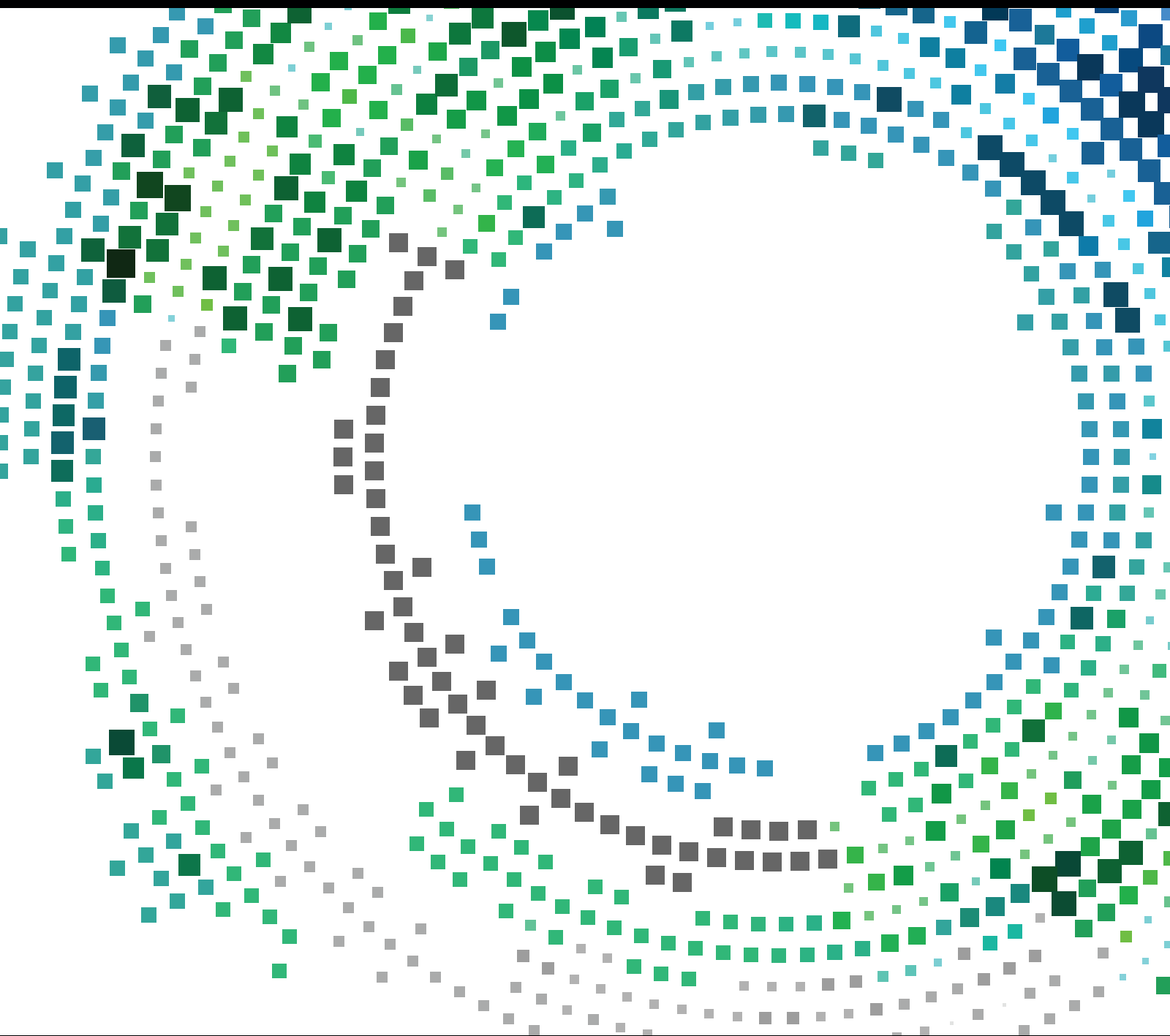
Mobile Information Systems

Big Data Management and Analytics for Mobile Crowd Sensing

Guest Editors: Tingting Chen, Fan Wu, Tony T. Luo, Mea Wang, and Qirong Ho



Big Data Management and Analytics for Mobile Crowd Sensing

Mobile Information Systems

Big Data Management and Analytics for Mobile Crowd Sensing

Guest Editors: Tingting Chen, Fan Wu, Tony T. Luo, Mea Wang, and Qirong Ho

Contents

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 8731802, 2 pages
http://dx.doi.org/10.1155/2016/8731802

Editorial

Big Data Management and Analytics for Mobile Crowd Sensing

Tingting Chen,[1] Fan Wu,[2] Tony T. Luo,[3] Mea Wang,[4] and Qirong Ho[3]

[1]*California State Polytechnic University, Pomona, CA 91768, USA*
[2]*Shanghai Jiao Tong University, Shanghai 200240, China*
[3]*Institute for Infocomm Research, A*STAR, Singapore*
[4]*University of Calgary, Calgary, AB, Canada T2N 1N4*

Correspondence should be addressed to Tingting Chen; tingtingchen@cpp.edu

Received 23 June 2016; Accepted 26 June 2016

With the fast increasing popularity of mobile smart devices, mobile crowd sensing has become a new paradigm of applications that enables the ubiquitous mobile devices with enhanced sensing capabilities, such as smartphones and wearable devices, to collect and to share local information towards a common goal. Most of the smart devices are equipped with a rich set of cheap and powerful sensors, including accelerometer, digital compass, GPS, microphone, and camera. These sensors can be utilized to monitor mobile users' surrounding environment and infer human activities and contexts. In recent years, a wide variety of applications have been developed to realize the potential of crowd sensing throughout everyday life, such as environmental, noise pollution assessment, road and traffic condition monitoring, road-side parking statistics, and indoor localization. The data acquired through mobile crowd sensing exhibits a number of important characteristics, such as being large in scale (volume), being fast generated (velocity), being different in forms (variety), and being uncertain in quality (veracity). The 4 Vs of crowd sensing data make it extremely interesting and challenging in designing participatory and opportunistic sensing technologies, human centric data management and analytics models, and novel visualization tools.

This special issue is composed of seven original research papers, carefully selected based on their merit contents. These works cover a variety of topics, including data sharing, compressed sensing protocols, privacy protection, cooperation issues, and application studies.

The paper "Share the Crowdsensing Data with Local Crowd by V2V Communications" by C. Song et al. investigates the communication and sharing of crowd sensing data by vehicles near the events. In a local crowd formed by vehicles, vehicles can transmit the data to each other by vehicle-to-vehicle (V2V) communication. This approach based on the vehicle-to-vehicle communications has a lower delay than the offloading-based approach.

Also on the topic of data sharing, the paper "ODMBP: Behavior Forwarding for Multiple Property Destinations in Mobile Social Networks" focuses on making the information sharing more effective among people with similar interests, by profiling the users' behavior in the mobile social network.

The paper "A Perturbed Compressed Sensing Protocol for Crowd Sensing" by Z. Zhang et al. proposes a data collection protocol for compressed sensing in wireless sensor networks. The protocol can protect the data confidentiality and is also time-efficient.

We have two excellent papers addressing the privacy issues. In the paper "How Dangerous Are Your Smartphones? App Usage Recommendation with Privacy Preserving," the authors K. Zhu et al. work on evaluating the mobile App privacy violation of mobile users by computing the danger coefficient. To help users reduce the privacy leakage, both the user preference to mobile Apps and the privacy risk are used in combination. The paper presents a mobile App usage recommendation method called AppURank to recommend secure Apps with the same function.

On the other hand, the paper "Privacy Leakage in Mobile Sensing: Your Unlock Passwords Can Be Leaked Through Wireless Hotspot Functionality" explores the snooping attack on smartphones leveraging the wireless hotspot functionality. The attacker leverages the impacts of finger motions on the

wireless signals during the unlocking period to analyze the passwords/patterns.

In the aspect of cooperation, the paper "Cooperation Dynamics on Mobile Crowd Networks of Device-To-Device Communications" has contributions in exploring the cooperation dynamics in mobile crowd networks by considering the elemental characteristics of crowd population, individual's mobility, and reciprocity policy. In particular, the authors model the cooperative behaviors in a mobile crowd into an evolutionary prison dilemma game and investigate the relationships between cooperation rate and some main influence factors.

Last but not least, the paper "Outdoor Air Quality Level Inference via Surveillance cameras" by Z. Zhang et al. presents an interesting application work of a novel air quality level inference approach based on outdoor images utilizing surveillance cameras. The proposed approach first extracts features from images and adopts multikernel learning to learn an adaptive classifier for air quality level inference. The contributions also include an Outdoor Air Quality Image Set (OAQIS) dataset, which contains high quality registered and calibrated images with rich labels.

We believe that the original works presented in this special issue would significantly contribute to the literature and the authors' innovative insights can influence the future work of people from academia and industry who are interested in the covered areas.

Acknowledgments

We would like to thank the authors for their valuable works. We are grateful to all the anonymous reviewers who carefully reviewed the assigned papers and provided valuable comments to further improve the paper quality.

Tingting Chen
Fan Wu
Tony T. Luo
Mea Wang
Qirong Ho

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 6804379, 10 pages
http://dx.doi.org/10.1155/2016/6804379

Research Article

How Dangerous Are Your Smartphones? App Usage Recommendation with Privacy Preserving

Konglin Zhu,[1] Xiaoman He,[1] Bin Xiang,[1] Lin Zhang,[1] and Achille Pattavina[2]

[1]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy*

Correspondence should be addressed to Konglin Zhu; klzhu@bupt.edu.cn

Received 30 December 2015; Revised 31 March 2016; Accepted 24 May 2016

Academic Editor: Mea Wang

With the rapid proliferation of mobile devices, explosive mobile applications (apps) are developed in the past few years. However, the functions of mobile apps are varied and the designs of them are not well understood by end users, especially the activities and functions related to user privacy. Therefore, understanding how much danger of mobile apps with respect to privacy violation to mobile users is becomes a critical issue when people use mobile devices. In this paper, we evaluate the mobile app privacy violation of mobile users by computing the danger coefficient. In order to help people reduce the privacy leakage, we combine both the user preference to mobile apps and the privacy risk of apps and propose a mobile app usage recommendation method named AppURank to recommend the secure apps with the same function as the "dangerous" one for people use. The evaluation results show that our recommendation can reduce the privacy leakage by 50%.

1. Introduction

The rapid growth of mobile devices has been leading to the prosperity of mobile applications (apps). For example, as of the end of 2014, the number of apps on Play Store has been over 1.4 million and over 1.2 million on Apple Store. This number is still growing dramatically with the proliferation of mobile devices. Mobile users download these mobile apps and use them on their mobile devices to satisfy different purposes. However, the functions of mobile apps are varied and the designs of them are not well understood by end users, especially the activities and functions related to user privacy. Indeed, to improve user experience and the functionality of mobile apps, developers start to move their eyes on the personalized service that can be provided by apps. They develop new functional apps or enhance the capability of apps by digging into the personal information, such as location information, contacts, camera, messaging, and even calling service. However, when users launch apps on their mobile devices, they may fall into danger as some unknown activities or functions might cause privacy issue.

Although app stores (e.g., Google Play) may remove those apps with malfunctions or low quality periodically, many well developed apps with privacy violations are not perceived by the stores. In other words, app stores release the right to end users to let them decide whether to install the apps or not. In such case, most end users download and install those apps by neglecting the privacy warning. Even when some users do notice the privacy issue, as the functionality and preference of mobile apps, they still install them into the mobile device. Once the apps are installed, users' privacy information will be leaked when they launch those apps.

As mobile apps serve for different functionalities, different types of privacy information may be leaked by users for launching different types of mobile apps. For instance, a location-based service (LBS) needs to collect user real time location information from users. It may refer to home and workplace where users may not expect to be exposed. A social-aware service needs to extract contacts from users, which violates many users' personal life as well, not to mention the information collected for function-unrelated purpose. In fact, it is reported that users have growing

concern about their privacy while using mobile apps. A recent survey from IDG news reveals that over 30% of mobile users prefer to uninstall those apps after learning the personal information they collected. Unfortunately, users do not know how much of the personal information has been collected and how much danger of mobile apps with respect to privacy violation to mobile users. Therefore, it is crucial to understand how dangerous are installed apps on mobile devices as well as how much privacy risk is taken by launching those apps. It will also be beneficial for mobile users to know how to reduce the privacy leakage by using those apps with less privacy concern and meanwhile maintain the quality of experience.

Thus far, majority of mobile app recommendation approaches have been developed based on the popularity of apps while neglecting the privacy issues existing in mobile apps [1–3]. Several privacy-concerned app detection and recommendation mechanisms are proposed to discover the malfunction of mobile apps. They either focus on the service provider side to let app stores recommend those apps with less privacy concern without considering the user personalization [4] or target the developer side to investigate the inside of apps, find malware code, and restrict the app access permission [5–9]. However, these actions need the cooperation with either service providers or apps developers, which makes them difficult to be implemented in practice.

In this paper, in order to measure how much privacy a mobile user attempts on the mobile phone and help to recommend apps with less privacy, we propose a privacy evaluation mechanism by analyzing the app usage data. The violation of privacy depends not only on the risks of apps, but also on the user usage pattern. Although some apps violate user privacy heavily, user information cannot leak if the app has not been used. To evaluate the privacy leakage, we define a danger coefficient to quantify the privacy and analyze the privacy violation distribution of mobile users. To reduce the privacy leakage from mobile devices, on one hand, we need to understand how much privacy a mobile app can expose. On the other hand, we need to investigate the apps usage of different users. Therefore, we combine both the user preference to mobile apps and the privacy risk of mobile apps for apps recommendation. To formulate the user preference to mobile apps, we apply the distribution of common preference probabilistic method, which can enrich the context of personalized preference. To understand the privacy violation of mobile apps, we measure the privacy access permissions of mobile apps. Finally, we seek a balance between the user preference to apps and the privacy violation of apps to propose an app usage recommendation method named AppURank. According to the functionality of mobile apps, we classify them into groups by the topic model. The proposed approach recommends the same functional apps with lighter privacy concern and high user preference. We evaluate our mobile app recommendation approach with extensive experiments. The results show that our proposed recommendation method can halve the danger of mobile devices and meanwhile maintain the same level of user preference.

The contributions of the paper are summarized as follows:

(i) We carry out a mechanism to evaluate the extent of privacy leakage when people use their mobile devices. We define a danger coefficient to measure the privacy violation of apps from usage perspective and analyze the privacy violation distribution of mobile users.

(ii) We propose an app usage recommendation approach for end users, named AppURank, by combining user preference, privacy risks, and functionality of apps. The proposed method is to recommend people with preferred apps but with less privacy violation.

(iii) We evaluate the app recommendation method on our collected data. It shows that the method can reduce the danger of mobile devices to half and meanwhile maintain the same level of usage preference.

The rest of the paper is structured as follows: Section 2 reviews the related literature. Section 3 describes the problem, introduces the definition of danger coefficient of mobile users, and provides a recommendation on the app usage that can minimize the risk of privacy leakage. Section 4 shows the experimental results. Conclusions are finally given in Section 5.

2. Related Literature

In this section, we review the state of the art for the privacy leakage evaluation of mobile apps and mobile app recommendation approaches.

The privacy issue of mobile apps has been studied for many years. One group of previous studies regarding privacy issue of mobile apps concerns the risk analysis of mobile apps. For instance, Au et al. [6] surveyed the permission systems of smartphone operating systems from the amount of controls, the information released to users, and the levels of interactivity from users. Felt et al. [5] focused on the permission request of various mobile apps to determine whether Android developers follow least privilege with their permission requests. They further built Stowaway to detect overprivilege in Android apps. Enck et al. [7] proposed TaintDroid, which provided real time analysis of mobile apps on the monitoring of their data access by leveraging Android's virtualized execution environment. In contrast, majority of mobile app privacy studies are proposed for privacy violation or mobile app malfunction detection. To deal with the information stealing of mobile apps, Zhou et al. [8] carried out the TISSA system, which can empower users to flexibly control the accessibility of mobile apps to personal information. Enck et al. [9] exploited a rule-based certification model named Kirin to perform lightweight certification of mobile apps at installing time to reduce the privacy issue caused by mobile apps.

However, these mechanisms need investigation on the code installed in mobile apps in order to identify the privacy violation, which is difficult for all apps installed on mobile devices. Meanwhile, people do not like apps scanning their mobile devices all the time. In this paper, we propose the

privacy preserving method by recommendation approach, which can avoid the installation of risky apps.

The majority of mobile app recommendation methods consider the popularity or user preference as factors for the decision making. For instance, AppJoy [1] made personalized mobile app recommendation by analyzing how users actually use their installed apps. They applied collaborative filtering algorithm for individual recommendation. Yu et al. [2] and Zhu et al. [3] considered the user context for mobile app recommendation and used Latent Dirichlet Allocation (LDA) topic model to describe the problem of mobile apps recommendation. Few papers focus on the mobile apps based on privacy issue. Peng et al. [10] proposed a risk ranking method of Android apps using probabilistic generative model to tell users the privacy risk of mobile apps before installation. The other one by Zhu et al. [4] used the modern portfolio theory to recommend mobile apps from the perspective of app store by considering the awareness of security and privacy. Our paper tries to recommend mobile apps from personalized user-app usage perspective to avoid privacy violation. There are some literatures focusing on permission settings recommendation to preserve user privacy. Lin et al. [11] proposed to provide reasonable default settings to help users configure their privacy settings by identifying distinct privacy profiles. Liu et al. [12] proposed and implemented PriWe, which leveraged the crowd sourced permission settings to understand users' privacy expectation and provides app specific recommendations to mitigate information leakage.

We use the similar methods as [10] to obtain the privacy risk of mobile apps. However, different from the state of the art [4, 10], we not only consider the privacy risk of mobile apps in general, but also combine the function relativity of mobile apps to the permissions and people usage patterns of mobile apps to evaluate the privacy danger of mobile apps to users. We then propose a mobile app recommendation method by considering both user preferences and mobile app privacy.

3. Problem Formation

3.1. Preliminaries. When an app is installed or launched on mobile device, it always asks the permission to access certain information. The permission means the capability users grant to mobile apps so that mobile apps can access certain part of mobile users' information. These permissions are associated with mobile apps to either help the mobile apps to achieve some functions (e.g., localization) or fulfill the mobile apps to collect user data. The permissions requested by a mobile app are independent of each other. When mobile app is installed or launched on the smartphone, the users have the right to make decision for the app to access the permission of information. In fact, the information those apps intend to access may refer to the sensitive personal private data, such as location information and the control of hardware (e.g., camera). For instance, Table 1 illustrates the access permission list requested by a version of Google search app in an app store. It can be seen that some information requested by the app permissions, such as network connection and storage, is necessary for the function of the app (i.e., information searching), referred here to as "function-related" for the app. At the same time some other information requested by the app permissions, such as phone state, personal data, payment service, and system tools, is not highly correlated with the function of the app, referred here to as "function-unrelated" for the app. As a matter of fact, for different types of permissions, the degree to which they violate the privacy is different. For instance, as shown in Table 1, the payment service permission that allows running the operation for payment is more severe than the permission of network connection. There are two reasons for such a judgement. First, the payment service permission may cause the economic loss, which is more vital than the network connection. Second, the payment service is function-unrelated while network connection is function-related for searching function. Therefore, all the information involved in an app can be categorized into several tiers according to its degree of privacy violation and its relativity to app function: (a) "normal permission," which does not involve sensitive information of mobile users, such as network connection and storage; (b) "severe permission," meaning the information is severely related to user privacy, such as personal information, location, and payment service; and (c) "system permissions," related to the control of hardware and system, such as the access of hardware and setup of system level configuration. Therefore, combining the function and the extent of privacy violation, six different types of information permission are considered: (1) function-related with normal permission, (2) function-related with severe permission, (3) function-related with system permission, (4) function-unrelated with normal permission, (5) function-unrelated with severe permission, and (6) function-unrelated with system permission.

Table 1: Access permission of a search engine app.

Permission	Description
Network connection	Allow the app to access the Internet
Storage	Allow accessing external SD card
Phone state	Allow accessing phone information
Personal information	Allow accessing contact information, messages, emails, and so forth
Location	Allow accessing the geographical information instantly
Hardware	Allow accessing camera, audio, and recorder
Payment service	Allow running the operation for payment
System tool	Allow setting up the display

Although apps violate user privacy by permissions of accessing user information, mobile users still install and launch different types of mobile apps on their smartphones according to their preferences. In order to recommend apps by considering both user privacy preserving and user preference, in this paper, we discuss three issues: (1) how to measure the privacy risk of an app and the privacy violation to mobile users by launching mobile apps, (2) how to determine

user preference to mobile apps, and (3) how to balance the privacy violation to mobile users and the user-app preference to meet the requirement of users.

In the following, we will define the danger coefficient to quantify the privacy violations of permissions to mobile users and also address the above-mentioned three issues to recommend mobile apps from usage perspective.

3.2. Danger Coefficient. Generally speaking, the privacy information is normally leaked when people launch an app with privacy permission. We introduce here a new parameter, called *danger coefficient* (DC), capable of expressing the leakage of privacy when users run apps on their mobile device.

