanwhile achieving the goal of effective control of the disease propagation is an important problem.

Besides the traditional random vaccination strategy, recent work used bridge users identified in the human contact networks as distribution points of vaccination [11]. 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. Since many infectious diseases propagate via human interactions, the social communities derived from mobile phone proximity traces in our daily lives can be utilized to choose the set of people that need to be vaccinated or alerted such that we can mitigate the disease propagation more effectively and economically as opposed to randomly choosing any person to be vaccinated or alerted.

In light of these benefits, in this paper, we design a social community-based method that exploits the social relationships derived from mobile phone Bluetooth traces to reduce the rate at which an infectious disease spreads. Based on human's encounter events, multiple communities are derived and kernel structures are extracted. The community information may vary over time. Previous works in community extraction typically find communities over the whole trace and such static community information is then used for making decisions, e.g., selecting the appropriate relaying users for message dissemination [5]. However, static community extraction cannot capture the time-varying community information present in the trace. In our work, we propose to extract community structures at different time periods and then merge these extracted communities to capture the dynamic community information so as to control the disease propagation more effectively.

People who are in the same community or kernel structure are present in the close proximity more frequently and thus may interact more with each other, whereas those people across different communities imply fewer interactions. We believe those people within the same communities or kernel structures as the sick people have higher risks to get infected, and thus should be at least given disease alert messages and receive vaccine shots if available. Moreover, we develop a framework which supports two architectures, centralized and distributed, to utilize the dynamic social community information to control the disease spreads.

We experimentally evaluated our framework through a trace-driven approach by using the MIT reality mining trace [12] and the Italian trace [13]. The results showed that our strategy is highly effective for efficient vaccination to control disease propagation when comparing to methods without using social relationships and schemes merely utilizing static community information.

The rest of the paper is organized as follows. We first put our work in the context of current research in Section II. We then present our mobile phone enabled social community based framework in Section III. It describes the decentralized along with the distributed system models in our framework and the disease infection model used in this work. We next present our dynamic social community based scheme in Section IV. In Section V, 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 VI.

II. RELATED WORK

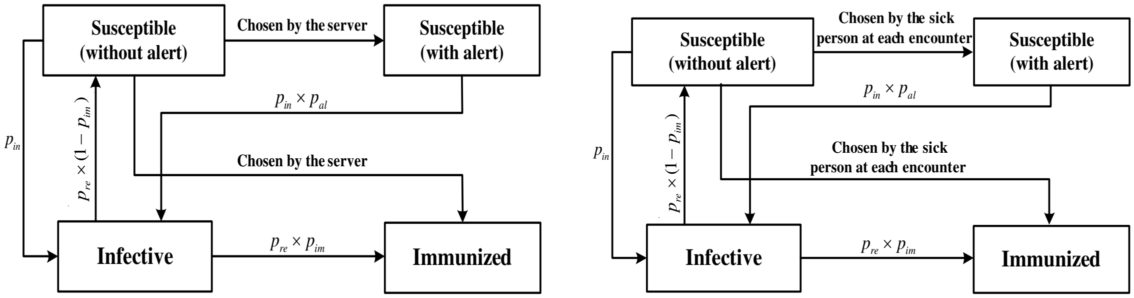
Group discovery and community detection have been an active research area. In [14], the Kernighan-Lin algorithm was introduced to improve the initial division of a network by optimizing the number of graph edges within and between the partitions using the greedy algorithm. [1] developed a sociological approach called hierarchical clustering. The idea behind this method is to develop a measure of similarity between each pair of vertices from the structure of the network and merge the communities with the highest similarity. The algorithm of Girvan and Newman [2] divided the network by iteratively removal of the edges. The betweenness metric is a centrality measure of a vertex within a graph. Vertices that occur on many shortest paths between other vertices have higher betweenness than those that do not. [15] further analyzed the computational cost of the betweenness metric in social networking applications.