To evaluate the DC of each user, two factors need to be determined. The first one is the privacy risk of each app, which is reflected by the permissions that the app asks from users. We measure the privacy risk of permissions requested by the app and consider it as one factor for evaluating the app's DC. The second one is the app usage pattern by the user. In what pattern the mobile app is used indicates the probability that the privacy information disclosed by the app will be leaked, which is considered as the other factor for evaluating app's DC.

We now address the problem by characterizing and quantify the privacy risk of permissions and then the privacy risk of an app. Permissions can be classified into three classes, which are normal permissions, critical permissions, and system permissions as we discussed in the previous section. Requesting a more critical permission increases risk more than requesting a less critical one. For the quantification, we leverage the probabilistic approach proposed in [10] to evaluate the risk of each of the categories of permissions.

For the generic app a_i ($i = 1, \ldots, M$), the permissions that will be accessed are denoted by the set $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,N}\}$, where N is the total number of permissions. The generic variable $p_{i,j}$ is binary and assumes value 0 or 1 if permission j is not or is present in app a_i. For mobile apps, different types of permissions may correlate with each other. For instance, permissions related to network (including Internet access, checking WiFi state, checking network status, changing WiFi status, and changing network connection) are mutually correlated. However, such dependence introduces sophisticated analysis to conduct privacy risk evaluation. In contrast, the study [10] discovers that assuming the independence of different permissions can still perform well for the overall privacy risk evaluation but it is more simplistic for the analysis compared with dependence situations. Furthermore, the assumption of independence of different permissions allows a monotonic model, which allows the consideration of each individual permission. Besides, it can also help to differentiate different classes of permissions.

Therefore, if we use $f(P_i)$ to indicate the privacy risk factor for app i, P_i is generated by N independent Bernoulli random variables and is given by

$$f(P_i) = \prod_{j=1}^{N} r_j^{p_{i,j}} \left(1 - r_j\right)^{1-p_{i,j}}, \quad (1)$$

where r_j is the probability that permission j ($j = 1, \ldots, N$) is accessed by an app.

Following [10], r_j is obtained using a Beta($r_i \mid a_0, b_0$) function. That is,

$$r_j = \frac{\sum_{i=1}^{M} p_{i,j} + a_0}{M + a_0 + b_0}, \quad (2)$$

where M is the total number of apps used for evaluation. In this paper, the value of M is set to 900 as the dataset contains 900 mobile apps. To suggest the several privacy violation risks, we set $a_0 = 1$, $b_0 = 2M$ with less penalty effect for critical permissions. For normal permissions, we set $a_0 = 1$, $b_0 = M1$, which is normal distribution suggesting the less effect of the privacy risk, $a_0 = 1$, $b_0 = M$ with less penalty effect for critical permissions, for normal permissions the value $a_0 = 1$, $b_0 = 1$, which is normal distribution suggesting the less effect of the privacy risk. With this method, different types of permission privacy risks can be distinguished.

Furthermore, apps may request both function-related and function-unrelated permissions. For example, an app providing map service needs location information as its function-related permission, whereas if an app serves as chatting service, requesting calling permission will very likely be considered as function-unrelated permission. If an app requests a function-related permission, the privacy violation is considered much weaker than that of function-unrelated permission request. To show the difference, we assign different weights for the privacy risk and define the weight factor $\omega_{i,j}$ of permission j for app a_i. The weights of the function-related permissions should be less than that of the function-unrelated permissions, as the function-related permissions are about to enable functions and services, whereas the function-unrelated permissions intend to collect user privacy information. For instance, we take an empirical value that the weight for function-related permissions is 0.5 and for function-unrelated permissions is 1. Then the overall risk factor $f(a_i)$ of app a_i that takes into account the weight of the different risks is given by

$$f(a_i) = f(P_i) * \prod_{j=1}^{N} \left(\omega_{i,j}\right). \quad (3)$$

Considering the app privacy risk is monotonically decreasing with respect to the probability of using granted permissions, which means removing a permission always reduces the risk value of an app, for privacy risk calculation of an app, we use the following function [10]:

$$R(a_i) = -\ln\left[f(a_i)\right]. \quad (4)$$

As far as the app usage pattern is concerned, a user with longer time duration of using an app will have more chance to access privacy information. Although there are some mobile apps that are launched in the background and steal the user privacy information without being invoked by users, they are not measured and counted in our analysis. In this paper, we assume that the probability that a mobile app accesses privacy information is proportional to the time duration in which the app is used. Therefore, we measure the usage pattern by

expressing the fraction of time user m ($m = 1, \ldots, L$) in which uses app a_i; that is, $U_{i,m} = t_{i,m}/T$, where $t_{i,m}$ is the total time usage of app a_i by user m and T is the total observation interval of the system.

The state of the art suggests that interevent times of human behaviors follow Poisson distribution [13]. In this paper, we assume the interusage time follows a Poisson distribution with the parameter equal to $\lambda_{i,m}$ for user m running app a_i. Then $U_{i,m} = 1 - 1/\lambda_{i,m}$.

The DC measures the danger degree of a user using a mobile device. For a user u_m, the danger coefficient can be derived by combining the app privacy risk and the app usage pattern and is expressed as

$$\mathrm{DC}(u_m) = \sum_{i=1}^{M} [R(a_i) * U_{i,m}], \tag{5}$$

where M is the number of apps launched in the device. The larger $\mathrm{DC}(u_m)$ indicates the more chance for privacy violation when user u_m launches apps in the mobile device. For a user with multiple devices, each device can be evaluated by the above procedure for the danger coefficient estimation. We use DC as a metric to evaluate the danger degree of the recommendation algorithms in Section 4.

3.3. User-App Preference. The user-app preference reflects the preference relationship between individuals and apps. As each individual user may not display enough information to fully discover his or her individual preference, we employ an individual user-app preference based on the distribution of common preference, as presented in [3]. Specifically, we firstly investigate the common preference of many users and then represent each user's preference by a distribution of common preference. If the common preferences are presented by z, the conditional probability that a user u_m prefers the category a given the set of all apps A can be represented as

$$P(a \mid A, u_m) = \frac{P(a, A \mid u_m) * P(u_m)}{P(A, u_m)}. \tag{6}$$

For given apps A and user u_m, $P(u_m)$ and $P(A, u_m)$ are constant. Therefore, we have

$$\begin{aligned} P(a \mid A, u_m) &\sim P(a, A \mid u_m) \sim \sum_z P(a, A, z \mid u_m) \\ &\sim \sum_z P(a, A \mid z) P(z \mid u_m), \end{aligned} \tag{7}$$

where the preference of user u_m to the category of apps a is determined by the common preferences of many users (i.e., $P(a, A \mid z)$) and also the user's personal preference conveyed by a distribution of common preferences (i.e., $P(z \mid u_m)$). The calculation of common preferences of other users and the distribution of common preferences of user u_m can be presented by the normalized number of apps launched by user u_m and other users.

The above user-app preference suggests the preference of a user to a category of mobile apps. To identify categories of mobile apps, we first find the categories of each mobile app with its assigned categories. However, such classification is not fine-grained enough for app usage preference recommendation. For example, many mobile apps are associated with offline services, such as online banking apps or social media apps. A Facebook social app cannot be replaced by a Twitter social app. Therefore, for each category of mobile apps, we use a topic model to categorize apps in different coarse grained preference categories into different groups according to their functions and usage context. For this purpose, we employ Latent Dirichlet Allocation (LDA) model [14], in which an app is associated with a word in a document, and each category is a topic. With such topic model, apps are put into different fine-grained preference groups. Specifically, for each user, we extract the context components (i.e., time stamp, and location) from the usage record and consider the set of context components of a user as this user's bag of context components. For the fine-grained preference of each user, the procedure is conducted as follows. It begins with a random assignment of fine-grained preference to each context component. It then iteratively estimates the conditional probability of assigning of the preference to context component and updates the preference of each context component according to the latest calculation. The assignment will converge finally, which means each context component is assigned with a fine-grained preference. Then the user fine-grained preference will be determined by her context component bag. But LDA has some drawbacks to identify the sequence of words and also it is in the topics composition in which the same words appear in the multiple topics. In our case, these drawbacks do not affect the performance of mobile app group identification. For the fine-grained preference classification, only the special words associated with the fine-grained group are used. Moreover, the classification does not involve the sequence of words. Therefore, LDA model is capable of fine-grained classification for mobile apps. If a person launches an app on the mobile device, it is indicated that the person is interested in the functionality provided by the app. If the app is considered with high risk, then another app in the same category should be recommended.

3.4. App Usage Recommendation. As the high risky app may cause more privacy leakage, recommending the user preferred apps with less privacy risk is more desirable. In this section, we will propose a method for app usage recommendation combining required function, user preference, and privacy, named AppURank.

Regarding required functions, we recommend apps which are in the same category as the launched apps to guarantee the function similarity. Then we recommend apps according to preference and privacy. The objective of the app usage recommendation is finding a group of apps $\mathscr{A}$ consisting of a collection of k apps with the corresponding weight α_i and $\sum_{i=1}^{k} \alpha_i = 1$ which display minimal privacy risk and meanwhile satisfy the individual preference and function requirement. The general formulation of the objective can be presented as

$$\max_{\mathscr{A}} \sum_{i=1}^{k} \alpha_i * [P(a \mid A, u_m)] - b * \sum_{i=1}^{k} \alpha_i * R(a_i), \tag{8}$$

where $\alpha_i \in a$ suggests α_i is one mobile app in category a and the value of b depends on the privacy and individual preference requirement of users. For example, if a user considers privacy more important, then the value of b will be larger, while if a user takes user preference more into account, then the value of b will be smaller. Given the vector $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_k]$, the corresponding preference vector and risk vector are represented by P and R. Equation (8) can be rewritten as

$$\max_{\mathscr{A}} \alpha^\top P - b * \alpha^\top R. \quad (9)$$

In this case, P and R are independent, and the selection of α is to rank mobile apps according to user preference and privacy risk according to the calculation of $R(a_i)$ and $P(a \mid A, u_m)$ and then combine the two aspects, as specified by (9). The app with the highest combination value is considered as the most recommended app. Apps with top k values are the top k apps on the recommendation list.

4. Performance Results

In this section, we describe the experimental data and evaluate the privacy risk of mobile apps, danger coefficient of mobile users, and AppURank recommendation method.

4.1. Experimental Data. In this paper, we need two data traces to evaluate the danger coefficient and the proposed recommendation approach.

One data trace contains apps with their requested permissions, and the permissions are marked with function-related and function-unrelated. There are 900 apps and their permissions in the data trace. For each app, we collect the permission information from the privacy description in the store and use a matrix to store them. The rows are apps, and columns are permissions. If an app requests a permission, the element in the matrix will be 1; otherwise, it will be 0. Then we take the function relativity into consideration. We investigate the function relativity in two stages. In the first stage, we crawl the coarse grained categories of mobile apps and identify the permission by their main functions. For example, a location-based social service should need the location information, which means location information is function-related for the location-based social service. In the second stage, we manually identify the other permissions accessed by each of the mobile apps. For instance, an alarm app could require access to microphone to let the user record some memo/voice to be played during timeout. If the permission is requested by malicious functions of an app, it is considered as function-unrelated. For a function-related permission, the value of element is changed to 0.5. For function-unrelated permissions, the value of element is still 1. Then the privacy risk of each app in this data trace can be calculated using the expression of $R(a_i)$ in (4). Figure 1 shows the distribution of apps in terms of the number of requested permissions. It shows that about 20% of apps request less than 10 permissions. The percentage of apps requesting 10 to 20 permissions reaches over 75%, and about 5% of apps request over 20 permissions. On average, each app requests 12 permissions. Figure 2 shows the distribution of apps in terms of different types of permissions. More than 70% of apps request Internet access, phone state, and network state. There are several apps requests for external device format, which highly violate the user privacy and even refer to the security of mobile devices. Regarding function relativity of apps, we draw Figure 3 to show the average number of function-related and function-unrelated permissions of different categories of apps. It shows that function-unrelated permissions are more than function-related permissions in most cases. In the apps of security, car/cab, social, voice, reader, helper, and tools, the function-unrelated permissions are requested twice as much as function-related permissions.

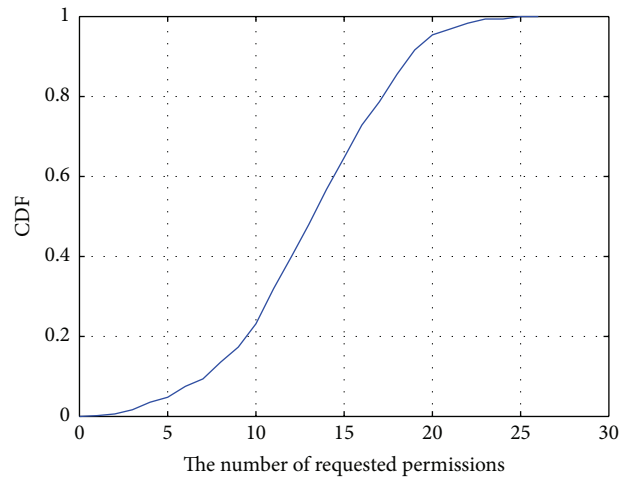


Figure 1: CDF of permission distribution from apps perspective.

Table 2: Characteristics of the dataset.

Characteristics	Value
Users	13,969
Duration	545 days
Network types	2G/3G/4G Wi-Fi
Applications	16,878
Total traffic	13.8 TB
Number of records	Over 103 million

The other data trace is the mobile users and their mobile app usage pattern. In order to obtain such data, we design a mobile app named AntTest (http://www.wandoujia.com/apps/edu.bupt.anttest) on Android platform. Indeed, the AntTest application is developed to measure network speed of mobile device. For such purpose, the data such as app usage pattern of each user is recorded every 5 seconds, which can be exactly applied in this study. We put the AntTest in the app store to provide a way for users to download. So far, the app has been available for more than 600 days since March 2014. The total number of records is over 103 million produced by 13,969 users. The detailed data description is presented in Table 2. The format of one record is presented as $\langle RecordID, IMEI, AppID, Time \rangle$, where *RecordID* is the ID of the record, *IMEI* indicates the user equipment ID, *AppID* is identified by the name and package of the app, and *Time* indicates the time of the record. We anonymize the dataset to conduct the experiment for the privacy concern.

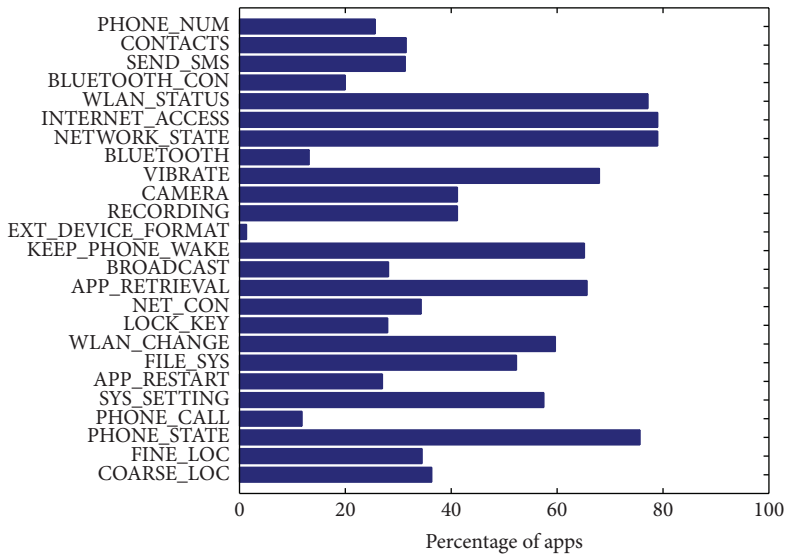


Figure 2: The permission distribution from apps perspective.

4.2. Evaluation of App Usage. In order to evaluate the privacy risk when people launch mobile apps, we measure the app privacy risk, app usage pattern, and DC of users when they launch apps from their mobile devices.

For the given 900 mobile apps in the permission dataset, we measure the privacy risk of each app, which is one factor of DC, combining the requested permissions and function relativity. The average privacy risks (by (4)) of different mobile app categories are presented in Figure 4. It shows that all the categories are with privacy risk more than 15. According to the statistics shown in Figure 1, one mobile app can access as many as 25 permissions in the dataset, indicating the upper bound of the privacy risk factor $R(a_i)$ is about 60. According to our observation, apps are risky if their privacy risks are over 20 (1/3 of the upper bound). From the figure, the privacy risk expectation of apps for video, audio, web browser, shopping, voice, tools, and security is high. In contrast, apps for news, live streaming, photography, reading, and radio are relatively secure.

To show the impact of function relativity, we conduct the evaluation with varied ω_{ij}, which is weight for function-related and function-unrelated permissions. We fix ω_{ij} to be equal to 1 for function-unrelated permissions and vary ω_{ij} for function-related permissions with 0.2, 0.5, and 0.8 (as the legend) in terms of five-category apps (WiFi, streaming, social, video, and mail). The result as shown in Figure 5 presents the average privacy risk of the five categories of mobile apps. It shows that the trend of privacy risk does not change with different values. Moreover, the higher the value of ω_{ij} chosen for function-related permissions selected, the higher the privacy risk the mobile apps own. Among all the rest of experimental results, we use $\omega_{ij} = 0.5$ for function-related permissions and use $\omega_{ij} = 1$ for the function-unrelated permissions to reflect the different privacy violation concern with respect to function relativity of mobile apps.

The app usage pattern, which is the other factor of DC, measures the time usage of different apps. Figure 6 shows the usage time distribution of all apps with respect to mobile apps usage (Figure 6(a)) and users usage time (Figure 6(b)). They record 900 mobile apps and around 2,000 users who run these 900 apps in the datasets. Both of them follow the heavy tail distribution (see the straight line). Majority of apps have short usage time, while few apps have a long usage time. Similarly, most users use their mobile devices for short time, while few users have long time duration for mobile device usage. To show the usage pattern of specific mobile apps, we select several typical apps (i.e., WeChat, QQ, and Tencent video) and show their usage patterns in Figure 7. The plot shows that over 90% of users run WeChat for less than 10^4 s, while this number of users decreases to 75% for Tencent video. This is due to the different attributes of the apps functionalities, where WeChat is used for messaging, whereas Tencent video is used for video playing. The usage time duration for video playing apps (i.e., Tencent video) should be longer than that for messaging apps (i.e., WeChat and QQ).

We measure the DC of mobile devices according to user-app usage pattern and the risk of mobile apps. Specifically, we evaluate the DC of 50 sampled users given the collected information of 900 mobile apps and draw the DC value of each person. In our experiment, the DC value is bounded by $[0, 60)$. For the convenience of illustration, we ordered the DC values as shown by the line with forward triangles in Figure 8. It shows that all users are with DC value larger than 20, and the highest one reaches almost 30. From our observation, if DC value is over 20, it indicates that the mobile phone usage becomes risky. Our selected 50 users are all in the risky status.

4.3. Evaluation of App Usage Recommendation. To evaluate the performance of the proposed recommendation approach, we calculate DC values of people under different conditions to quantify how much danger of mobile apps with respect to privacy violation to mobile users. Specifically, we vary the parameter b from 0 to 1 and finally to 100, to see the DC turbulence for different users. We compare the proposed

Table 3: The requested permissions of news apps.

Permissions	Baidu	Phoenix	Sohu	Tencent	NetEase	CCTV	Sina	Online retail	The paper
Location		×	×	×	×				
SMS sending				×					
Contacts	×			×					
Network connection	×	×	×	×	×	×	×	×	×
Recording		×	×	×	×				
Camera	×				×				×
System setting	×		×	×	×	×	×		×

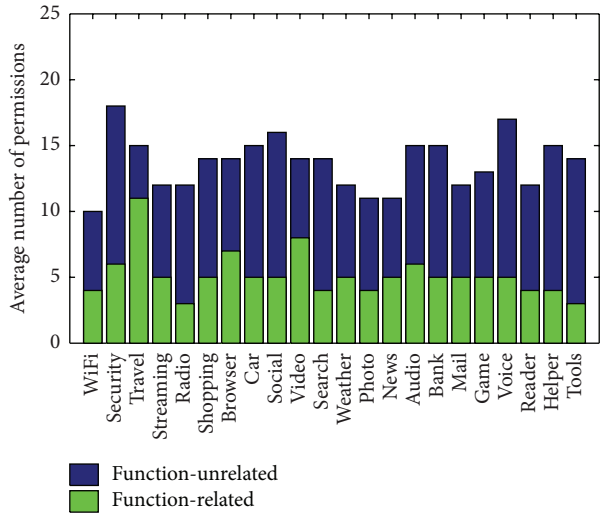


Figure 3: The permission distribution of different categories of apps.

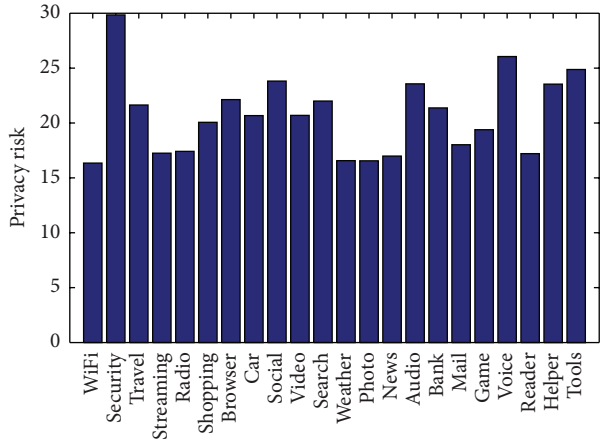


Figure 4: The risk distribution of different categories of apps.

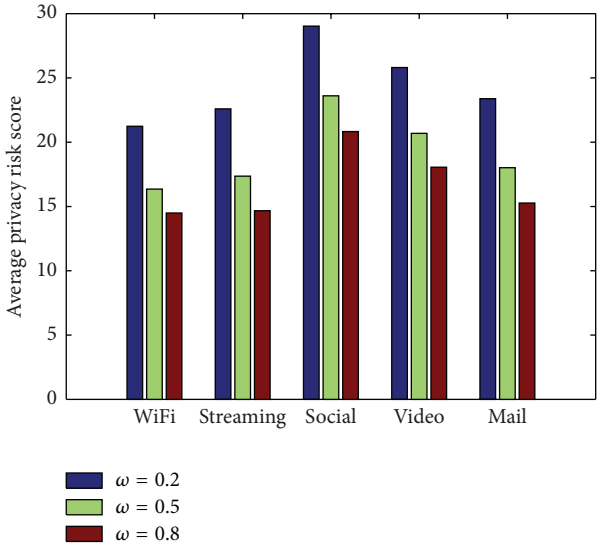


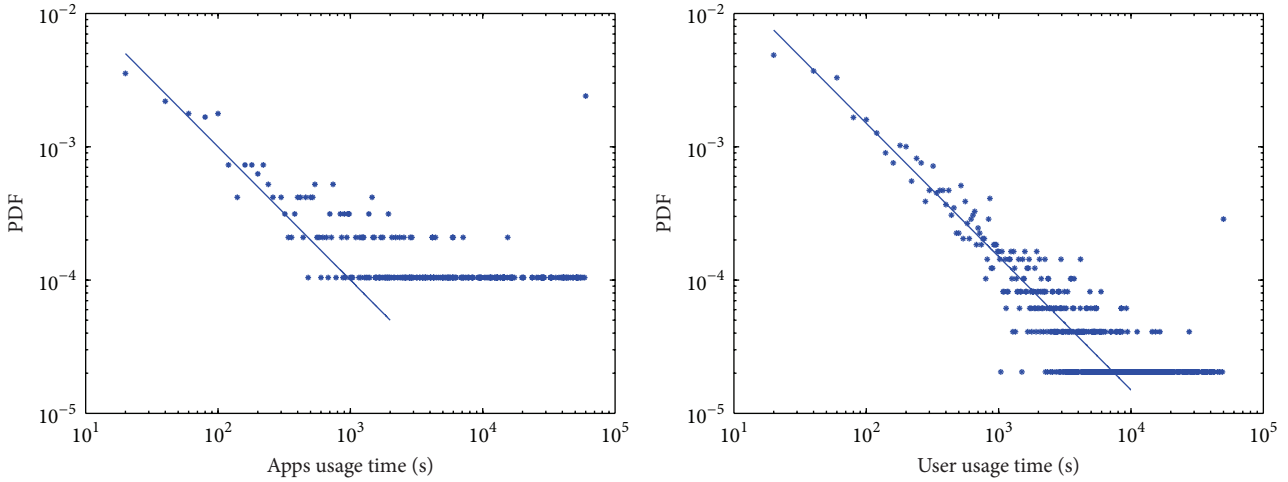
Figure 5: The impact of function relativity as function of ω_{ij}.

approach with the PNB method [10], which only considers the app risk for recommendation. Indeed, PNB provides the baseline for the evaluation.

The result of the evaluation is shown in Figure 8. Besides the ground truth observed in the real world, it presents the DC value of the proposed recommendation approach in case of $b = 0$, $b = 1$, and $b = 100$ and PNB method. When $b = 0$, the recommendation is led by the personalized user-app preference, as illustrated by the line with plus marks. It shows that the danger coefficient is much higher than ground truth if the privacy risk issue is neglected. Meanwhile, it also shows that some users' preference can reach a lower danger coefficient. This is due to the intrinsic low risk of the mobile apps.

Furthermore, we consider the situation in which both user-app preference and privacy have the same importance, which is considered setting $b = 1$, and the line with backward triangles is obtained. The DC value is much lower than the ground truth. Only two users are still with DC value higher than 20. We further increase the value of b to 100 to show the DC values when the risk takes the dominant role. The result is shown as the line with solid circles. It reaches almost the baseline obtained from PNB marked by the line with circles. Without loss of generality, we consider that the majority of people would like to consider preference and privacy equally important. In such case ($b = 1$), the DC values reduce to about 50% on average compared with ground truth.

We further investigate the recommendation results for news app categories. We first show several common permissions of the news apps in Table 3. In all 7 listed permissions, only network connection is the function-related permission



(a) App usage time (b) User usage time

Figure 6: The PDF of usage time in terms of apps and users.

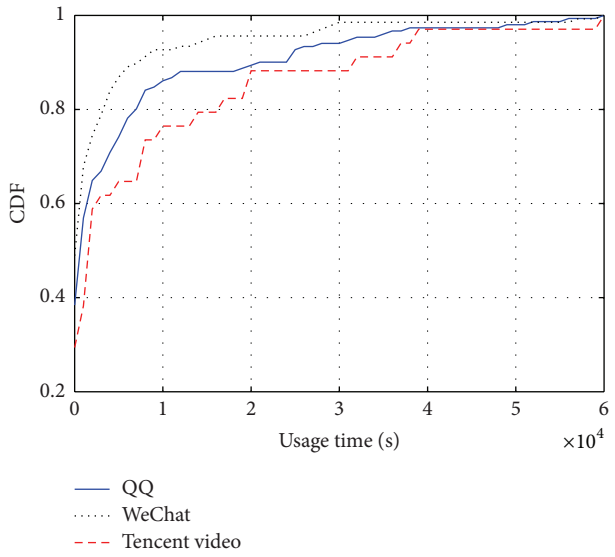


Figure 7: The usage pattern of three mobile apps.

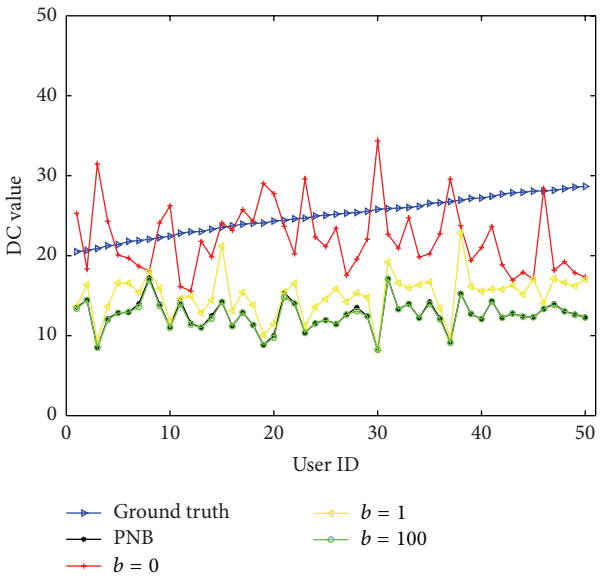


Figure 8: The DC of random selected 50 users.

to news reading. The remaining 6 permissions such as location, SMS, contacts, recording, camera, and system settings are function-unrelated to these apps. For each app, we mark a cross if it asks for a certain permission in the corresponding place. It shows that the app online retail requests only one function-related permission. CCTV and Sina request one function-unrelated permission system setting besides network connection. Baidu, Sohu, NetEase, Phoenix, and the paper request even more function-unrelated permissions. The most severe one is Tencent, which requests nearly all the listed permissions. We show the recommendation list of news apps as a function of different parameters shown in Table 4. When $b = 0$, Baidu News is the first app recommended, indicating it fits user preference for news reading. The paper news is the last one as it is not as popular as the others. When $b = 1$, Baidu News is still in the first place. If people consider privacy and preference equally important, Baidu News is the best choice. However, Tencent News goes to the last position. This is mainly due to its violation of user privacy. If the privacy is considered as the most important factor (when $b = 100$ and PNB), the online retail news app becomes the first recommended app, due to its lowest privacy violation to users, as we have shown in Table 3. In contrast, Baidu News falls into the fourth position. Tencent News still holds the last position, given its high violation to user privacy.

5. Conclusions

In this paper, we proposed a method to evaluate the privacy risk from mobile apps when people use a mobile device. We evaluated the mobile app privacy risk and defined a

Table 4: Recommendation list of news apps.

$b = 0$	Baidu	Phoenix	Sohu	Tencent	NetEase	CCTV	Sina	Online retail	The paper
$b = 1$	Baidu	CCTV	Online retail	Sina	Phoenix	Sohu	NetEase	The paper	Tencent
$b = 100$	Online retail	CCTV	Sina	Baidu	NetEase	Sohu	Phoenix	The paper	Tencent
PNB	Online retail	CCTV	Sina	Baidu	NetEase	Sohu	Phoenix	The paper	Tencent

danger coefficient for each user by combining the mobile apps risk and user preference. According to the requirement of privacy and satisfaction of app usage preference, we proposed a mobile app recommendation method named AppURank. The evaluation results showed that the privacy risks of apps are different, and the DC of mobile users is very high for many of them. The proposed recommendation method can help to reduce the danger coefficient by 50% on average and meanwhile maintains personalized user preference. For the future work, we will build the mobile app recommendation system on the mobile device and evaluate the performance of the recommendation algorithm by implementation and deployment.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work has been partially sponsored by National Science Foundation of China (no. 61502045), EU FP7 IRSES Mobile Cloud Project (Grant no. 612212), the 111 Project (no. B08004), the Fundamental Research Funds for the Central Universities, and the Beijing Higher Education Young Elite Teacher Project.

References

[1] B. Yan and G. Chen, "AppJoy: personalized mobile application discovery," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pp. 113–126, ACM, Washington, DC, USA, July 2011.

[2] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian, "Towards personalized context-aware recommendation by mining context logs through topic models," in *Advances in Knowledge Discovery and Data Mining: 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part I*, vol. 7301 of *Lecture Notes in Computer Science*, pp. 431–443, Springer, Berlin, Germany, 2012.

[3] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, "Mining mobile user preferences for personalized context-aware recommendation," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 4, article 58, 2015.

[4] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Mobile app recommendations with security and privacy awareness," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*, pp. 951–960, ACM, New York, NY, USA, August 2014.

[5] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS '11)*, pp. 627–638, ACM, Chicago, Ill, USA, October 2011.

[6] K. W. Y. Au, Y. F. Zhou, Z. Huang, P. Gill, and D. Lie, "Short paper: a look at smartphone permission models," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '11)*, pp. 63–67, ACM, October 2011.

[7] W. Enck, P. Gilbert, B.-G. Chun et al., "An information-flow tracking system for realtime privacy monitoring on smartphones," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI '10)*, pp. 1–6, USENIX Association, Berkeley, Calif, USA, 2010.

[8] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, "Taming information-stealing smartphone applications (on android)," in *Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST '11)*, pp. 93–107, Springer, Pittsburgh, Pa, USA, June 2011.

[9] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 235–245, ACM, Chicago, Ill, USA, November 2009.

[10] H. Peng, C. Gates, B. Sarma et al., "Using probabilistic generative models for ranking risks of Android apps," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 241–252, October 2012.

[11] J. Lin, B. Liu, N. Sadeh, and J. I. Hong, "Modeling users' mobile app privacy preferences: restoring usability in a sea of permission settings," in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS '14)*, pp. 199–212, USENIX Association, Menlo Park, Calif, USA, July 2014.

[12] R. Liu, J. Cao, L. Yang, and K. Zhang, "PriWe: recommendation for privacy settings of mobile apps based on crowdsourced users' expectations," in *Proceedings of the IEEE International Conference on Mobile Services (MS '15)*, pp. 150–157, New York City, NY, USA, June 2015.

[13] S. K. Katti and A. V. Rao, "Handbook of the poisson distribution," *Technometrics*, vol. 10, no. 2, pp. 412–412, 1968.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 1763416, 9 pages
http://dx.doi.org/10.1155/2016/1763416

Research Article

A Perturbed Compressed Sensing Protocol for Crowd Sensing

Zijian Zhang, Chengcheng Jin, Meng Li, and Liehuang Zhu

Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Liehuang Zhu; liehuangz@bit.edu.cn

Received 10 December 2015; Revised 28 April 2016; Accepted 10 May 2016

Academic Editor: Tony T. Luo

Crowd sensing network is a data-centric network consisting of many participants uploading environmental data by smart mobile devices or predeployed sensors; however, concerns about communication complexity and data confidentiality arise in real application. Recently, Compressed Sensing (CS) is a booming theory which employs nonadaptive linear projections to reduce data quantity and then reconstructs the original signal. Unfortunately, privacy issues induced by untrusted network still remain to be unsettled practically. In this paper, we consider crowd sensing using CS in wireless sensor network (WSN) as the application scenario and propose a data collection protocol called perturbed compressed sensing protocol (PCSP) to preserve data confidentiality as well as its practicality. At first, we briefly introduce the CS theory and three factors correlated with reconstruction effect. Secondly, a secure CS-based framework using a secret disturbance is developed to protect raw data in WSN, in which each node collects, encrypts, measures, and transmits the sampled data in our protocol. Formally, we prove that our protocol is CPA-secure on the basis of a theorem. Finally, evaluation on real and simulative datasets shows that our protocol could not only achieve higher efficiency than related algorithms but also protect signal's confidentiality.

1. Introduction

Crowd sensing network is a powerful sensor network utilizing the force from crowd. Crowd sensing is a form of network wireless sensing, which can be achieved by exploiting WSN. With enormous sensors deployed, WSN is limited by its relatively weak computational capability and low energy reservation. The primary task of WSN is to sense, transmit, and process packets while maintaining the energy cost to the minimum.

In traditional WSN, where communication is conducted via intranet or private network, bandwidth is severely consumed and certain commands from sensor nodes cannot be timely relayed to information server because great amounts of data collected during collection phase need to be transmitted. On the other hand, since trust management is maintained in public network, data confidentiality may be exposed. Hence, how to reasonably design secure transmission schemes in WSN has become a precondition for applying WSN to many fields extensively.

Without the traditional signal acquisition process constraint, Compressed Sensing (CS), proposed by Candes et al. [1] and Donoho [2] in 2006, is a booming theory that captures and represents compressible signals at a sampling rate significantly lower than the Nyquist rate [3–6]. It first employs nonadaptive linear projections that preserve the structure of the signal, and then the signal reconstruction can be conducted using an optimization process from these projections. Compressive sensing has a wide range of applications such as compressive detection and estimation, DNA microarray, and distributed compressed video sending [7].

Moreover, traditional data compressing method of WSN comes with several disadvantages, including the following. (1) Several important components and corresponding locations need to be preserved after orthogonal transformation in data compressing; otherwise, the original data could not be recovered [7]. (2) In layered multihop WSN, owing to the hardware limitation, sensors' energy storage is constrained to a low level. Intuitively, nodes closer to sink node will die sooner thanks to their faster battery consumption rate, which would result in the imbalance of energy consumption among sensors in different positions. Due to the advantages of CS, more and more CS techniques have been integrated into WSN, but most of them only consider the time relativity

of a single node. In fact, space relativity can also be traced in nodes of WSN, leading to Distributed Compressed Sensing (DCS) which views the raw data as original signal and compress the signal before transmitting. DCS has advantages as follows. (1) The random measurement from DCS is a random linear combination of every element in original signal. Thus, losing part of measurement will not affect the reconstruction of original signal. (2) In DCS-based WSN model, data quantity of each node remains the same, so energy consumption is balanced and network lifetime is prolonged.

Although DCS can effectively solve the problems raised by traditional methods, data security can never be overlooked. Researches on CS security still need to be explored. Some [8–11] tried to modify the measurement matrix but failed to apply their schemes in WSN; others [12] performed encryption (like AES, etc.) after the data is compressed to protect data security, but secure network is required. Notice that most WSN is deployed in remote, unattended, or even hostile environment, meaning node's reliability is difficult to guarantee. Therefore, it is crucial to design a secure model. In this paper, we propose a perturbed compressed sensing protocol (PCSP) to preserve data confidentiality with high practicality. Our contributions are listed as follows.

(i) We propose a perturbed compressed sensing protocol (PCSP) in WSN for crowding sensing and our PCSP can reduce communication complexity explicitly.

(ii) We prove that our PCSP can provide data confidentiality; to be more specific, our PCSP is proved to be chosen-plaintext attack secure.

(iii) We systematically evaluate our PCSP by comparing its performance with existing approaches. Experiments show that our PCSP achieves higher accuracy of recovery.

Organization. The rest of this paper is organized as follows. In Section 2, we review the related work presented in the literature. Then, we briefly introduce the main idea of CS in Section 3. Section 4 illustrates our protocol in detail. While security is discussed in Section 5. We systematically evaluate performance of PCSP by making comparisons with existing approaches in Section 6; in addition, limitations of our protocol and future work are explained in Section 7. At last, we conclude this paper in Section 8.

2. Related Work

Compressed Sensing (CS) is a new method for compressing signal which breaks through the traditional limit of sampling frequency. Through matrix computation at the encoding end, we can compress the original signal from high dimension to low dimension with a small sampling frequency and low computation complexity. At the decoding end, the original signal is reconstructed by solving a convex optimization problem.

Meanwhile, CS is capable of providing a good encryption feature on its interior structure level. Because the projection is a function value of measurement matrix which can be seen as a shared key between encoding end and decoding end.

Researches on CS put focus upon three factors associated with the reconstruction effect: sparse representation, measurement matrix, and reconstruction algorithm improvement. As a precondition for applying CS, common methods for sparse representation are discrete cosine transform basis, fast Fourier transform basis, disperse wavelet transform basis, Curvelet basis, Gabor basis, and redundant dictionary [15]. In particular, redundant dictionary or overcomplete dictionary can adaptively find out the optimal base according to the sparse property of different signal such that the minimum sparsity on this base and the best signal compression degree are both reached. For measurement matrix, Null Space Property (NSP) [16] and Restricted Isometry Property (RIP) [1, 17–19] should be satisfied; these matrixes include Gauss random matrix, Bernoulli measurement matrix, sparse stochastic matrix, toeplitz matrix, and circulant matrix. The work in [1, 2, 15, 20] proved that measurement matrix making up of independent and identical distributed Gauss random variable is irrelevant with any overcomplete redundant dictionaries, and accurate recovery of original signal can be guaranteed even after the signal is compressed. Hence, Gauss random matrix is one of the best options for measurement matrix, but doing so brings high complexity and pseudorandom matrix is an alternative choice in researches. In recent years, researchers have been working on robust pursuit algorithm, such as greedy pursuit (including MP [21], OMP [22], StOMP [23], and ROMP [24]), convex relaxed approach (including BP [25], interior point method [26], gradient projection method [27], and iterative threshold method [28]), and the combination of the former two (including Fourier sampling [29] and HHS [30]).

The classic OMP [22] is a greedy pursuit, the basic idea is transvection computation, and the most related (to compressed value Y) column vector is selected in each iteration, until the reconstruction sparse representation of original signal is found. Then we can retrieve original signal through spares inverse operation and decryption. Its advantage is convenient implementation, whereas the disadvantage is that multiple measurements are required.

As long as CS is proposed, how to use CS to provide data security is also a research hotspot. The work in [30–33] pointed out that the linear projection on measurement matrix is essentially a protection of data secrecy to some extent. The work in [30] analyzed the security of CS under several possible attacks. The work in [31] compared CS with other encryption methods through quantization. The work in [32, 33] designed the measurement matrix as symmetric secret keys such that eavesdroppers cannot obtain original signal. The work in [12] adopted AES and SHA to provide data confidentiality and data integrity after data compression.

Regarding the security problem raised by applying CS to WSN, this paper proposes an encryption method based on existing DCS model. Analysis and experiments show that our approach can provide data confidentiality with high accuracy.

3. Preliminary

First, let us take a review at the basic principles of CS. CS theory suggests that N-dimension original signal X can be

linearly projected into $M \times 1$ matrix by $M \times N$ measurement matrix Φ. If using some orthogonal basis or atomic set Ψ, such as Gabor basis and redundant dictionary [15], which is used in our frame, X can be interpreted as a vector $\theta \in R^N$ with only k nonzero elements which means

$$X = \Psi\theta = \sum_{i=1}^{N} \psi_i \theta_i. \tag{1}$$

We call X k-sparse and the solution to equation above sparse representation or sparse decomposition. To further explain (1), we have

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{1N} \\ \psi_{21} & \psi_{22} & \cdots & \psi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{N1} & \psi_{N2} & \cdots & \psi_{NN} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix} \tag{2}$$

and (2) can be inferred by substituting $x_i = \psi_{i1}\theta_1 + \psi_{i2}\theta_2 + \cdots + \psi_{iN}\theta_N$, so we have

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \psi_{11} \\ \psi_{21} \\ \vdots \\ \psi_{N1} \end{bmatrix} \theta_1 + \begin{bmatrix} \psi_{12} \\ \psi_{21} \\ \vdots \\ \psi_{N2} \end{bmatrix} \theta_2 + \cdots + \begin{bmatrix} \psi_{1N} \\ \psi_{2N} \\ \vdots \\ \psi_{NN} \end{bmatrix} \theta_N. \tag{3}$$

Then X can be projected on $M \times N$ measurement matric Φ to obtain $M \times 1$ vector:

$$Y = \Phi X = \Phi\Psi\theta = AX, \tag{4}$$

where $A = \Phi\Psi$ is the sensing matrix. Meanwhile, the measurement matrix requires satisfying NSP and RIP. In [18, 34], Gauss random matrix is proved to be appropriate, so it is used in our protocol to measure signal. Then the M-dimension projection is transmitted to receiver for recovering original signal. As introduced in Section 2, in CS field, OMP algorithm is a classical recovery algorithm, which can obtain the sparsity coefficient of data. Therefore, our recovery algorithm is based on OMP algorithm. To further study it, OMP algorithm is described in Algorithm 1.