Another important metric in community detection is modularity Q as described in [3], [16] where a larger modularity indicates more frequent within-group interaction. In general, one aims to find a community structure such that Q is maximized. On the other hand, instead of relying on a centralized server, [17] proposed distributed community detection, which makes mobile devices sense and detect their own local communities.

The active development of group discovery and community detection provides promising techniques for applying social relationships to support various application domains. [3] performed online group discovery in multi-dimensional networks obtained from various social media (e.g., YouTube and Flickr). [18], [19] conducted social network analysis in Delay Tolerant Networks by utilizing betweenness and similarity metrics. Moreover, the social community structures were utilized to cope with the propagation of malware on smartphones in mobile networks as proposed in [4], and [10] developed a social network based patching scheme for effectively limiting the spread of MMS and SMS based worms in cellular networks.

However, little work has been done in applying social community structures to effectively control the disease propagation through vaccination in the healthcare domain. [20] 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. [21] proposed to study contact networks, where each person's role in a population is treated as distinct, i.e., a heterogeneous population. It suggested that by restricting the contacts themselves, one can also limit disease spread effectively. This would correspond to deleting edges in the modeled contact network.

[11] further considered the modeling of the disease spread over populations. It proposed that vaccinating the groups of more sociable persons can prevent a larger number of infectious than if administering the same number of vaccinations to random members of the population. None of these works have systematically investigated the



(a) Centralized architecture (b) Distributed architecture

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

effectiveness of exploiting social community structures for efficient vaccination such that the propagation rate of an infectious disease is 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.

III. FRAMEWORK OVERVIEW

In this section, we first provide the system model for our mobile phone enabled disease control framework, and present descriptions of both the centralized and distributed 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 [22].

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 [12]. 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 IV. Moreover, the extracted community information will be stored at the server and updated from time to time.

2) Centralized vs. Distributed 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. To protect the useres' privacy, such messages will be sent anonymously so that the receivers do not know who the senders are. The full discussion of the privacy issue is out of the scope of this paper and will be included in our future work.

When actions need to be taken for an infectious disease, in the centralized architecture, 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.

On the other hand, in the distributed architecture, each user who has already been infected by the disease will download the community information related to his device ID from the server. We note that although the user based community information only contains a partial view of the whole community information stored in the server, the memory requirement of storing such partial community information is much less than that of storing the whole community information. This makes it applicable to store the partial community view in individual mobile devices. Based on the downloaded social community information,

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.

each sick user will then decide on whether to send out a *vaccination* message or a *alert* message when he encounters the next person, i.e., a new person discovered by the sick person's mobile device.

The distributed approach is a challenging architecture as it requires peer-to-peer communications over an ad hoc wireless network. Moreover, in our future work, we plan to develop distributed social community extraction scheme so that each user has the capability to derive his local community and does not rely on the centralized server.

B. Infection Model

In our framework, we extend the standard epidemic SIR model [23] 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. We depicted the state transition diagram of our infection model in Figure 1 and list all possible state transitions as follows:

(1) From *Susceptible (without alert)* to *Immunized*: In the centralized architecture, the server will choose the users who should receive the *vaccination* messages based on the extracted social community information and send the *vaccination* messages to these users directly. Whereas in the distributed architecture, the user who is already infected will decide on who should receive the *vaccination* messages based on the social community information downloaded from the server and the next set of people he will encounter.

(2) From *Susceptible (without alert)* to *Susceptible (with alert)*: Likewise, in the centralized architecture, the server chooses the users who should receive the *alert* messages based on the extracted social community information and sends the *alert* messages to these users directly. Whereas in the distributed architecture, the user who is already infected will determine who should receive the *alert* messages based on the social community information downloaded from the server and the next set of people he will encounter.

(3) From *Susceptible (without alert)* to *Infective*: If a user encounters a sick person, the user has a probability p_{in} of being infected.

(4) From *Susceptible (with alert)* to *Infective*: Since precautions will be taken for users at the *Susceptible (with alert)* state, the probability of such a user get infected when he encounters a sick person is reduced to $p_{in} \times p_{al}$.