Input: compressed signal Y, sensing matrix A;
and signal sparsity k:
begin
(1) $\text{res}_0 = Y$, $\Lambda_0 = \emptyset$, $A_0 = \emptyset$, $t = 1$;
(2) **while** $t \le k$ **do**
$\lambda_t = \arg\max |\langle r_{t-1}, a_j \rangle|$, $j \in [1, 2, \ldots, N]$;
$\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$, $A_t = A_{t-1} \cup a_\lambda$;
$Y = A_t\theta_t$, compute the least square solution:
$\widehat{\theta}_t = \arg\min \|Y - At\theta t\| = (A_t^T A_t)^{-1} A_t^T Y$;
$\text{res} = Y - A_t\widehat{\theta}_t = Y - (A_t^T A_t)^{-1} A_t^T Y$;
$t = t + 1$;
(3) **Output** $\widehat{\theta}$;
end

Algorithm 1: Orthogonal matching pursuit (OMP).

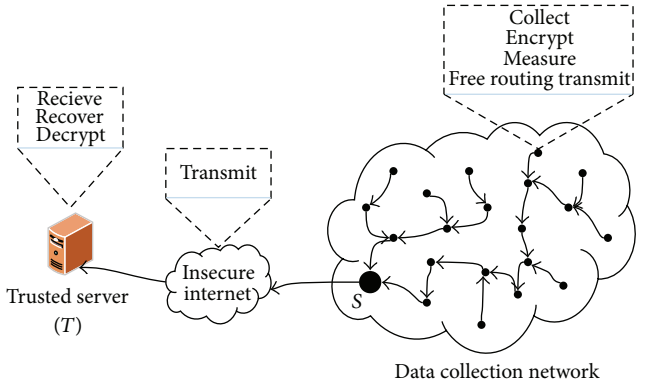


Figure 1: Overview of WSN.

4. Perturbed Compressed Sensing Protocol (PCSP)

4.1. Network Assumption. For simplicity, we denote smart device and sensor as node in the rest of the paper. Also we assume a general multihop network with n nodes and N alive, a sink node S, and a trusted server T. The overview of WSN is shown in Figure 1. Each node is required to register with the trusted registration authority RA to share a secret key with T. Nodes can collect environmental information such as temperature, humidity, and pressure. They can also receive node information from last hop node and forward node information to the next hop node. T can compute each node's corresponding measurement coefficient matrix as sensing matrix for reconstruction.

Each alive node generates a packet. As packets travel towards S, our protocol allows each node to choose the nearest node whose distance to S is smaller as the next hop node to forward the packet. The category of collected information is distinguished by the network layer data packets. The format of a packet (8 bytes) is shown in Figure 2. Where ID is the ID number of current node. Flag represents the category of collected information by nodes (1 is temperature, while 2 indicates humidity, and 3 represents pressure; also, we use 4 and 5 to denote light and salt, resp.). The value of collected environmental information can be read from Value. For ID List, it is NULL if the node is a leaf node; otherwise, we use received ID List with ID of current node appended as the ID List. The number of node information gathering round is stored in Round Number. The checksum of all bits is written in Check. Due to the fact that framework of this paper is independent with network layer protocol, the data is abstracted to pure digital signal in our following discussion.

4.2. Adversarial Model. We consider a setting with a polynomially bounded adversary capable of controlling a certain number of nodes completely. Once the adversary compromises a node, it can obtain all the node's secret keys and

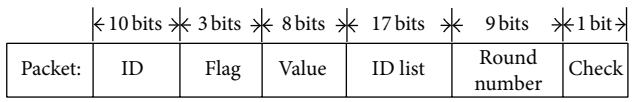


Figure 2: Format of data packet.

modify, forge, or discard messages or simply transmit false aggregation results, and its goal is to launch stealthy attacks [35] where the attacker's goal is to make T accept false aggregation results while not being detected.

4.3. PCSP. We assume that the final result X sensed by nodes is $N \times 1$ matrix. Disturbances $f(k_i, r)$ $(i = 1, 2, \ldots, N)$ are added to correlative element of X to ensure confidentiality, where r is the number of the round. Each node encrypts its sensory data x_i to $\mathrm{Enc}(x_i)$ which is transformed into linear projection Y_i on measurement matrix Φ_i. From the perspective of the whole network, the raw data X is changed to encrypted data $\mathrm{Enc}(X)$, which is transformed into compressed data Y. When final projection Y arrives at T through Internet from S, a perturbed orthogonal matching pursuit algorithm (POMP) is performed to recover the data $\mathrm{Enc}(\widehat{X})$, and then T should decrypt it to obtain original data $\widehat{X}$. Data transformation based on PCSP is shown in Figure 3. Our protocol can be divided into two major components expounded as follows.

4.3.1. Data Compression and Encryption during Free Routing. Before sensing from nodes, the trusted server should do some preparing work, as shown in Algorithm 2.

For node i, its task is to collect raw data, compute linear projection on measurement matrix, and forward message, which are described as follows.

In round r, i first senses raw data (like temperature) x_i and encrypts x_i to ciphertext:

$$\mathrm{Enc}\left(x_i\right) = x_i + f\left(k_i, r\right), \tag{5}$$

where k_i is the secret key of i and f is a hash function. We can see $\mathrm{Enc}(X)$ as

$$\mathrm{Enc}\left(X\right) = \left[\mathrm{Enc}\left(x_1\right), \mathrm{Enc}\left(x_2\right), \ldots, \mathrm{Enc}\left(x_N\right)\right]^T. \tag{6}$$

Then i computes its corresponding measurement coefficient matrix:

$$\Phi_i = \left[\phi_{1i}, \phi_{2i}, \ldots, \phi_{Mi}\right]^T \tag{7}$$

which is ith column in measurement matrix Φ. At last, i forwards signal (message):

$$Y_i = \Phi_i \times \mathrm{Enc}\left(x_i\right) \tag{8}$$

to the next node.

After receiving message Y_i, node j (using the same method to obtain Y_j) only needs to add its measurement Y_j to Y_i and sends the result to next hop until the last one sends data to S. The final compressed data Y ($M \times 1$ matrix) is transmitted to T through unsafe Internet, where

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{M1} & \phi_{M2} & \cdots & \phi_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}. \tag{9}$$

4.3.2. Data Recovery and Decryption Algorithm. For T, when compressed signal Y is received, it first computes sensing matrix A and utilizes POMP algorithm (see details in Algorithm 3) to reconstruct $\widehat{\theta}$, which is the sparse representation of $\mathrm{Enc}(\widehat{X})$, thereby $\mathrm{Enc}(\widehat{X})$ can be computed as

$$\mathrm{Enc}\left(\widehat{X}\right) = \Psi\widehat{\theta}. \tag{10}$$

Original data $\widehat{X}$ can be recovered by decrypting $\mathrm{Enc}(\widehat{X})$ employing the shared key between nodes and T.

Input: length N, key generation algorithm *keyGen*, original signal X:
begin
(1) round number $r = 1$;
(2) **for** $i \leftarrow 1$ to N **do**
$k_i = \mathrm{keyGen}(i)$;
distribute k_i to node i;
(3) construct Gabor dictionary parameter group $\langle s, N/2, w, v\rangle$;
(4) residual $\mathrm{res}_d = X$;
(5) **while** $\mathrm{res}_d >$ *threshold* **do**
$i = 1$;
$\mathrm{res}_i = \mathrm{res}_d$;
while *no* **do**
search with res_d and compute optimal subgroup: $\langle s, w, v\rangle$;
if $\langle s, w, v\rangle$ *has been chosen* **then**
acquire corresponding atomic dictionary;
else
generate new atomic dictionary;
search to find the optimal parameter u;
remove the chosen atoms in subgroup;
store the corresponding parameters;
orthogonal projection
$Pv = \Psi * (\Psi' * \Psi)\hat{}(-1) * \Psi' * \mathrm{res}_i$;
$\mathrm{res}_{i+1} = \mathrm{res}_i - Pv$;
$i++$;
$\Psi = \{g_{ri}(n)\}$;
orthogonal projection
$Pv = \Psi * (\Psi' * \Psi)\hat{}(-1) * \Psi' * \mathrm{res}_d$;
$\mathrm{res}_d = \mathrm{res}_d - Pv$;
(6) sparse representation
$S = (\Psi' * \Psi)\hat{}(-1) * \Psi' * X$;
(7) **Output** sparse matrix Ψ and sparsity k;
end

Algorithm 2: Initialization algorithm.

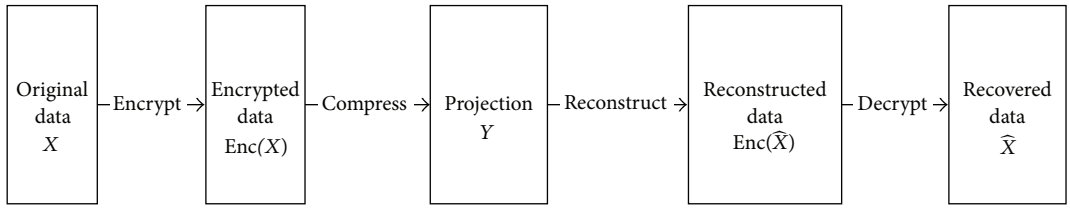


Figure 3: Data transformation based on PCSP.

Input: compressed signal Y, measurement matrix Φ, sparse matrix Ψ, key K, round number r and signal sparsity k:
begin
(1) sensing matrix $A = \Phi\Psi$;
(2) $\widehat{\theta} = \mathrm{OMP}(Y, A, k)$;
(3) $\mathrm{Enc}(\widehat{X}) = \Psi\widehat{\theta}$;
(4) **for** $i \leftarrow 1$ to length(Enc($\widehat{X}$)) **do**
$\widehat{X}_i = \mathrm{Enc}(\widehat{X}_i) - f(k_i, r)$;
Output $\widehat{X}$;
end

Algorithm 3: Perturbed orthogonal matching pursuit (POMP).

5. Security Analysis

Adversaries can compromise a fraction of nodes in sensor network. After a node i is compromised, its private information such as secret key k_i and ID will be leaked to adversary who can launch stealthy attack to make T accept false data without being detected.

We consider the situation where the adversary is trying to forge a valid $\mathrm{Enc}(x_i)$ without the knowledge of k_i. Apparently, the possibility relies on the pseudorandomness of the hash function f we chose and we believe the probability of generating an authentic $\mathrm{Enc}(x_i)$ is approximately $1/2^N$. Formally, our protocol is proved to be a chosen-plaintext attack secure based on Theorem 1.

Theorem 1. *If F is a pseudorandom function, the PCSP scheme is secure under a chosen-plaintext attack.*

Proof. Assume that f is a random function. We construct a new scheme which is exactly same as PCSP scheme, except that the pseudorandom function F is replaced by f. Since f is a random function, the probability that the adversary chooses the correct plaintext from the challenge cipher text is exactly $1/2$.

Now we consider the PCSP scheme in the chosen-plaintext attack. Here we define the probability that the adversary wins the chosen-plaintext attack: that is, $1/2 + \epsilon(n)$, where n is the security parameter. We then construct a distinguisher D to distinguish F and f as below: D runs the adversary to attack PCSP scheme under chosen-plaintext attack experiment.

(1) When a message x needs to be encrypted, D sends the adversary $x + F(k, r)$.

(2) When two plaintexts m_0 and m_1 are received, D flips a coin i, ($i = 0$ or 1), and sends the adversary $x_i + g(k, r)$. Here g is one of pseudorandom functions or random functions.

(3) When the output j of the adversary is received, D outputs $g = F$ if the adversary wins; otherwise, D outputs $g = f$.

From the viewpoint of D, if $g = F$, the probability that the adversary wins is $1/2 + \epsilon(n)$. Otherwise, the probability that the adversary wins is $1/2$, since the challenge cipher text is a random number. Therefore, the probability that D wins is $\epsilon(n)$. Finally, $\epsilon(n)$ must be negligible. □

6. Evaluation

In this section, we attempt to present the performance evaluation results on the real and simulative datasets. To evaluate the efficiency of our protocol, we follow the estimation error used in [36] to compare the accuracy among PCSP and three related algorithms (see details in Experiment 1). Later, we conduct simulation experiments with encryption/decryption and then encryption/decryption is removed in Experiment 2 for proving that our proposed protocol is effective to protect the confidentiality of data while preserving accuracy (as shown in Experiment 2).

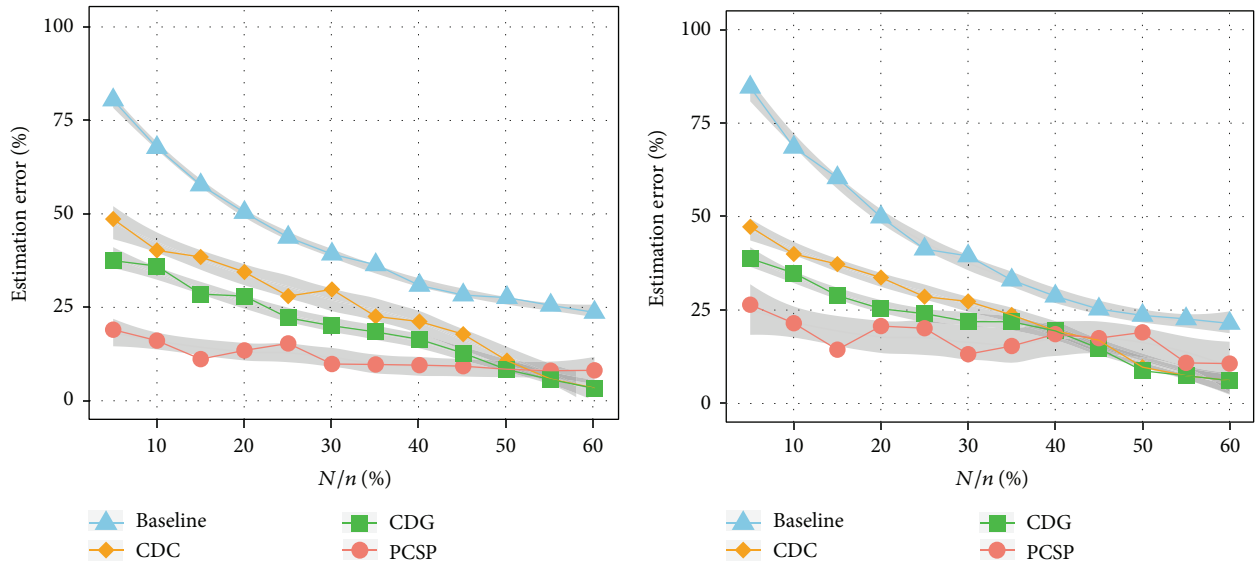
Experiment 1 (comparison with related algorithms on real datasets). Datasets used in this experiment contain NBDC-CTD [14] and InelLab [13], of which attributes are summarized in Table 1. We investigate performance of our method compared with the following state-of-art methods.

(1) *Baseline*. This algorithm uses basic routing and estimation methods, which is seen as baseline in [36]. Sensor node transmits packets to S using the shortest path. When S receives the final packet, it sends the final packet to information server, which takes advantage of the k-Nearest Neighbors (kNN) [37] Algorithm to recover the data.

(2) *CDG [38]*. In this framework, the following tree-based routing and traditional methods of CS for reconstructing the data collected from WSN are used. A sensor node will not send a packet to its parent node until receiving all packets from its children, so it collects all sensor readings to a packet. Convex optimization methods are used by information server to estimate the signal.

Table 1: Datasets for Experiment 1.

Name	Time period	Environment	Physical condition
IntelLab [13]	Feb. 28–Apr. 5, 2004	Indoor	Temperature, humidity, light
NBDC-CTD [14]	Oct. 26–Oct. 28, 2012	Ocean	Temperature, humidity, salt



(a) Estimation error on NBDC-CTD

(b) Estimation error on IntelLab

Figure 4: Comparison with related algorithms on real datasets.

(3) *CDC [36]*. Opportunistic routing with compression and a NSRP-based estimator are utilized in the CDC scheme. The compression scheme adds or subtracts the reading of last hop node as the packet travels towards S. Information server employs random linear projections of the orthonormal basis to estimate the coefficient vector to recover original data, because nonuniform sparse random projections (NSRP) used in compressing can preserve inner products within a small error.

We follow a classic evaluation criterion named as estimation error [36] (EE) defined in (11) and observe the performance of our method compared with CDG, CDC, and baseline algorithms:

$$\mathrm{EE} = \frac{\left\| X - \widehat{X} \right\|_2}{\left\| \widehat{X} \right\|_2}. \tag{11}$$

We run all of these algorithms 50 times and calculate the mean of their EE, respectively. A conclusion that the estimation error of our protocol is robust to the scale of the WSN can be inferred from Figure 4. As shown in Figures 4(a) and 4(b), our PCSP outperforms the competing algorithms when the number of alive nodes is small. In particularly, PCSP achieves estimation error as low as 18.89% and 26.39% on NBDC-CTD and IntelLab whilst results of other approaches are all higher.

Experiment 2 (comparison with encrypted and unencrypted data on simulative datasets). First of all, initialization algorithm (Algorithm 2) is run to start network sparse learning on encrypted and unencrypted data. In encryption process, nodes sense and encrypt data. Then we use pseudo-random Gaussian matrix to generate measurement matrix Φ and final signal Y is arrived at T. T takes advantage of POMP algorithm to obtain $\widehat{X}$. In the process without encryption, nodes just sense data. Then we make use of the measurement matrix Φ generated in encryption process, and then Y arrives at T. Later on, T runs OMP algorithm to reconstruct original signal $\widehat{X}$. If round number r is bigger than the threshold, then reinitialize the whole network. The parameters of two experiments are listed in Table 2.

To estimate the performance of our method compared with unencrypted data method, we employ EE mentioned in Experiment 1 and another criterion E defined in

$$E = \frac{\widehat{X} - X}{X}. \tag{12}$$

We conduct experiments 50 times in which the mean of EE is calculated, also original data, recovered data, encrypted data, compressed data, and estimation error EE as well as error E are recorded, as shown in Figure 5. Figure 5(a) indicates that original signal, recovered encrypted signal, and unencrypted signal keep the same trend. While Figure 5(b) presents the encrypted data, which cannot be utilized to speculate on the original data. Compressed result of encrypted data can be seen in Figure 5(c), whose dimensionality is lower than original data ($125 < 200$). As shown in Figure 5(d), estimation error of encrypted data has small variation with

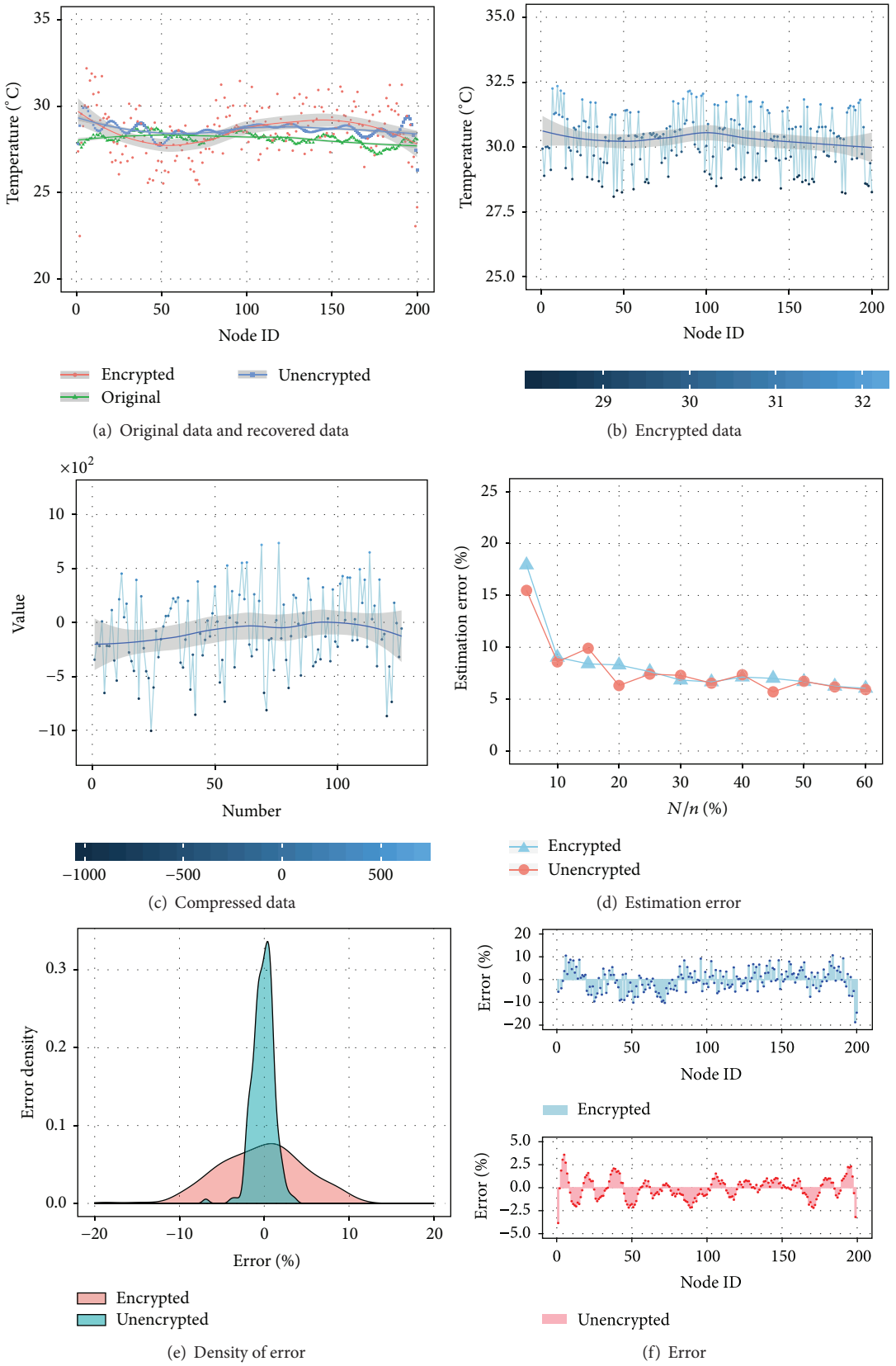


FIGURE 5: Comparison between encrypted data and unencrypted data.