(5) From *Infective* to *Immunized*: An infected user moves to the *Immunized* state after he has recovered from the disease with immunity. This happens with a probability of $p_{re} \times p_{im}$ every recovery cycle (set to N_p days).

(6) From *Infective* to *Susceptible (without alert)*: An infected user moves to the *Susceptible (without alert)* state if he has recovered from the disease, but not immunized from being infected again. Based on item (5), this probability should be $p_{re} \times (1 - p_{im})$ for every recovery cycle.

C. Analysis of State Transitions

The epidemic infection model includes rules on how the members of users transit from one state to another. Directly analyzing the state transitions within the social community model is prohibitive. Thus, we analyze the state transitions based on users' encounter events. Because we believe that this analysis can provide us with an upper bound of the final infected ratio for our dynamic community based system since the contact rate is assumed to be constant in the analytical model, although in real system this rate decreases as more people gets infected or vaccinated.

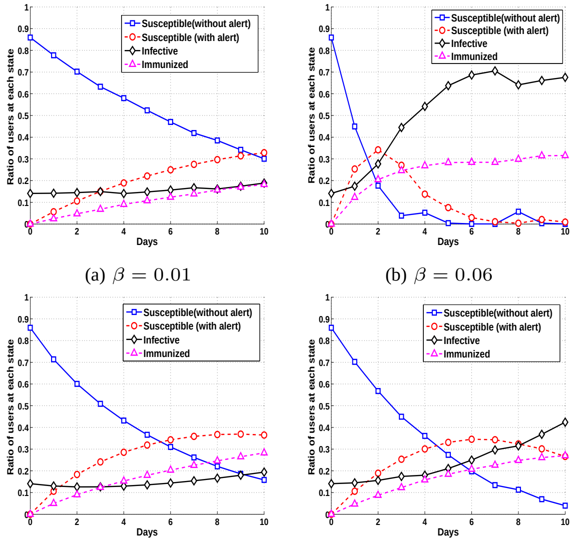
We assume that the total number of users N_{total} will not change and users only transit from one state to another state. Let $S_1(t)$ and $S_2(t)$ denote the number of users in the state *Susceptible (without alert)* and *Susceptible (with alert)*, respectively. $I(t)$ denotes the number of users in the state *Infective* and $R(t)$ denotes the number of users *Immunized*. Time t is taken as a variable here. We assume that the encounters between infected users and other ones occur at an average rate of β, which is often referred to as the contact rate. The state transitions can be represented through the following equations:

$$\begin{aligned}\frac{S_1(t+\Delta t)-S_1(t)}{\Delta t}=&-\beta\cdot I(t)\cdot S_1(t)\cdot(1-p_{va})-\beta\cdot I(t)\cdot S_1(t)\cdot p_{va}\\&+p_{re}\cdot(1-p_{im})\cdot I(t)\cdot\delta(t-N_p\cdot N)\end{aligned}\tag{1}$$

$$\frac{S_2(t+\Delta t)-S_2(t)}{\Delta t}=\beta\cdot I(t)\cdot S_1(t)\cdot(1-p_{va})-\beta\cdot I(t)\cdot S_2\cdot(t)\cdot p_{in}\cdot p_{al}\tag{2}$$

$$\begin{aligned}\frac{I(t+\Delta t)-I(t)}{\Delta t}=&\beta\cdot I(t)\cdot S_2(t)\cdot p_{in}\cdot p_{al}-p_{re}\cdot p_{im}\cdot I(t)\cdot\delta(t-N_p\cdot N)\\&-p_{re}\cdot(1-p_{im})I(t)\delta(t-N_p\cdot N)\end{aligned}\tag{3}$$

$$\frac{R(t+\Delta t)-R(t)}{\Delta t}=\beta\cdot I(t)\cdot S_1(t)\cdot p_{va}+p_{re}\cdot p_{im}\cdot I(t)\cdot\delta(t-N_p\cdot N)\tag{4}$$



(a) $\beta = 0.01$ (b) $\beta = 0.06$

(c) $N_p = 1$ day, $\beta = 0.02$ (d) $N_p = 4$ days, $\beta = 0.02$

Fig. 2. Ratio of the users at different states when varying recovery cycle and contact rate.