Table 2: Parameter setting of Experiment 2.

Parameter	$N = 200$, threshold = 4
Data	Encrypting after randomly generating data $X \in [25, 30]$
Ψ	Search on Gabor overcomplete dictionary
Φ	Random Gaussian matrix, expectation = 0, variance = 1
Result of encrypted data	$k = 18$, estimation error = 25.444
Result of unencrypted data	$k = 20$, estimation error = 25.457

that of original data. We demonstrate error density of these two experiments in Figure 5(e) and details in Figure 5(f).

7. Discussion and Future Work

Crowd sensing by applying compressed sensing to WSN is an extremely complex task. Despite the fact that work done in this paper can initially perform the task with sensor energy balanced while preserving data privacy, some challenges still remain to be addressed. Firstly, network synchronization is necessary between WSN and T to obtain the number of rounds and keys for encryption/decryption. Another one is that our work only considers protecting data confidentiality rather than preserving data integrity and availability. Several improvements still need to be considered as follows. (1) The accuracy of reconstruction algorithm can be increased. (2) More security features (such as data integrity and availability) can be further studied.

8. Conclusion

In the context of crowd sensing in WSN, we proposed a perturbed compressed sensing protocol (PCSP) combined with compressed sensing technology to solve the issues about data confidentiality and sensor energy. Our protocol can be summarized into two components, in which encrypted data is obtained by perturbing sensor data gathered by each node; then, data compression by crowd sensing in WSN is enforced by linear projection utilizing compressed sensing. Afterwards, we presented performance analysis and security analysis along with experiments results which demonstrated that our protocol is capable of transmitting signal at a low energy cost while preserving data confidentiality. At last, we described limitations of our protocol with future work followed.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by National Natural Science Foundation of China no. 61272512 and no. 61300177 and Beijing Natural Science Foundation no. 4132054.

References

[1] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] L. Palopoli, R. Passerone, and T. Rizano, "Scalable offline optimization of industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 328–339, 2011.

[4] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, 2008.

[5] J. Bobin, J.-L. Starck, and R. Ottensamer, "Compressed sensing in astronomy," *IEEE Journal on Selected Topics in Signal Processing*, vol. 2, no. 5, pp. 718–726, 2008.

[6] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.

[7] YZ, *Design and research on CS-based wireless sensor network spatial sparse signals network models [Ph.D. thesis]*, Nankai University, 2012.

[8] A. Orsdemir, H. O. Altun, G. Sharma, and M. F. Bocko, "On the security and robustness of encryption via compressed sensing," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.

[9] Y. Rachlin and R. D. Baron, "The secrecy of compressed sensing measurements," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 813–817, IEEE, Urbana, Ill, USA, September 2008.

[10] A. M. Abdulghani and E. Rodriguez-Villegas, "Compressive sensing: from 'compressing while sampling' to 'compressing and securing while sampling'," in *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '10)*, pp. 1127–1130, Buenos Aires, Argentina, September 2010.

[11] R. Dautov and G. R. Tsouri, "Establishing secure measurement matrix for compressed sensing using wireless physical layer security," in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC '13)*, pp. 354–358, San Diego, Calif, USA, January 2013.

[12] M. Zhang, M. M. Kermani, A. Raghunathan, and N. K. Jha, "Energy-efficient and secure sensor data transmission using encompression," in *Proceedings of the 26th International Conference on VLSI Design and 12th International Conference on Embedded Systems (ES '13)*, pp. 31–36, IEEE, Pune, India, January 2013.

[13] http://www.select.cs.cmu.edu/data/labapp3/index.html.

[14] http://tao.ndbc.noaa.gov/.

[15] S.-T. Li and D. Wei, "A survey on compressive sensing," *Acta Automatica Sinica*, vol. 35, no. 11, pp. 1369–1377, 2009.

[16] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 211–231, 2009.

[17] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, vol. 3, pp. 1433–1452, Madrid, Spain, August 2006.

[18] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[19] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[20] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31, no. 1, pp. 59–73, 2011.

[21] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[22] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[23] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[24] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[25] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[26] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ1-regularized least squares," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[27] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[28] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[29] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, "Near-optimal sparse fourier representations via sampling," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pp. 152–161, ACM, 2002.

[30] A. C. Gilbert, M. J. Strauss, and R. Vershynin, "One sketch for all: fast algorithms for compressed sensing," in *Proceedings of the 39th ACM Symosium on the Theory of Computing (STOC '07)*, pp. 237–246, San Diego, Calif, USA, June 2007.

[31] A. M. Abdulghani and E. Rodriguez-Villegas, "Compressive sensing: from 'compressing while sampling' to 'compressing and securing while sampling'," *Proceedings of the 32rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1127–1130, 2010.

[32] Y. Rachlin and R. D. Baron, "The secrecy of compressed sensing measurements," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 813–817, September 2008.

[33] R. Dautov and G. R. Tsouri, "Establishing secure measurement matrix for compressed sensing using wireless physical layer security," in *Proceedings of the 10th International Conference on Computing, Networking and Communications (ICNC '13)*, pp. 354–358, San Diego, Calif, USA, January 2013.

[34] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal Processing*, vol. 86, no. 3, pp. 549–571, 2006.

[35] B. Przydatek, D. Song, and A. Perrig, "SIA: secure information aggregation in sensor networks," in *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 255–265, November 2003.

[36] X.-Y. Liu, Y. Zhu, L. Kong et al., "CDC: compressive data collection for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2188–2197, 2015.

[37] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[38] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 145–156, ACM, Beijing, China, September 2009.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 9825820, 10 pages
http://dx.doi.org/10.1155/2016/9825820

Research Article

Outdoor Air Quality Level Inference via Surveillance Cameras

Zheng Zhang, Huadong Ma, Huiyuan Fu, Liang Liu, and Cheng Zhang

Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Huadong Ma; mhd@bupt.edu.cn

Received 10 December 2015; Accepted 10 May 2016

Academic Editor: Tony T. Luo

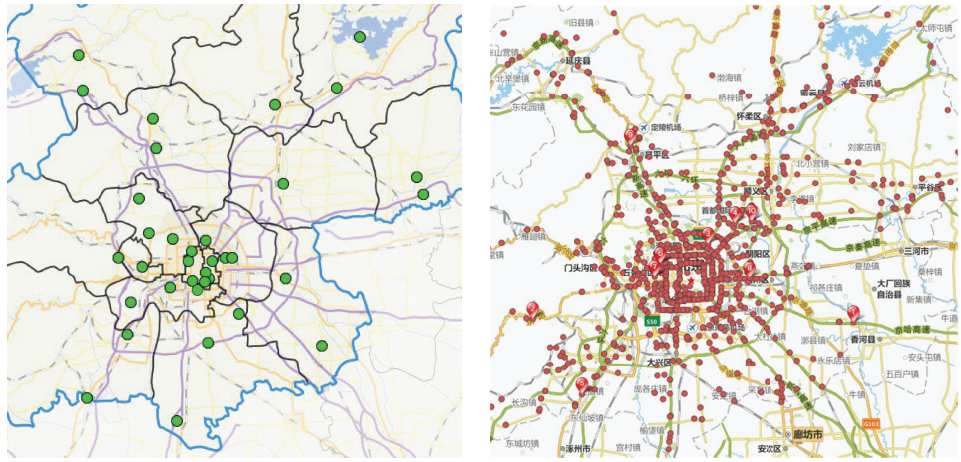
Air pollution is a universal problem confronted by many developing countries. Because there are very few air quality monitoring stations in cities, it is difficult for people to know the exact air quality level anytime and anywhere. Fortunately, large amount of surveillance cameras have been deployed and can capture image densely and conveniently. In this case, this provides the possibility to utilize surveillance cameras as sensors to obtain data and predict the air quality level. To this end, we present a novel air quality level inference approach based on outdoor images. Firstly, we explore several features extracted from images as the robust representation for air quality prediction. Then, to effectively fuse these heterogeneous and complementary features, we adopt multikernel learning to learn an adaptive classifier for air quality level inference. In addition, to facilitate the research, we construct an Outdoor Air Quality Image Set (OAQIS) dataset, which contains high quality registered and calibrated images with rich labels, that is, concentration of particles mass (PM), weather, temperature, humidity, and wind. Extensive experiments on the OAQIS dataset demonstrate the effectiveness of the proposed approach.

1. Introduction

Over the last few decades, many developing countries have suffered dramatic urbanization and industrialization processes in an unprecedented scale. In China, the population had increased to more than one million in more than 120 cities. This rapid growth in such a short period of time has caused serious complicated air pollution in China [1]. Due to the expensive cost of building and maintaining, the air quality monitoring stations cannot be placed on each block in cities. Figure 1(a) shows the distribution of air quality monitoring stations in Beijing (6336 square miles). In addition, there are only 10 air quality monitoring stations in Shanghai, which is the largest Chinese city in population and the largest urban area in population in the world. Moreover, air quality varies nonlinearly, so that the effective range of an air quality monitoring station is limited. We barely know the exact air quality on each block in metropolis by those sparse monitoring stations, so how to obtain the air quality fast and conveniently will attract much attention.

Many existing methods are based on satellite remote sensing technologies [2–6]. However, these methods only can reflect the air quality of atmosphere which is far from the ground air quality. Recently, some works focused on air quality inference via massive sensing data [7–11]. These works achieve good results at the cost of time consumption on the complex algorithms. Moreover, the massive sensing data used in these works are difficult to obtain.

With the development of Internet of Things (IoT) [12, 13], various sensors such as smart phones and cameras play important roles in urban sensing [14]. Different from the limited monitoring stations, there is large amount of surveillance and traffic cameras in many cities especially in urban area. For example, Figure 1(b) illustrates 3254 cameras in Beijing which are available in Sougo Map [15]. Furthermore, there is more than 400 thousand public cameras in Beijing [16], that is, nearly 20000 times more than monitoring stations, let alone the number of cameras in buildings. Therefore, we present a convenient and efficient air quality level inference approach based on multiple features and multiple kernel learning from single images via surveillance cameras. We first extract several features such as dark channel, medium transmission, sky color, power spectrum slope, contrast, and saturation from single images. In the previous work [17], we proposed an approach for air quality inference from image based on air quality index decision tree and SVM.



(a) (b)

Figure 1: (a) The distribution of air quality monitoring stations in Beijing. (b) The distribution of 3254 traffic cameras in Beijing which are available in Sougo Map.

To effectively fuse these heterogeneous and complementary features, in this paper, we utilize multikernel learning to learn an adaptive classifier based on multiple kernels. In addition, we build an Outdoor Air Quality Image Set dataset, which contains high quality registered and calibrated images captured by surveillance cameras. The dataset covers a wide range of daylight illumination and air pollution conditions. Each image in QAQIS has various data labels, including the concentration of particles mass ($PM_{2.5}$), weather, temperature, humidity, and wind, which are related to the air quality level.

To summarize, we develop an approach for air quality level inference from a single image with the following contributions:

(i) We propose a novel air quality level inference approach from a single image based on multiple features and multiple kernel learning.

(ii) We improve some existing algorithms for obtaining the features and take multiple kernel learning for air quality level inference.

(iii) We construct a dataset of high quality registered and calibrated images, which covers a wide range of daylight illumination and air quality conditions. It has the potential value for image processing and atmospheric sciences and can be used as a test bed for many algorithms.

The rest of this paper is organized as follows. Section 2 reviews previous approaches in brief. Section 3 presents several corresponding features of images and takes proper processing. In Section 4, multiple kernel learning will be used for training an adaptive classifier to fuse these heterogeneous and complementary features effectively. Next, Section 5 describes the dataset we use in this work and compares our approach with the traditional classification methods. Finally, a brief conclusion and future work will be given in Section 6.

2. Related Works

In this section, we discuss the existing works relevant to this research topic. Satellite remote sensing technologies have been widely used in the area of atmosphere air quality estimating. Van Donkelaar et al. [2] estimated the ground-level $PM_{2.5}$ for January 2001 to October 2002 using space-based measurements from the Moderate Resolution Imaging Spectroradiometer (MODIS) and the Multiangle Imaging Spectroradiometer (MISR) satellite instruments and additional information from a global chemical transport model (GEOS-CHEM). Liu et al. [3] used a generalized linear regression model to examine the relationship between ground-level $PM_{2.5}$ measurements and aerosol optical thickness from Multiangle Imaging Spectroradiometer (MISR) measurements in the eastern United States. Lamsal et al. [4] presented an approach to infer ground-level nitrogen dioxide (NO_2) concentrations by applying local scaling factors from GEOS-CHEM to tropospheric NO_2 columns retrieved from the Ozone Monitoring Instrument (OMI) on board the Aura satellite. Li et al. [6] proposed the Haze Optical Thickness retrieval model based on the assumption that surface reflectance varies slowly in a relative short period that could monitor the haze distribution and intensity for Beijing Olympic Games and help Beijing municipal government to carry out more measures to improve air quality conditions. Moumtzidou et al. [18] proposed a configurable semiautomatic framework for processing air quality and pollen forecast heatmaps. They integrated several existing environmental quality forecast data extraction tools with text processing and OCR (Optical Character Recognition) techniques tailored for heatmap analysis.

Recently, some works focused on air quality inference via massive sensing data. Zheng et al. [7, 8] inferred the air quality information based on the historical and real-time air quality data reported by existing monitor stations and a variety of data sources such as meteorology, traffic flow, human mobility, structure of road networks, and point

of interests. Chen et al. [9] proposed a big spatiotemporal data framework for the analysis of severe smog in China. They collected about 35,000,000 detailed historical and real-time air quality records (containing the concentrations of $PM_{2.5}$ and air pollutants including SO_2, CO, NO_2, O_3, and PM_{10}) and 30,000,000 meteorological records in 77 major cities of China through air quality and weather stations. It conducts scalable correlation analysis to find the possible short-term and long-term factors to $PM_{2.5}$. Li et al. [19] estimated 9 haze levels and 3 $PM_{2.5}$ levels based on images. For each image, they computed the transmission matrix and the depth map via established methods. Then, they analyzed the correlation between the haze levels with the official $PM_{2.5}$ record. However, the correlation between the haze levels and $PM_{2.5}$ is not stable due to the impact of weather or illumination. Hasenfratz et al. [10] developed land-use regression models to create pollution maps based on spatially resolved ultrafine particles dataset that is publicly available containing over 25 million measurements. The measurements were collected throughout more than one year using mobile sensor nodes installed on the top of public transport vehicles in the city of Zurich. Devarakonda et al. [11] proposed a method for air quality estimate from social media post such as Weibo text content based on a series of progressively more sophisticated machine learning models.

There were some vehicular-based works for collecting or measuring air quality. For example, Mei et al. [20] presented a vehicular-based mobile approach for measuring fine-grained air quality in real time and Al-Ali et al. [21] designed a wireless distributed mobile air pollution monitoring system which utilized city buses to collect pollutant gases.

However, existing methods are usually time-consuming in computation or difficult and expensive in data collection. Aiming at convenient and efficient air quality level inference, we extract multiple features from a single image and utilize multiple kernel learning to learn an adaptive classifier.

3. Feature Extraction

For inferring the air quality from images, we extract some features first. In traditional image classification tasks, the mainly used mid-level features such as SIFT, HOG, and LBP cannot describe the subtle and minute difference between images captured in the same scene. But what is even worse is the fact that some low-level features cannot distinguish the subtle difference, such as color. Different images may have the same pixel intensity in color spaces [22]. Unlike traditional methods, we adopt five discriminate features, that is, dark channel, medium transmission, sky color, power spectrum slope, contrast, and saturation, that are derived by analyzing the visual and spectral clues from images. These features are more suitable for the task of air quality inference from images.

3.1. Dark Channel and Medium Transmission. Particulate matter is one of the main air pollution sources. In the process of transmission, light intensity attenuated because of the particulate matter scattering. Figure 2 demonstrates the process of atmospheric light scattering and attenuation. In computer vision and computer graphics, the model widely used [23, 24] to describe the formation of a haze image is as follows:

$$I(x) = t(x) J(x) + (1 - t(x)) A, \tag{1}$$

where I denotes the observed intensity, J denotes the scene radiance, A denotes the atmospheric light, and t denotes the medium transmission describing the portion of the light that is not scattered and reaches the camera.

Haze is an atmospheric phenomenon where dust, smoke, and other particulate matters obscure the clarity of the sky. The concentration of haze can reflect the air quality level. He et al. [25] found that most local patches in haze-free outdoor images contain some pixels which have very low intensities in at least one color channel. That is to say, the minimum intensity in such a patch should have a very low value. Similarly, we compute the dark channel as

$$J(x) = \min_{c \in r,g,b} \left(\min_{y \in \Omega(x)} \left(J^c(y) \right) \right), \tag{2}$$

where J^c is a color channel of input image J and $\Omega(x)$ is a local patch centered at x. Then, we can estimate the medium transmission by

$$\widetilde{t}(x) = 1 - \min_c \left(\min_{y \in \Omega(x)} \left(\frac{J^c(y)}{A^c} \right) \right). \tag{3}$$

In our experiment, we resize the input images into 450 × 450. The patch is set to 45 × 45 for an image. We use a 101-dimensional feature vector to indicate the haze level. The 101-dimensional feature vector includes 100 dimension of the median value of dark channel intensities in these patches and 1 dimension of the medium transmission of the image.

3.2. Sky. Sky might be the most obvious features to indicate air pollution in images. As shown in Figure 3, the sunny images almost have a clear and blue sky, while the sky is gray in a haze image. For the sky part, firstly, we detect the sky region in an image with the method suggested in [22, 26]. First, we collect 20000 sky and non-sky patches, each of size 15 × 15. For each patch, we extract a 131-dimensional feature, which contains the SIFT descriptor and mean HSV color. Then, a random forest classifier is learned on the two-class patches. For an input image, we uniformly sample 15 × 15 patches and test their labels (sky or non-sky patch). Sky region can be segmented by implementing graph cuts on those patches. Then, we extract A and B channels in the LAB color space of the sky region to form a 200-dimensional feature vector.

3.3. Power Spectrum Slope. With the decrease of the air quality, the captured image becomes of low-resolution even blur. Due to the low-pass-filtering characteristic of a blurred region, some high frequency components are lost. So the amplitude spectrum slope of a blurred region tends to be steeper than that of an unblurred region. In order to analyze the impact of low-resolution of an image, we extract the power spectrum slope suggested in Liu et al.'s work [27]. First,

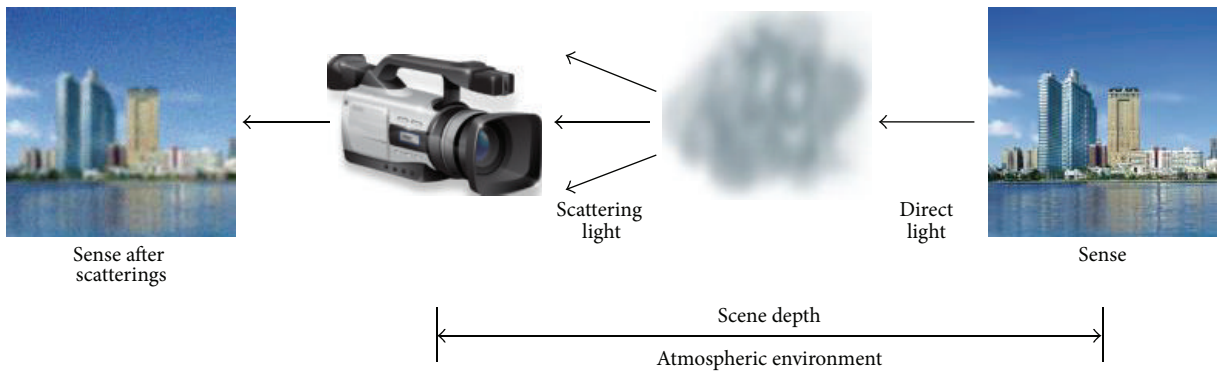


Figure 2: Atmospheric light scattering and attenuation.

Figure 3: (a) and (c) Input images of different weather and different air quality conditions. (b) and (d) The detected sky regions of these input images.

we compute the power spectrum of an image I with size $N \times N$ by taking the squared magnitude after Discrete Fourier Transform (DFT):

$$S(u,v) = \frac{1}{N^2} |I(u,v)|^2, \tag{4}$$

where $I(u,v)$ denotes the image transformed by DFT. We represent the two-dimensional frequency in polar coordinates, thus $u = f\cos\theta$ and $v = f\sin\theta$, f denotes the radius of power spectrum image, and θ denotes the angle of the polar coordinates, and we construct $S = (f,\theta)$. By summing the power spectra S over all directions θ and using polar coordinates, $S(f)$ can be approximated by

$$S(f) = \sum_{\theta} S(f,\theta). \tag{5}$$

Burton and Moorhead [28] found that $S(f)$ approximate to an exponential function of f, so slope of power spectrum α can be calculated as

$$\alpha \approx \ln B - \frac{\ln(S(f))}{\ln(f)}, \tag{6}$$

where B denotes a constant.

3.4. Contrast. There are various particulate matters in the atmosphere; light intensity attenuated during the transmission because of these particulate matters. Therefore, the same scenes in different air pollution conditions have different contrast. The Michelson contrast is commonly used for patterns where both bright and dark features are equivalent and take up similar fractions of the area.

However, the Michelson contrast does not consider the error due to noise points. We compute the contrast according to Root Mean Square (RMS) [29]:

$$C = \left(\frac{N_I \sum L_{(x,y)}^2 - \left(\sum L_{(x,y)} \right)^2}{{N_I}^2} \right)^{1/2}, \tag{7}$$

where $L_{(x,y)}$ denotes the luminance of the pixel (x, y) of the image and N_I denotes the number of pixels in the image. Therefore, we can obtain the contrast of an image as the one-dimensional feature vector.

3.5. Normalized Saturation. We also consider the color information of images for air pollution inference. As the saturation is independent of illumination, it can represent different images under various illumination conditions. For an image I, we calculate the normalized saturation for each pixel by

$$S_{(x,y)} = \frac{S_{x,y} - \min\left(S_I\right)}{\max\left(S_I\right) - \min\left(S_I\right)}, \tag{8}$$

where $\max(S_I)$ is the maximum saturation value and $\min(S_I)$ is the minimum saturation value of image I. For convenience of calculations in the following steps, we compute the histogram of the normalized saturation of an image to form the 10-dimensional feature vector.

4. Multiple Kernel Learning

There are several challenges during classification. For example, how to combine the features, how to choose the suitable kernels, and how to set the parameters of kernels. However, simple classifiers cannot handle the challenges well. Thus, a proper feature selection and fusion method is required for adapting the model to the specific problem. To effectively fuse these heterogeneous and complementary features that have different notions, we utilize multiple kernel learning [30] to learn an adaptive classifier by using multiple kernels. Instead of creating a new kernel, multiple kernel algorithms can be used to combine kernels already established for each individual feature. Moreover, multiple kernel learning is able to select for an optimal kernel and parameters from a larger set of kernels, reducing bias due to kernel selection.

Let $D_L = \{x_i, y_i\}_{i=1}^N$ be the training image dataset, where x_i denotes the ith sample and y_i denotes the corresponding class label and N is the number of training images. We aim to train a multikernel based classifier with a decision function $f(x)$ to predict the air quality level of an unlabeled image x. In this paper, we use some linearly combined base kernel functions to determine an optimal kernel function:

$$k\left(x_i, x\right) = \sum_{m=1}^{M} \beta_m k_m\left(x_i, x\right), \tag{9}$$

where β_m is one of the linear combination coefficients, $\sum_{m=1}^{M} \beta_m = 1$, and $\beta_m \geq 0$. Given the input feature x of the image, the decision function is defined as follows:

$$f\left(x\right) = \sum_{m=1}^{M} \beta_m k_m\left(x\right) \alpha + b, \tag{10}$$

where α and b are the parameters of the standard SVM.

In this paper, we adopt the simple multiple kernel learning [31], so the objective function can be formulated as follows:

$$\begin{aligned} \min_{\beta,\alpha,b} \quad & J = \frac{1}{2} \sum_{m=1}^{M} \beta_m \alpha^T K_m \alpha + C \sum_i \xi_i, \\ \text{s.t.} \quad & y_i \cdot \sum_{m=1}^{M} \beta_m K_m\left(x_i\right) \alpha + y_i b \geq 1 - \xi_i \quad \forall i, \\ & \xi_i \geq 0 \quad \forall i, \end{aligned} \tag{11}$$

in which $K_m(x_i) = [k_m(x_i, x_1), \ldots, k_m(x_i, x_p)]$ and p is the number of air quality levels. We adopt the gradient descent algorithm to solve the optimization problem in (11). For each iteration, α and b can be obtained by the given weight β. Then, we can recalculate β using α and b. We adopt one-against-all strategy to transform the multiclass classification into two-class classification. If there are P classes, then the objective function can be rewritten as

$$J = \sum_{p=1}^{P} J_p\left(\beta, \alpha_p, b_p\right), \tag{12}$$

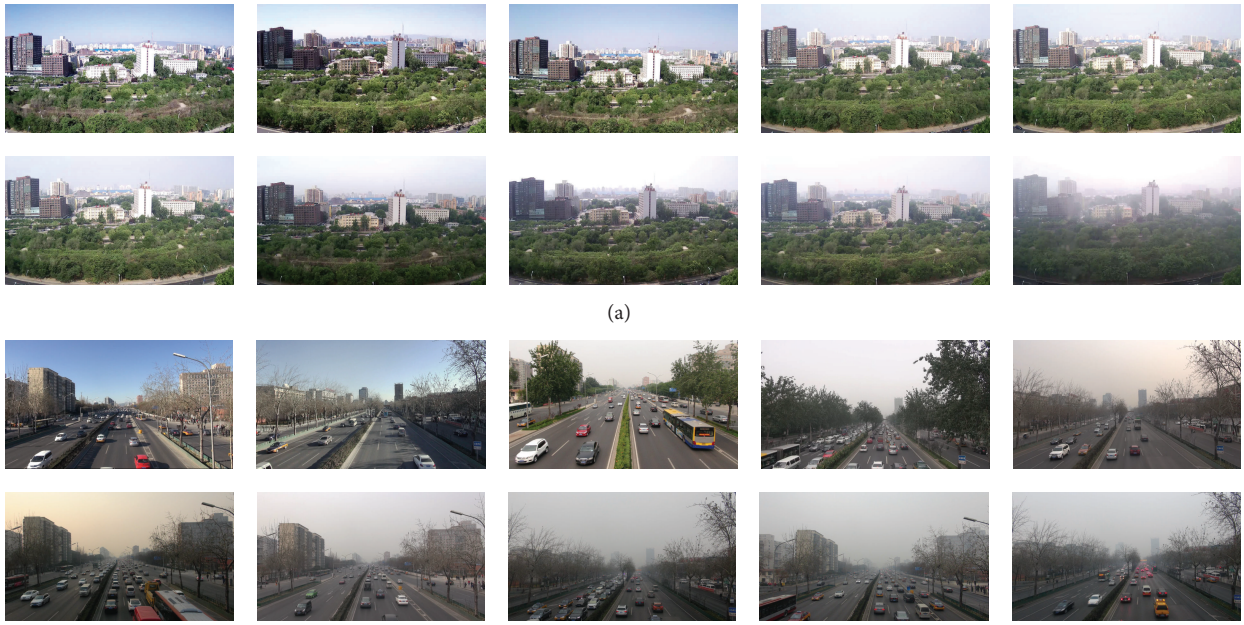
where J_p is a two-class classifier, the positive samples are the samples with class label p and the negative samples are the samples with other labels. We can obtain the class label by

$$y = \arg\max_{y_p} f_p\left(x\right). \tag{13}$$

Compared with traditional classifiers, multiple kernel learning can improve the accuracy by learning an optimal kernel combination. By comparing the weights of different kernels, we can adopt the effective features and abandon less effective features. Thus, we can reduce the time consumption on feature calculation.

5. Dataset and Experiments

5.1. Dataset. We construct a dataset with high quality registered and calibrated images named Outdoor Air Quality Image Set (OAQIS) to evaluate our approach. All of the images are captured in Beijing, suffering serious air pollution. In addition, there are two scenes in OAQIS. Scene A is captured automatically from 8:00 am to 6:00 pm by a camera



(a)

(b)

Figure 4: Exemplar images from OAQIS. (a) Scene A: the images are captured by a camera installed in a building. (b) Scene B: the images are captured by a camera fixed on a pedestrian bridge. The concentration of $PM_{2.5}$ in the images changes from low to high.

installed in No. 3 Teaching Building of BUPT, as shown in Figure 4(a). Scene B is captured randomly from 9:00 am to 15:00 pm by a traffic camera fixed on the first pedestrian bridge which is on the east side of Jimen flyover, as shown in Figure 4(b). The dataset covers a wide range of daylight illumination conditions and air pollution conditions. The spatial resolution of each image is 1280×720 pixels.

5.2. Ground Truth. We have tagged our images with a variety of ground truth information. The most important categories of the ground truth we collected are the concentration of particles ($PM_{2.5}$) in the air and weather information.

We collect the concentration of particles in the air by two air quality monitors made by Suzhou Beiang Technology Co., Ltd., and Zhongzhiwuxian Trade Co., Ltd., as shown in Figure 5. We also collect illumination intensity by an optical sensor for potential uses in the future. We gather $PM_{2.5}$ data every five minutes in an hour and compute the average value.

Then, we set the value as the ground truth for all images captured in the hour. We automatically collect standard weather information from Weather China website [32]. This includes information about weather condition (sunny, cloudy, overcast, misty, foggy, hazy, rainy, snowy, etc.), temperature, humidity, and wind, as shown in Figure 6.

Air quality index (AQI) is affected by several atmospheric pollutants. $PM_{2.5}$ is an important measure of AQI, so we use $PM_{2.5}$ to indicate AQI level. In many countries, AQI is divided into six levels indicating increasing health concern as shown in Figure 7. An AQI value over 300 means hazardous air quality, whereas if it is below 50 the air quality is good.

Table 1: Results related to features.

Features	F_{dm}	F_{sk}	F_{po}	F_{co}	F_{sa}	F_{all}
Accuracy	0.67	0.72	0.63	0.42	0.67	0.84

5.3. Experimental Results. We evaluate our approach on OAQIS, and the performance on the dataset demonstrates the effectiveness of the proposed approach. In our experiment, we fuse these features to form a 313-dimensional vector for an image. In MKL, we use 5 linear base kernels, respectively, constructed for the multiple features. First, we evaluate our approach on scene A. We divide images into 10 parts, while choosing 9 parts as the training set, leaving one as the testing part. Then we use 10-fold cross-validation to test 10 times, and use the average result as the final result.

We justify the performance of features described in Section 3. F_{dm}, F_{sk}, F_{po}, F_{co}, and F_{sa} are the notations of dark channel and medium transmission, sky color, power spectrum slope, contrast, and saturation. As shown in Table 1, the accuracy of these features is improved a lot by using all of the features, and the sky color feature has the best performance among the five features. In addition, the dark channel and medium transmission feature, power spectrum slope feature, and saturation feature also achieve good results. So the weights are different, in accordance with the performance of these features.

Table 2 shows the inference results on six image categories with different air quality levels. We can notice that the results of air quality inference are understandable. Due to the lack of sample images, the category with very unhealthy air quality

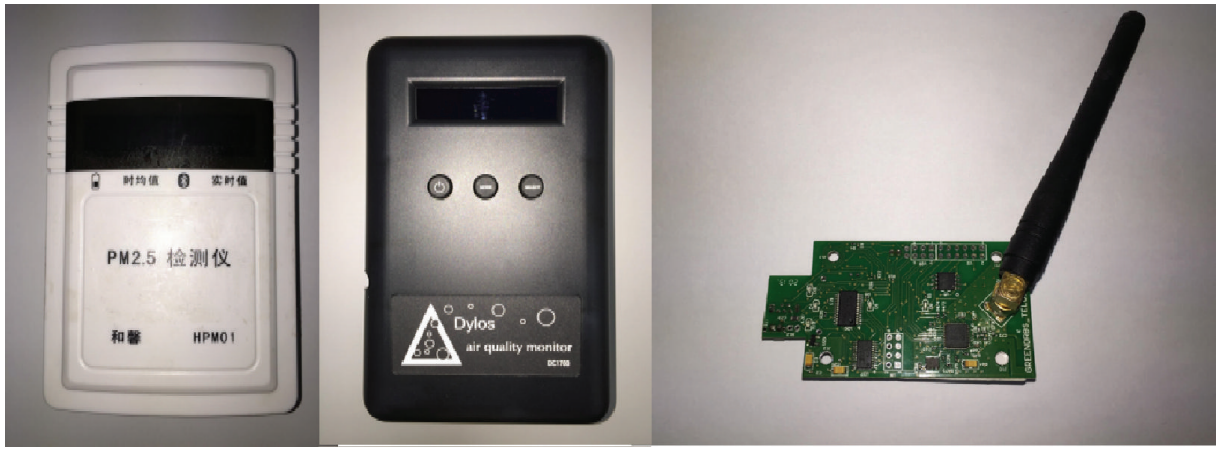


Figure 5: The air quality monitors and optical sensor we used for labeling images.

Conditions at 2014.05.12 10:33 am	
$PM_{2.5}$	34 $\mu g/m^3$
Weather	Sunny
Temperature	73.4 F/23.0°C
Humidity	44%
Wind	NW 1.6–3.3 m/s

Figure 6: Sample ground truth data. The image was captured at 12 May 2014 in Beijing. The concentration of $PM_{2.5}$ acquired by our air quality monitor and weather data obtained from Weather China website.

Table 2: Inference results on different air quality levels.

Air quality levels	Accuracy
Good	0.72
Moderate	0.90
Unhealthy for sensitive groups	0.81
Unhealthy	0.90
Very unhealthy	0.64
Hazardous	0.84

Table 3: Inference results of different method on the OAQIS dataset.

	SVM	AdaBoost	[17]	Our approach
Accuracy	0.70	0.33	0.77	0.84

gets lower accuracy of 64 percent while the results of other categories are all above 70 percent. Specially, the categories with hazardous and unhealthy air quality achieve impressive performance with 90 percent accuracy.

The confusion table of inference results is depicted in Figure 8. Green, yellow, orange, red, purple, and maroon are corresponding to six air quality levels mentioned before. For some air quality levels, such as moderate and unhealthy, our approach achieves excellent performances.

Table 3 presents the comparisons between our approach and the baseline methods. To show the best performance of all methods, every method produced multiple results using a group of reasonable parameters. The first baseline is to implement SVM directly on the 313-dimensional feature. The second baseline is the traditional Adaboost, which combines several classifiers to build a stronger classifier. Experimental result shows that our approach outperforms the baseline methods.

Table 4: Efficiency study.

Procedures	Features					Inference	Total
	F_{dm}	F_{sk}	F_{po}	F_{co}	F_{sa}		
Time (s)	5.29	11.62	0.37	8.92	0.91	0.22	27.33

In order to evaluate our approach, we randomly select five days in OAQIS. Figure 9 shows the measured values of $PM_{2.5}$ and the inference result of $PM_{2.5}$ by the proposed approach at 9:00–11:00 am and 2:00–4:00 pm on May 26–30, 2014. The measured values were collected every 5 minutes by an air quality monitor made by Zhongzhiwuxian Trade Co., Ltd., during the experiment period. We calculate the mean value of each hour, and set it as the $PM_{2.5}$ label of the images captured in the same period. Some measured values of $PM_{2.5}$ on 29 May change a lot within a short time, but the mean value of that period is not so big, so the inference result is consistent with the correct air quality level. We also evaluate our approach on scene B; the inference accuracy is 88.26%.

AQI values	Levels of health concern	Colors	Health implications
0–50	Good	Green	No health implications
51–100	Moderate	Yellow	Few hypersensitive individuals should reduce outdoor exercise
101–150	Unhealthy for sensitive groups	Orange	Slight irritations may occur, and individuals with breathing or heart problems should reduce outdoor exercise
151–200	Unhealthy	Red	Slight irritations may occur, and individuals with breathing or heart problems should reduce outdoor exercise
201–300	Very unhealthy	Purple	Healthy people will be noticeably affected. People with breathing or heart problems will experience reduced endurance in activities. These individuals and elders should remain indoors and restrict activities
301–500	Hazardous	Maroon	Healthy people will experience reduced endurance in activities. There may be strong irritations and symptoms and may trigger other illnesses. Elders and the sick should remain indoors and avoid exercise. Healthy individuals should avoid out door activities

Figure 7: AQI values, descriptors, colors, and health implications.

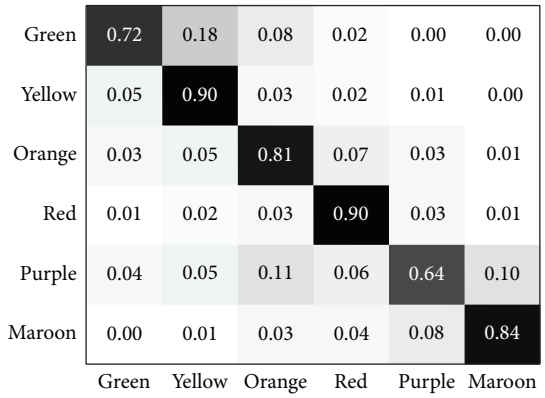


Figure 8: Confusion table of the proposed approach on OAQIS dataset.

The accuracy result shows the effectiveness of fusing multiple features for air quality level inference on images.

5.4. Efficiency. The experiments were evaluated on a 64-bit PC with a Dual-Core CPU @3.20 GHz and 4 GB memory. We use Matlab R2013b to extract features from images, and use Visual Studio 2013 for the inference part.

In our first experiment, we compare the time required to extract different features from images and the procedure of inference. As shown in Table 4, on average, the consumptions of feature extraction are 3.0136 seconds, 0.3661 seconds, 8.9181 seconds, and 0.9068 seconds, the inference procedure requires 0.029 seconds, and the total time cost of the whole procedure is 13.2336 s. Compared with existing methods, our approach can estimate air quality from a single image with reasonable cost.

6. Conclusion

We have presented an convenient and efficient approach for air quality inference from a single image. Our approach is based on multiple features and multiple kernel learning. We first extracted several features such as dark channel, medium transmission sky color, power spectrum slope, contrast, and saturation from images. To effectively fuse these heterogeneous and complementary features, we utilized multikernel learning to learn an adaptive classifier based on multiple kernels. We collected a dataset of high quality registered and calibrated images named OAQIS. The dataset covers a wide range of daylight illumination and air pollution conditions and has potential implications for image processing and atmospheric sciences and can be used as a test bed for many algorithms. We evaluated our approach on the dataset, and the results show the effectiveness of our approach. In the future, we will extend our dataset and evaluate our approach on more scenes. We will explore fine-grained air quality inference approaches from unrestricted images and consider the influence of wind, humidity, and weather on air quality inference.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The research reported in this paper is supported by the National Natural Science Foundation of China under Grant no. 61332005 and no. 61190114, The Funds for Creative Research Groups of China under Grant no. 61421061, The Cosponsored Project of Beijing Committee of Education, The Beijing Training Project for the Leading Talents in S&T

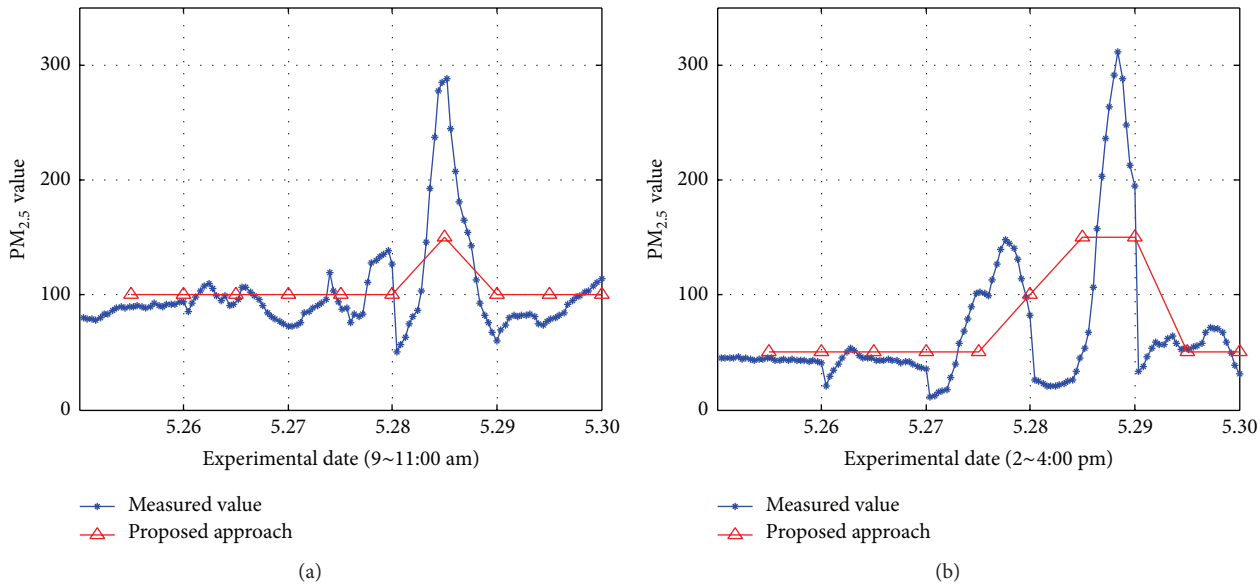


FIGURE 9: Experimental result of the proposed approach on 2014.5.26–2014.5.30 at 9:00–11:00 am and 2:00–4:00 pm.