In the equations above, the $N = 1, 2, 3, ...$ and the function $\delta(t)$ is defined as:

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0; \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

For the above equations, we also have:

$$S_1(t) + S_2(t) + I(t) + R(t) = N_{total}. \quad (6)$$

D. Analytical Results

Based on the above state transition analysis, we vary the parameters in the infection model to provide an illustration of the number of users at each state as a function of time. We first vary the contact rate β to study its impact: we consider a contact rate of 0.06 and 0.01. Based on the results depicted in Figure 2 (a) and (b), we concluded that the infected ratio remained low for the whole duration when users meet less frequently, while the number of infected people increases significantly if users meet more frequently. This indicates that our infection model can precisely capture the propagation trend of an infectious disease.

We then set the contact rate $\beta = 0.02$ based on the mean value observed from the MIT reality trace [12] and examine the number of users at different states when the initial ratio of the sick person is set to 0.15. Figure 2 (c) and (d) plotted the results for 10 testing days when recovery cycle is 1 day and 4 days respectively. We observed that the ratio of the infected persons increases as the time moves along. In particular, it increases to 0.2 and 0.45 for 1 day and 4 days recovery cycle at the end of the testing period.

IV. 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 3. 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. 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 when the next request coming from the server or from an individual user based on the specific architecture used in our framework.

In 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. Dividing Based Contact Graph Construction

The whole contact trace is divided into multiple trace files. We assume that each trace file consists of recorded encounter events that happened during the 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

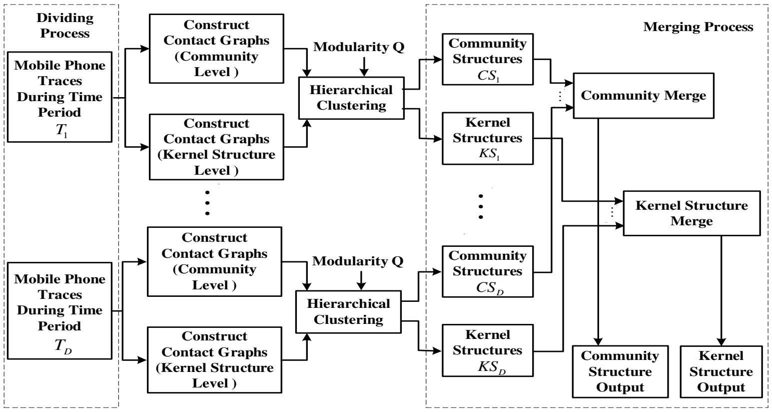


Fig. 3. Flow overview: components to extract community information.

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.

An illustration of community and kernel structures is shown in Figure 4. Two communities can be extracted using a contact graph. Community 1 consists of users {A, B, C, D, H} while Community 2 consists of users {E, F, G, I}. The communities are shown as dotted circles. Moreover, by using a higher W, three sets of users, namely $\{A, B, C\}$, $\{E, F, G\}$ and $\{D, H\}$ are found having higher encounter frequencies within the same set than the rest of users. Thus, they form 3 different kernel structures (shown with solid circles).

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 [24]. Further, to verify whether a particular division is meaningful or not, we use the modularity metric, Q [24]. 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:

$$A_i = A_i^1 \cup A_i^2 \cup ... \cup A_i^{k_i} \tag{7}$$

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:

$$\frac{|A_i^j \cap A_{i+1}^l|}{Max(|A_i^j|, |A_{i+1}^l|)} > \tau \tag{8}$$

- 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.

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 and 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.

Centralized Community Based Algorithm. As described in Section III, the server will decide who to receive *vaccination* or *alert* messages. The flow of the centralized 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 in $V_c \bigcup V_k$, the remainder messages are held till the next update of the community information as new persons may appear.