(ljrc 201502), and Beijing University of Posts and Telecommunications Foundation of Youth Science and Technology Innovation under no. 500401132.

References

[1] J. Li, Z. Wang, G. Zhuang, G. Luo, Y. Sun, and Q. Wang, "Mixing of Asian mineral dust with anthropogenic pollutants over East Asia: a model case study of a super-duststorm in March 2010," *Atmospheric Chemistry and Physics*, vol. 12, no. 16, pp. 7591–7607, 2012.

[2] A. Van Donkelaar, R. V. Martin, and R. J. Park, "Estimating ground-level $PM_{2.5}$ using aerosol optical depth determined from satellite remote sensing," *Journal of Geophysical Research: Atmospheres*, vol. 111, no. 21, 2006.

[3] Y. Liu, J. A. Sarnat, V. Kilaru, D. J. Jacob, and P. Koutrakis, "Estimating ground-level $PM_{2.5}$ in the eastern United States using satellite remote sensing," *Environmental Science and Technology*, vol. 39, no. 9, pp. 3269–3278, 2005.

[4] L. N. Lamsal, R. V. Martin, A. van Donkelaar et al., "Ground-level nitrogen dioxide concentrations inferred from the satellite-borne ozone monitoring instrument," *Journal of Geophysical Research: Atmospheres*, vol. 113, no. D16, 2008.

[5] R. V. Martin, "Satellite remote sensing of surface air quality," *Atmospheric Environment*, vol. 42, no. 34, pp. 7823–7843, 2008.

[6] S. Li, L. Chen, F. Zheng, D. Han, and Z. Wang, "Design and application of haze optic thickness retrieval model for Beijing olympic games," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '09)*, vol. 2, pp. II507–II510, IEEE, Cape Town, South Africa, July 2009.

[7] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: when urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*, pp. 1436–1444, Chicago, Ill, USA, August 2013.

[8] X. Chen, Y. Zheng, Y. Chen et al., "Indoor air quality monitoring system for smart buildings," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 471–475, Seattle, Wash, USA, September 2014.

[9] J. Chen, H. Chen, J. Z. Pan, M. Wu, N. Zhang, and G. Zheng, "When big data meets big smog: a big spatio-temporal data framework for China severe smog analysis," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data (BigSpatial '13)*, pp. 13–22, ACM, Orlando, Fla, USA, November 2013.

[10] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the spatio-temporal resolution limit of urban air pollution maps," in *Proceedings of the 12th IEEE International Conference on Pervasive Computing and Communications (PerCom '14)*, pp. 69–77, March 2014.

[11] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," in *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing (UrbComp '13)*, article 15, ACM, Chicago, Ill, USA, August 2013.

[12] H.-D. Ma, "Internet of things: objectives and scientific challenges," *Journal of Computer Science and Technology*, vol. 26, no. 6, pp. 919–924, 2011.

[13] H.-D. Ma, L. Liu, A. Zhou, and D. Zhao, "On networking of internet of things: explorations and challenges," *IEEE Internet of Things Journal*, 2015.

[14] L. Liu, W.-Y. Wei, D. Zhao, and H.-D. Ma, "Urban resolution: new metric for measuring the quality of urban sensing," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2560–2575, 2015.

[15] Sougo Map, http://map.sogou.com/store/beijing.

[16] Sohu News, http://news.sohu.com/20100422/n271668614.shtml.

[17] Z. Zhang, H.-D. Ma, H.-Y. Fu, and X.-P. Wang, "Outdoor air quality inference from single image," in *Proceedings of the 21st International Conference on MultiMedia Modeling*, pp. 13–25, Sydney, Australia, January 2015.

[18] A. Moumtzidou, V. Epitropou, S. Vrochidis et al., "Environmental data extraction from multimedia resources," in *Proceedings of the 1st ACM International Workshop on Multimedia Analysis for Ecological Data*, pp. 13–18, November 2012.

[19] Y. Li, J. Huang, and J. Luo, "Using user generated online photos to estimate and monitor air pollution in major cities," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, article 79, ACM, Hunan, China, August 2015.

[20] S. Mei, H. Li, J. Fan, X. Zhu, and C. R. Dyer, "Inferring air pollution by sniffing social media," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '14)*, pp. 534–539, Beijing, China, August 2014.

[21] A. R. Al-Ali, I. Zualkernan, and F. Aloul, "A mobile GPRS-sensors array for air pollution monitoring," *IEEE Sensors Journal*, vol. 10, no. 10, pp. 1666–1671, 2010.

[22] C. Lu, D. Lin, J. Jia, and C.-K. Tang, "Two-class weather classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 3718–3725, June 2014.

[23] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *International Journal of Computer Vision*, vol. 48, no. 3, pp. 233–254, 2002.

[24] R. T. Tan, "Visibility in bad weather from a single image," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[25] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011.

[26] L. Tao, L. Yuan, and J. Sun, "SkyFinder: attribute-based sky image search," *ACM Transactions on Graphics*, vol. 28, no. 3, article 68, 2009.

[27] R. Liu, Z. Li, and J. Jia, "Image partial blur detection and classification," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

[28] G. J. Burton and I. R. Moorhead, "Color and spatial structure in natural scenes," *Applied Optics*, vol. 26, no. 1, pp. 157–170, 1987.

[29] E. Peli, "Contrast in complex images," *Journal of the Optical Society of America A: Optics and image science*, vol. 7, no. 10, pp. 2032–2040, 1990.

[30] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, p. 6, ACM, can, July 2004.

[31] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[32] Weather China, http://www.weather.com.cn/.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 6406981, 14 pages
http://dx.doi.org/10.1155/2016/6406981

Research Article

Share the Crowdsensing Data with Local Crowd by V2V Communications

Chao Song,[1,2] Ming Liu,[1,2] and Xili Dai[1,2]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Chao Song; chaosong@uestc.edu.cn

Received 12 December 2015; Revised 25 February 2016; Accepted 31 March 2016

Academic Editor: Qirong Ho

With an increase in the number of mobile applications, the development of mobile crowdsensing systems has recently attracted significant attention from both academic researchers and industries. In mobile crowdsensing system, the remote cloud (or back-end server) harvests all the crowdsensing data from the mobile devices, and the crowdsensing data can be uploaded immediately via 3G/4G. To reduce the cost and energy consumption, many academic researchers and industries investigate the way of mobile data offloading. Due to the sparse distribution of the WiFi APs, offloading the crowdsensing data is often delayed. In this paper, compared with offloading data via WiFi APs, we investigate the communication and sharing of crowdsensing data by vehicles near the event (such as a pothole on the road), termed as a local crowd. In such crowd, a vehicle can transmit the data to each other by vehicle-to-vehicle (V2V) communication. The crowd-based approach has a lower delay than the offloading-based approach, by considering the quality of truth discovery. We define a utility function related to the crowdsensing data shared by the local crowd in order to quantify the trade-off between the quality of the truth discovery and the user satisfaction. Our extensional simulations verify the effectiveness of our proposed schemes.

1. Introduction

Over 6.8 billion mobile phones were in use all over the world in 2013 [1]. With an increase in the number of mobile applications, many outdoor mobile applications are getting a lot of attention from both academic researchers and industries [2], such as Waze and pothole detection [3]. The ubiquity of smartphones has led to the emergence of mobile crowdsensing (MCS) tasks such as the detection of spatial events when smartphone users move around in their daily lives [4].

The cloud-based architecture has been widely used in the mobile crowdsensing system. A mobile device senses an event and then generates the crowdsensing data to describe it, termed as *report*. The mobile device communicates with the remote cloud (or back-end server) to upload its report. The back-end server harvests all the reports from these mobile devices and aggregates them to discover the truth of the event. Finally, the back-end server publishes the feedback result to the querying users, who are interested in the event. The communication between the mobile devices and the back-end server can be immediate via the cellular network (3G/4G).

However, the problem of crowdsensing with cellular network is the increase in traffic demand and high energy consumption. Furthermore, the tasks of crowdsensing, which include gathering raw data and querying the services, can severely increase the overhead of the cloud, such as the computation, the storage, and the bandwidth. For the users, mobile data offloading through WiFi APs has demonstrated its feasibility in reducing the data burden on the cellular networks. More recently, delayed offloading has been proposed [5]: if there is currently no WiFi availability, (some) traffic can be delayed instead of being sent/received immediately over the cellular interface. Although the delayed offloading can reduce the cost of mobile data communication, this approach obviously increases the data delivery delay, so as to decrease the user satisfaction.

In this paper, we investigate the communication and sharing of the mobile crowdsensing data under the traditional cloud-based architecture. We propose a distributed architecture, which does not require communicating with the remote cloud. The crowdsensing data is disseminated to all the vehicles near the event in a predefined range, and we term these vehicles as the *local crowd* of this event. In such crowd, a vehicle can transmit the data to each other by vehicle-to-vehicle (V2V) communication. Like the remote cloud, the local crowd harvests the crowdsensing data of this event and then aggregates them to discover the truth of the event. Finally, the crowd sends the feedback results to the querying users, who move in this crowd. Thus, the communication and sharing of the crowdsensing data all happen in the local crowd, in order to reduce the data delivery delay. We define a utility function related to the crowdsensing data shared by the local crowd, in order to quantify the trade-off between the quality of the truth discovery and the user satisfaction. Compared with the cloud-based approach, we model the quality of the truth discovery by Kullback-Leibler divergence (or relative entropy). Our extensional simulations verify the effectiveness of our proposed schemes.

The remainder of this paper is organized as follows: Section 2 surveys the related work; Section 3 introduces our crowdsensing-based system and discusses the different approaches for communicating and sharing the crowdsensing data; in Section 4, we model the problem and propose an algorithm to optimize it; Section 5 evaluates the performance of the proposed approach; and the last section concludes this paper.

2. Related Work

We present related work in the following two parts, which are mobile crowdsensing and mobile data offloading.

2.1. Mobile Crowdsensing. Mobile phone sensing is a paradigm which takes advantage of the pervasive smartphones to collect and analyze data beyond the scale of what was previously possible. Yang et al. in [6] investigate novel sensors integrated in modern mobile phones and leverage user motions to construct the radio map of a floor plan, which was previously obtained only by site survey. Zhou et al. in [7] investigate the application of the prediction for the bus arrival time. They do not require the absolute physical location reference, and they mainly wardrive the bus routes and record the sequences of observed cell-tower IDs, which reduces the initial construction overhead. Yang et al. in [8] design incentive schemes for mobile phone sensing, with two system models: the platform-centric model, where the platform provides a reward shared by participating users, and the user-centric model, where users have more control over the payment they will receive. He et al. in [9] investigate the optimal task allocation and show that the allocation problem is NP hard. They also discuss how to decide fair prices of sensing tasks to provide incentives, since mobile users tend to decline the tasks with low incentives.

2.2. Mobile Data Offloading. This increase in traffic demand is overloading cellular networks, forcing them to operate close to (and often beyond) their capacity limits [5]. A more cost-effective way to cope with the problem of highly congested mobile networks is by offloading some of the traffic through Femtocells and the use of WiFi. There exist two types of WiFi offloading. The usual way of offloading is on-the-spot offloading: when there is WiFi available, all traffic is sent over the WiFi network; otherwise, all traffic is sent over the cellular interface. More recently, delayed offloading has been proposed: if there is currently no WiFi availability, (some) traffic can be delayed instead of being sent/received immediately over the cellular interface. In the simplest case, traffic is delayed until WiFi connectivity becomes available. A more interesting case is when the user (or the device on her behalf) can choose a deadline (e.g., per application and per file). If up to that point no AP is detected, the data are transmitted through the cellular network [10, 11]. The authors in [12] define a utility function related to delayed offloading to quantitatively describe the trade-offs between user satisfaction in terms of the price and the experienced delay of waiting for WiFi connectivity. Ristanovic et al. in [13] propose two algorithms for delay-tolerant offloading of bulky, socially recommended content from 3G networks. The first one (called MixZones) uses opportunistic, ad hoc transfers between the users, and the second one (called HotZones) exploits delay tolerance and tries to download contents when users are close to WiFi access points.

3. Problem Statement

In this section, we first take our crowdsensing-based system, called Follow Us (FU), as an example to demonstrate the typical cloud-based architecture. The system FU is a road traffic condition monitoring and alerting application. Then, we discuss the communication and sharing of the crowdsensing data with the local crowd, which is a distributed approach.

3.1. Remote Cloud. For processing the crowdsensing data, one effective approach is to utilize a centralized architecture. A typical architecture of crowdsensing-based system consists of a centralized back-end server and a collection of mobile users. The mobile users have two responsibilities: (1) *probing user*, who uses mobile phones as well as the sensors to sense and report the events to the back-end server, and (2) *querying user*, who queries the information of the nearby events. The back-end server is responsible for collecting the reports of the events from the probing vehicles and intellectually aggregates such information. The smartphones in the system are assumed to be cooperative, which belong to or are affiliated to the system, willing to take sensing tasks and provide sensing services to the system. The issue of participation incentive [14, 15] of a rational or even strategic smartphone user is out of the scope of the paper.

In order to demonstrate the communication and sharing in crowdsensing-based system, we take a system as an example. One attractive example is a traffic monitoring application that uses a number of active probing vehicles to sense the road condition. To monitor the road condition and alert the anomalies, we have developed a crowdsensing-based system, called Follow Us (FU). The basic idea of FU is that the vehicles

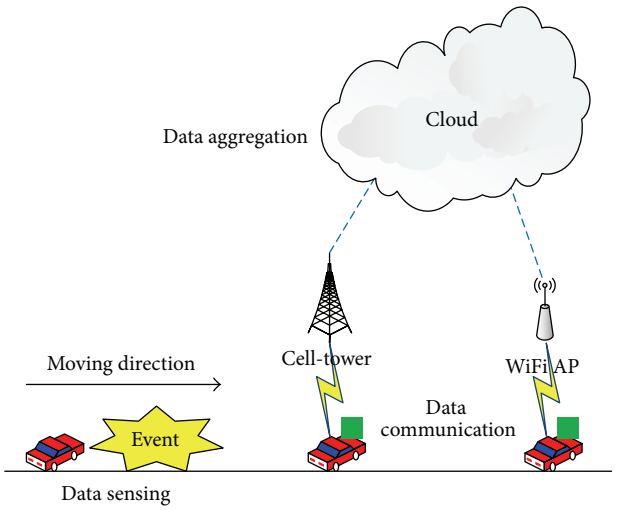


Figure 1: The cloud-based architecture of a mobile crowdsensing application.

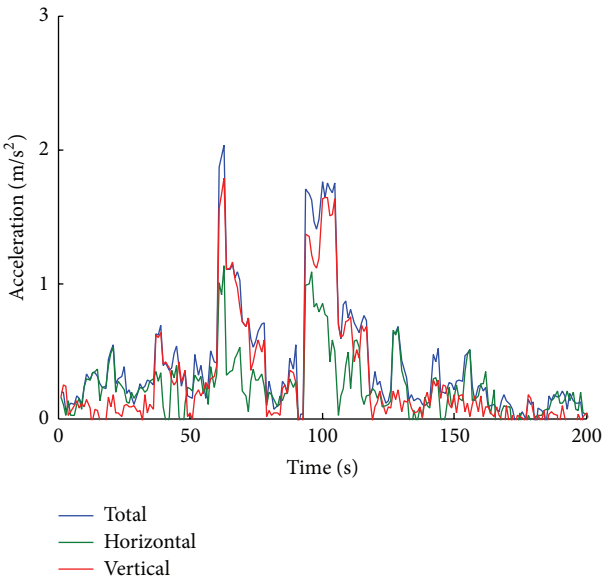


Figure 2: An example of sensing event.

moving ahead can report the road conditions to those behind. As shown in Figure 1, this application includes three primary parts as follows:

(i) Data sensing: the smartphones in the probing vehicles sense the events on the road. As an example in Figure 1, when a vehicle meets an obstacle on the road, the driver drives the vehicle to avoid the obstacle. Thus, the smartphone mounted in the vehicle senses the anomaly of the motion of the vehicle by its accelerometer.

(ii) Data communication: the sensing data are uploaded to the back-end server, and the feedback results will be downloaded from the server. In Figure 1, the smartphone on the vehicle will upload the report of this anomaly to the back-end server when it has an opportunity of connection to Internet (e.g., 3G/4G or WiFi AP).

(iii) Data aggregation: the back-end server harvests and aggregates all the uploaded sensing data to obtain the feedback results to the querying users. The back-end server harvests the reports from all the probing vehicles and aggregates them by the algorithm of truth discovery. As a result, the back-end server publishes the report to the vehicles heading for the place of the anomaly.

Figure 2 is an example of sensing data in FU. When the vehicle drives through the road with speed bumps, it will be shaken. The smartphone senses the shake by the accelerometers. The FU system in the smartphone records the changes of the accelerations and maps them into three accelerations: total acceleration, horizontal acceleration, and vertical acceleration. In order to describe the intensity of the accelerations, FU generates a report of this sensing event as follows:

$$r_i = \left(l_i, t_i, a_i^t, a_i^h, a_i^v\right), \quad (1)$$

where l_i denotes the location of the sensing event e_i, which includes the latitude, longitude, and altitude of its center by GPS. t_i denotes the sensing time of this event e_i. a_i^t, a_i^h, and a_i^v denote the total acceleration, horizontal acceleration, and vertical acceleration, respectively. They contain the mean and the standard variation of the accelerations during the sensing window.

For the centralized data aggregation, there are two possible ways for data communication as shown in Figure 1: (1) the vehicles can immediately communicate the data via cellular network (3G/4G); otherwise, (2) the vehicles can offload the data via the WiFi APs. The problem of crowdsensing with cellular network is the increase in traffic demand and high energy consumption. Furthermore, the tasks of crowdsensing, which include gathering raw data and querying the services, can severely increase the overhead of the cloud, such as the computation, the storage, and the bandwidth. For vehicular users, mobile data offloading through WiFi has demonstrated its feasibility in reducing the data burden on the cellular networks. More recently, delayed offloading has been proposed [5]: if there is currently no WiFi availability, the traffic can be delayed instead of being sent/received immediately over the cellular interface. Although the delayed offloading can reduce the costs of mobile data communication, this approach obviously increases the data delivery delay.

Definition 1 (delivery delay of a single sensing data). We define the delivery delay of a single sensing data (denoted by d) in such crowdsensing-based system as the duration from the time when a probing vehicle senses the data and generates a report to the time when the feedback result from the SAME report is received by another querying vehicle.

Thus, in the cloud-based architecture, the data delivery includes three parts, which are the data sensing, the data

communication, and the data aggregation. The data delivery delay with remote cloud (denoted by D_{RC}) via immediate cellular network can be calculated as follows:

$$D_{RC} = D_{sen} + D_{cell}^{up} + D_{agg} + D_{cell}^{down}, \quad (2)$$

where D_{sen} denotes the delay of sensing data, which includes the delays of sensing the event and generating its report. D_{agg} denotes the delay of data aggregation in the remote cloud. D_{cell}^{up} and D_{cell}^{down} denote the delay of uploading to the remote cloud via cellular network and that of downloading from the remote cloud via cellular network, respectively.

By way of delayed data offloading, the vehicle will carry the sensing data to move until it meets with an AP. The data delivery delay by mobile data offloading can be calculated as follows:

$$D_{RC} = D_{sen} + D_{carry} + D_{AP}^{up} + D_{agg} + D_{AP}^{down} + D_{wait}, \quad (3)$$

where D_{carry} denotes the delay of carrying the sensing data to an AP and D_{wait} denotes the delay of waiting for a querying user to download the feedback result from an AP. D_{AP}^{up} and D_{AP}^{down} denote the delay of uploading to the remote cloud via AP and that of downloading from the remote cloud via AP, respectively. Due to the delays of D_{carry} and D_{wait}, the data delivery delay via mobile data offloading is much longer than that via the cellular network.

3.2. Local Crowd. Distinct from the centralized data aggregation, the authors in [16] introduce a distributed road information sharing architecture with rumor and report. The basic idea behind this mechanism is that each vehicle that hears a rumor about the event maintains a time decaying belief about it. Rumors from multiple vehicles are combined additively until they exceed a prescribed threshold, at which point they are converted to confirm event reports. This threshold is committed to effect a desired trade-off in information reliability, between the rate of false negatives and the rate of false positives. Both rumors and reports are distributed through the network in an epidemic gossip spread fashion. As time goes, the belief value of the rumor is changed following a predetermined *decay function* in order to discount the aged information. Although the decay function may be any nonincreasing function of the elapsed time from the creation of the rumor, we focus on the exponentially decreasing function.