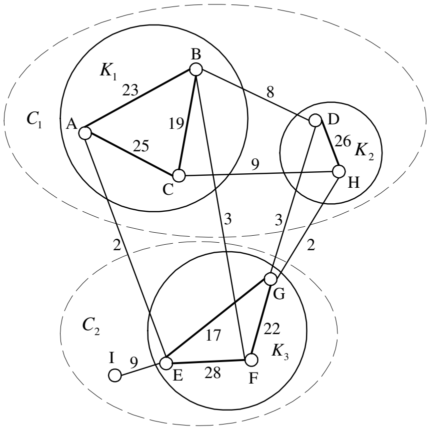


Fig. 4. An example of social community.

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

Distributed Community Based Algorithm. The sick person i downloads his community set V_c^i and kernel structure set V_k^i information from the server. In the meantime, the sick person will check the availability of the *vaccination* and *alert* messages with the server. We note that the community and kernel structure sets may also be possibly determined by users themselves, e.g., [17]. This issue is further explored in our future work. The sick person then performs the following:

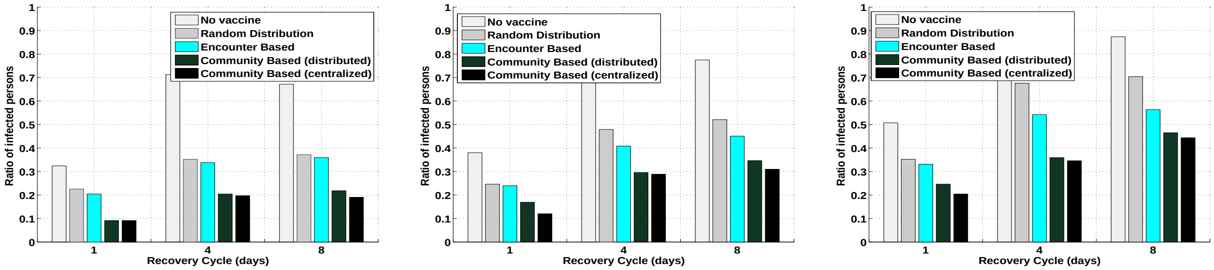
- The sick person sets the persons who are in V_k^i as candidate user list for sending out the *vaccination* messages.
- The sick person sets the persons who are in V_c^i as candidate user list for sending out the *alert* messages.
- The sick person sends out corresponding messages when he encounters another user who belongs to the above candidate lists. We note that if there are not enough *vaccination* messages for the people in V_k^i, the *alert* messages will be sent instead.

V. 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 two human contact-based traces, namely the MIT reality [12] and Italian [13] traces. Both traces were collected using smart phones equipped with



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

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. In the Italian traces, there are 44 people who carried the smart phones and the experiment lasted for 19 days. Similarly, we used the first 9 days of the trace as training data to extract the communities and kernel structures, and the remaining 10 days to evaluate our approach. We conducted extensive experiments on these two sets of traces by varying different parameters in our epidemiology infection model. 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 [25] 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 $1 - p_{va}$.

Betweenness-based Community Method. This method is an improvement over *Random Distribution*. The concept of the betweenness [2] is used to identify a set of key users that act as bridge users in a contact network [11]. The extraction of bridge users is based on the whole trace and thus is static. Each susceptible person is ranked based on the betweenness metric, and then the persons with higher ranks will be chosen to receive *vaccination* messages while the persons with lower ranks will be chosen to receive *alert* messages. This approach has an advantage of obtaining a set of more active users from the population than *Random Distribution*.

C. *Effectiveness of Disease Propagation Control*

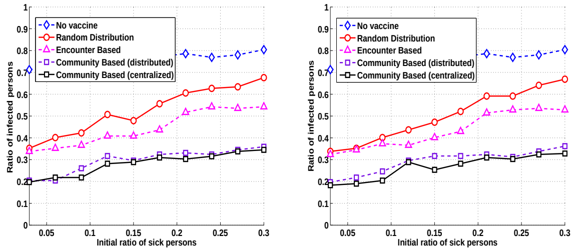
In the first set of experiments, we evaluate the effectiveness of our social community based methods in terms of the final ratio of infected persons at the end of our test by comparing to existing methods of *Random Distribution* and *Encounter-based*. We use the MIT traces and vary both the recovery cycle N_p and the initial sick ratio. Figure 5 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 methods, both centralized and distributed, consistently achieve a lower infection ratio than *Random Distribution* and *Encounter-based* methods for each initial sick ratio and each recovery cycle. This observation is also inline with our analytical results depicted in Figure 2 (a) and (b). 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.

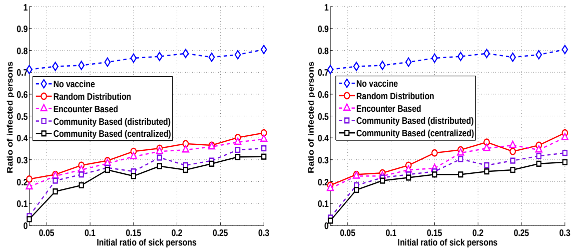
Moreover, we found that the performance of our centralized community based method is better than its distributed version. This is because in the centralized approach, the distribution of *vaccination* or *alert* messages is based on the complete picture of the social community information, instead of the partial local community information in the distributed case.

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



(a) vaccine: 15; alert: 30 (b) vaccine: 15; alert: unlimited



(c) vaccine: 35; alert: 30 (d) vaccine: 35; alert: unlimited

Fig. 6. 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.

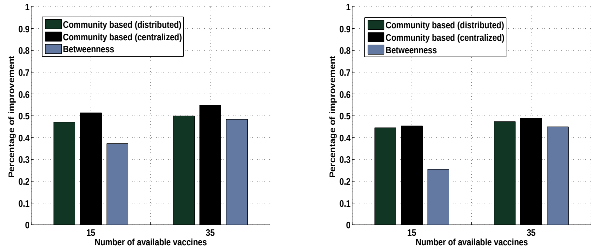
of infected persons. The results from MIT traces are presented. Figure 6(a) and (b) depicted the results of 30 alert messages and unlimited alert messages respectively, when the available number of vaccines is 15, which is about 20% of the total number of people in the experiment. While Figure 6(c) and (d) presented the 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 6 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.

E. Comparison with the Static Betweenness Method

In Figure 7, we compare the results obtained from our community-based approach and the static betweenness based community method (shortened as *Betweenness*). The y-axis shows the improvement in the final infection ratio for different methods when compared to the *No vaccine* scenario, whereas the x-axis shows the number of vaccines used in our community-based methods and *Betweenness*. The results clearly showed that our community based approach outperforms the static betweenness based method, especially when there are fewer number of vaccines is available (i.e., the case with 15 vaccines) for both initial sick ratios of 0.09 and 0.24. This is because *Betweenness* selects the relaying users for message dissemination from static community information. However, static community information cannot capture the time-varying community information present in the trace. On the contrary, our community based method uses dynamic community information produced from our community extraction process, and hence it can control the disease propagation more effectively.



(a) Scenario 1: Initial sick ratio: 0.09 (b) Scenario 2: Initial sick ratio: 0.24

Fig. 7. MIT traces: Comparison of community based methods and the static betweenness method under different number of vaccines when there are 30 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.

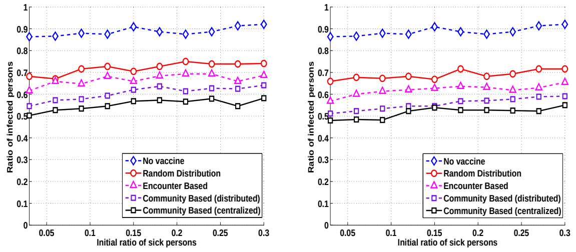
F. Results from Italian Traces

Finally, we present our study using the Italian traces. Due to the space limitation, only the key results are presented. We changed the available number of the *vaccination* and *alert* messages using different initial ratios of infected persons and examined the final ratio of infected persons. The results depicted in Figure 8 exhibit the same trend as we observed when using MIT traces: our social community based methods outperforms the existing methods.

Comparing these results with those obtained from MIT traces in Figure 6, we further observed that the performance difference between our proposed community based methods and the existing methods is relatively smaller in the Italian traces. We found that this is because there is a large group of people who always encounter one another in the Italian traces and hence they belong to the same kernel structure and community. Thus, our community based methods can not distinguish further among these people when they attempt to choose a subset of appropriate people for receiving the *vaccination* or *alert* messages.

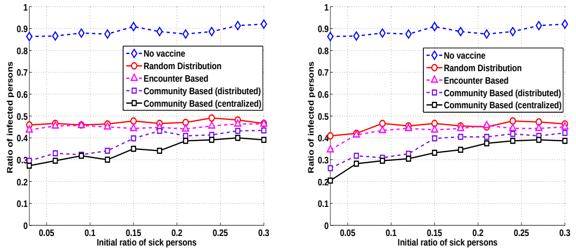
VI. CONCLUSION

In this paper, we proposed a mobile phone enabled community based disease control framework, which utilizes human social relationship information to reduce the rate



(a) vaccine: 9; alert: 18

(b) vaccine: 9; alert: unlimited



(c) vaccine: 22; alert: 18

(d) vaccine: 22; alert: unlimited

Fig. 8. Italian 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..

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 and Italian traces. Our results showed that the propagation rate of an infectious disease can be significantly reduced by utilizing the social community information. In addition, we compare our approach that utilizes dynamic community information with a betweenness-based approach using static community information. The results also confirmed our observation that our community based method is more effective in achieving a lower infection ratio. Our study demonstrated more opportunities for utilizing social relationships information to support healthcare related applications.

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 *in Proceedings of the National Academy of Sciences of the United States of America*, June 2002.

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

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

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

[6] K. J. Rothman, S. Greenland, and T. L. Lash, *Modern Epidemiology*. lippincott williams and wilkins, 2008.

[7] R.M.Anderson and R.M.May, *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, 1992.

[8] , “Introduction to influenza,” Nov. 2010. [Online]. Available: http://en.wikipedia.org/wiki/Influenza

[9] , “Public health fact sheet,” Nov. 2009. [Online]. Available: www.mass.gov/Eeohhs2/docs/dph/cdc/factsheets/swine_flu.pdf

[10] Z. Zhu, G. Cao, S. Zhu, S. Ranjan, and A. Nucci, “A social network based patching scheme for worm containment in cellular networks,” in *in Proceedings of IEEE Infocom*, 2009.

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

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

[13] S. Gaito, E. Pagani, and G. P. Rossi, “Opportunistic forwarding in workplaces,” in *in Proceedings of the Second ACM SIGCOMM Workshop on Online Social Networks*, 2009.

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

[15] U. Brandes, “A faster algorithm for betweenness centrality,” *In Journal of Mathematical Sociology*, vol. 25, pp. 163–177, 2001.

[16] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *in Physical Review E*, vol. 69, no. 2, 2004.

[17] P. Hui, E. Yoneki, S. yan Chan, and J. Crowcroft, “Distributed community detection in delay tolerant networks,” in *in Proceedings of Sigcomm Workshop MobiArch*, 2007.

[18] E. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in *Proceedings of the ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, 2007.

[19] M. Chuah and A. Coman, “Identifying connectors and communities: Understanding their impacts on the performance of a dtn publish/subscribe system,” in *in Proceedings of International Conference on Computational Science and Engineering*, 2009.

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

[21] L. A. Myers, “Contact network epidemiology: Bond percolation applied to infectious disease prediction and control,” *American Mathematical Society*, vol. 44, no. 1, pp. 63–86, 2007.

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

[23] 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.

[24] 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.

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