In this paper, we investigate the distributed approach for the crowdsensing data. We suppose that the vehicles can communicate with each other by vehicle-to-vehicle (V2V) communication, such as vehicular ad hoc networks (VANET) [17] or Device-to-Device (D2D) communication [18]. The sensing data are aggregated by the vehicles near the related event. We define these vehicles near the event in a predefined range (denoted by r) as the *local crowd* of this event. Consider

$$c_i = \{v \mid \|v.l - e.l\| \le r,\ v \in \mathscr{V}\}, \quad (4)$$

where $v.l$ and $e.l$ denote the location of the vehicle and the event, respectively. $\mathscr{V}$ denotes the set of the vehicles. The size

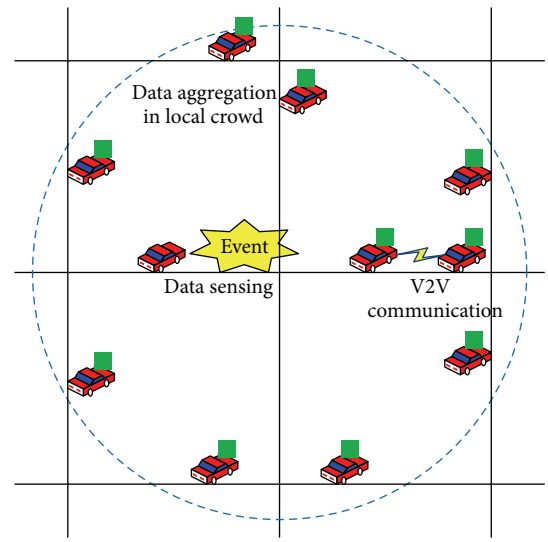


Figure 3: Crowdsensing data with local crowd.

of the local crowd is related to the requirement of notification for the feedback results.

As shown in Figure 3, a moving vehicle senses an event on the road, and then it generates a report to describe the event. When the vehicle meets another one in the local crowd of this event, it will send a copy of the report to another one. Thus, all the reports of this event will be disseminated to all the vehicles in the local crowd of this event by the way of epidemic routing [19]. When a vehicle moves out of the crowd, it will handle all the reports of the event in two different ways, according to the traffic density: (1) it will drop all the reports of the event under high traffic density; (2) under low traffic density, it cannot transmit the reports of the event with other vehicles, but the reports in its buffer will not be dropped. The vehicles in the crowd maintain the crowdsensing data from the event and calculate the feedback results by the algorithm of truth discovery. When a querying vehicle moves into this local crowd and meets another vehicle, it will receive a feedback result of this event. Therefore, the crowdsensing data are communicated and shared in this local crowd.

In such crowd-based architecture, the data delivery delay with local crowd (denoted by D_{LC}) can be calculated as follows:

$$D_{LC} = D_{sen} + D_{diss} + D_{agg} + D_{back}, \quad (5)$$

where D_{sen} denotes the delay of sensing data and D_{agg} denotes the delay of data aggregation in the vehicles of the local crowd. D_{diss} denotes the average delay of disseminating the reports to the vehicles in the crowd. D_{back} denotes the delay from the time when the feedback result is calculated to the time when the querying vehicle receives the result.

3.3. Comparison. We have discussed three different approaches for the crowdsensing-based system: (1) to immediately communicate with remote cloud via cellular network (such as 3G/4G); (2) to delay offloading to the remote cloud

Table 1: Comparison among the three approaches.

	Delay	Cost	Energy	Quality
3G/4G to remote cloud	Low	High	High	High
Offload to remote cloud	High	Low	Low	High
V2V to local crowd	Median	Low	Low	Median

via roadside APs; (3) to share the crowdsensing data in a local crowd by V2V communications. Table 1 shows the comparison among the three approaches for crowdsensing data by four different metrics, which are data delivery delay, cost, energy consumption, and quality of truth discovery.

3.3.1. 3G/4G to Remote Cloud. The crowdsensing data are aggregated by the back-end server (or cloud) and are immediately communicated by the cellular networks (3G/4G). The data delivery delay of this approach is the lowest. It consumes the highest cost and energy for the users, which has been discussed in [5]. The back-end server can harvest all the crowdsensing data, so the quality of the feedback results is high.

3.3.2. Offload to Remote Cloud. Due to the sparse distribution of the access points, the crowdsensing data is delayed in offloading to the back-end server (or cloud). Thus, the data delivery delay of this approach is the highest. It consumes the lowest cost and energy for the users, which has been discussed in [5]. Like the 3G/4G-based approach, the quality of the feedback results is high.

3.3.3. V2V to Local Crowd. The crowdsensing data are communicated and shared in this local crowd by V2V communication, such as MANET or Device-to-Device (D2D) communication. The data delivery delay is less than that of offloading-based approach under a high traffic density, which has also been discussed in [13]. It consumes much less cost and energy than 3G/4G-based approach. The local crowd may harvest part of the crowdsensing data; the quality of the feedback results is not higher than 3G/4G-based and offloading-based approaches, and we will discuss it in the next section.

4. Truth Discovery in Local Crowd

In this section, we analyze the performance of the scheme with the local crowd. First, we discuss the trade-off between the quality of the truth discovery and the user satisfaction in the local crowd. Then, we define a utility function related to the crowdsensing data shared by the local crowd, by considering both of the quality of the truth discovery and the user satisfaction. Compared with the cloud-based approach, we model the quality of the truth discovery by Kullback-Leibler divergence (or relative entropy). Last, we formulate the sharing crowdsensing data in local crowd as an optimization problem. The notations used in this paper are given in Notations Section.

4.1. Trade-Off between the Quality and the Satisfaction. As our discussion in the previous section, in such crowdsensing-based system, when a vehicle moves through an object (e.g., an obstacle) or an event (e.g., an accident), the smartphone in it can sense it. Then, the mobile device generates a report

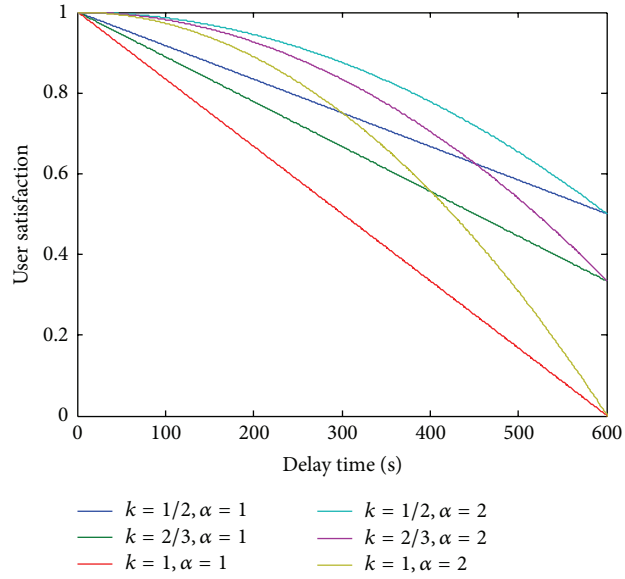


Figure 4: User satisfaction.

to describe the anomaly and disseminates the report to the vehicles in the local crowd. Thus, all the reports of the event will be shared by all the vehicles in the local crowd.

With the increase of time, the vehicles in the local crowd will receive more and more reports about the event, so the quality of the truth discovery is getting better. Meanwhile, the mobile users' satisfaction will be reduced. Thus, in the crowdsensing data shared by the local crowd, there is a trade-off between the quality of the truth discovery and the user satisfaction.

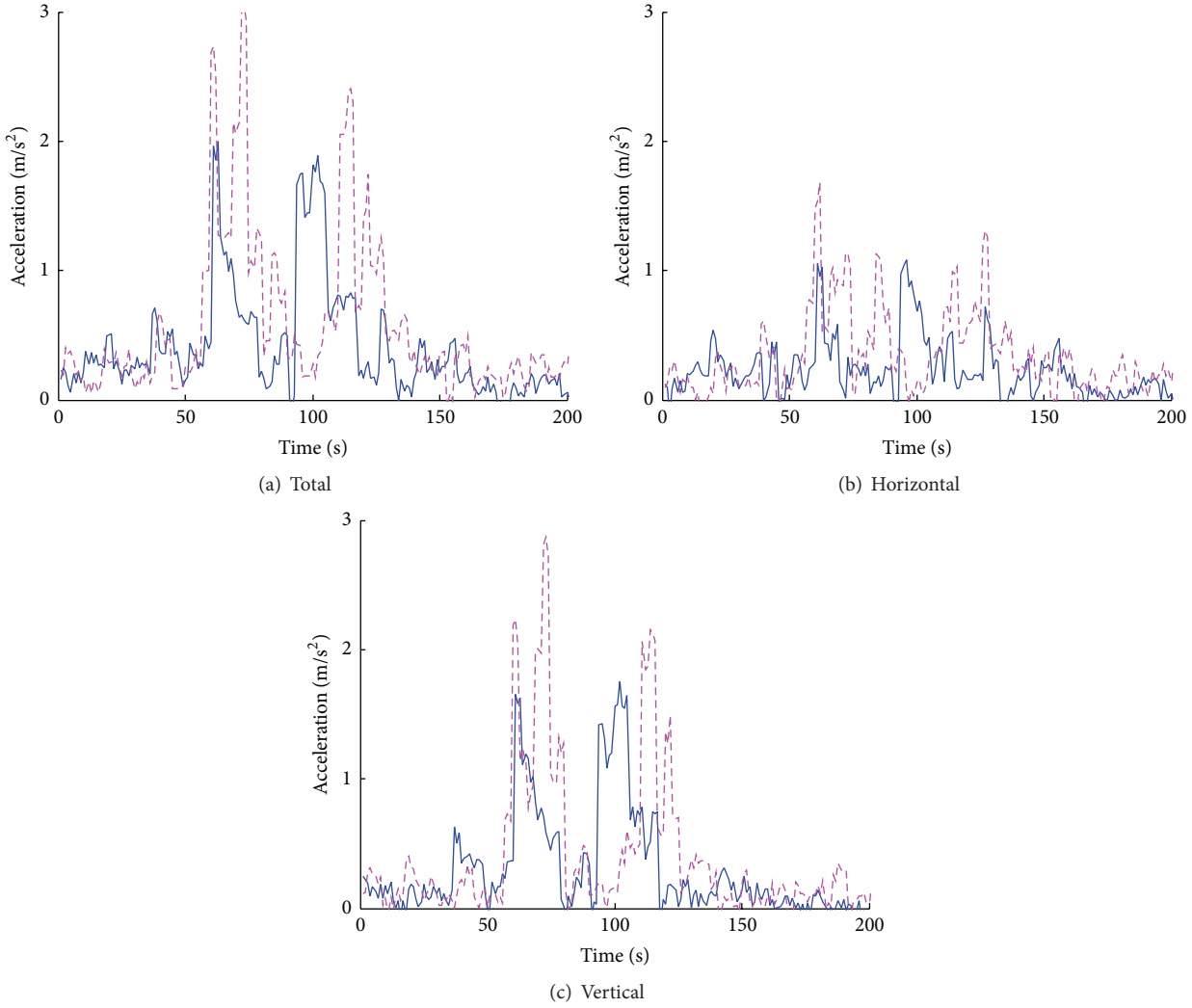
Definition 2 (feedback delay for a querying user). We define the feedback delay for a querying user (denoted by t) in such crowdsensing-based system as the duration from the time when the first probing user senses the event and generates a report, to the time when any feedback results are received by the querying users.

Here, we consider two metrics to evaluate the performance of the scheme with the local crowd, which are the quality of the truth discovery and the user satisfaction. Inspired by the definition of the utility function in [12], we define a utility function related to the crowdsensing data shared by the local crowd, in order to quantify the trade-off between the quality of the truth discovery and the user satisfaction. The utility function $U(t)$ of delay period t starting from the time of event detection by the probing user ($t = 0$) until the time of the feedback to the querying user is as follows:

$$U(t) = Q(t) \cdot S(t), \tag{6}$$

where $Q(t)$ denotes the quality of the truth discovery by the local crowd, compared with the cloud-based approach. $S(t)$ is a function measuring the degree of user satisfaction based on the delay time t.

Thus, the higher quality of the truth discovery and the higher user satisfaction can increase the utility of crowdsensing-based system with local crowd. Next, we will discuss



(a) Total (b) Horizontal

(c) Vertical

Figure 5: Compare the sensing data for the same event.

our models of quality of the truth discovery and user satisfaction, respectively.

4.2. User Satisfaction. With the increase of delay, the mobile users become impatient and hence their satisfaction will be greatly reduced [5]. In (6), $S(t)$ is a general definition of the user's perception of the delay of crowdsensing. We assume that the querying user will usually become more and more impatient while waiting for these feedback results. For simplicity without losing generality, we define the satisfaction function as follows:

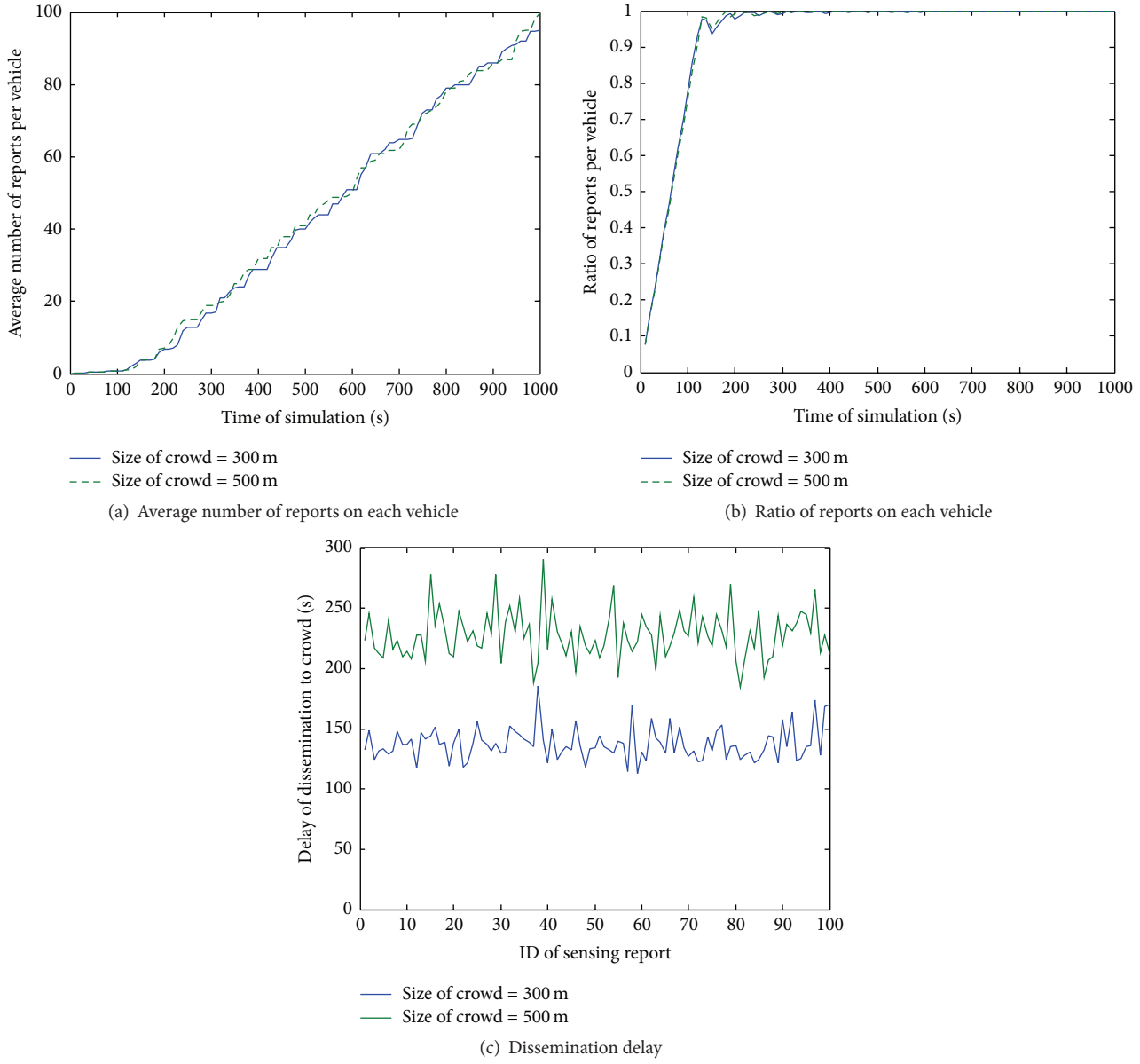
$$S(t) = \begin{cases} 1 - k \cdot \left(\dfrac{t}{T_{\max}}\right)^{\alpha}, & t \le T_{\max}, \\ 0, & t > T_{\max}, \end{cases} \tag{7}$$

where $T_{\max}$ is the maximum delay tolerance of the user with respect to the requested content and is used to normalize the function. k is the decreasing rate of user satisfaction as time elapses, to avoid a zero utility value at $t = T_{\max}$, and α is a decay factor for the delay time t.

Hence, the function $S(t)$ for the user would be a monotonically decreasing function between 0 and 1, after $T_{\max}$, meaning that the reception afterwards is useless. Figure 4 shows the user satisfaction as a function of the delay time, where the parameter $T_{\max}$ is set as 10 minutes. In this paper, the parameter k is set as $1/2$, and α is set as 1.

4.3. Quality of Truth Discovery. The quality of the truth discovery is related to the number of crowdsensing data. We take the centralized approach as the benchmark, because the back-end server can harvest all the crowdsensing data. Thus, the quality of the truth discovery by the local crowd is evaluated by the comparison with the centralized approach.

However, the crowdsensing data are stochastic. The same vehicle meets the same event at different times; the sensing data are different. We do an experiment where the identical vehicle moves through the same dump twice and record the changes of its accelerations. Figure 5 shows the results to compare two tests, which include the total acceleration,



(a) Average number of reports on each vehicle

(b) Ratio of reports on each vehicle

(c) Dissemination delay

FIGURE 6: Dissemination of the crowdsensing data in a local crowd.

horizontal acceleration, and vertical acceleration. We notice that the results of the two tests are different. The uncertainty of the crowdsensing data could be affected by many factors, such as the mobile devices, the drivers, the vehicle, and the environment.

Due to the uncertainty of the crowdsensing data, we utilize the Kullback-Leibler divergence (or relative entropy) to evaluate the quality of the local crowd. In probability theory and information theory, the Kullback-Leibler divergence (also information divergence, relative entropy, or KL divergence) is a nonsymmetric measure of the difference between two probability distributions [20].

Let $R(t)$ denote the discrete probability distributions of the crowdsensing data in the remote cloud at time t, and let $L(t)$ denote the discrete probability distributions of the crowdsensing data in the local crowd at time t. Thus, the quality of the truth discovery by the local crowd is evaluated by the Kullback-Leibler divergence of R from L as follows:

$$\begin{aligned} Q(t) &= Q_{\max} - D_{\mathrm{KL}}(L(t) \parallel R(t)) \\ &= Q_{\max} - \sum_{i} L(t,i) \ln \frac{L(t,i)}{R(t,i)}, \end{aligned} \tag{8}$$

where $Q_{\max}$ denotes the maximal quality. D_{KL} denotes the function of the Kullback-Leibler divergence, and it is equal to 0 when L and R have the same distribution. $L(t,i)$ denotes the probability of the crowdsensing data by the local crowd during the period t, whose value is equal to i. Likewise, $R(t,i)$ denotes the probability of the crowdsensing data by the remote cloud during the period t, whose value is equal to i. Thus, with the growth of time, the vehicles in the local crowd

Table 2: Parameters of simulation.

Parameter	Value
Size of scenario	4000 m × 3000 m
Traffic density (k)	50 or 10 vehicles per km
Average speed of the vehicles (v)	40 km/h
Size of the local crowd	500 m
Transmission range of V2V communication	50 m

will receive more and more reports about the event, so the quality of the truth discovery is getting better.

We use our customized simulator to evaluate the performance of communication and sharing of the crowdsensing data with the local crowd. The scenario of our simulation is 4000 m × 3000 m. The traffic density (k) is defined as the number of vehicles per unit length of the roadway. The average speed of the vehicles is 40 km/h. The transmission range of V2V communication is 50 m. All the parameters used in our simulation are listed in Table 2.

We simulate the dissemination of the crowdsensing data in the local crowd. During the period of the simulation, more and more vehicles sense the event and disseminate the reports to the crowd. Thus, the vehicles in the crowd will receive more and more reports, to improve the quality of the truth discovery. Figure 6(a) shows the average number of the reports received by the vehicles in the crowd. We notice that the number of the reports received by the vehicles in the crowd is increasing, because more vehicles sense the event and disseminate to the crowd. The dissemination under the smaller size of the crowd is faster than that under the bigger crowd. Figure 6(b) indicates the ratio between the average number of reports received by each vehicle and the total number of the reports. We find that the ratio is sharply increasing before about 150 seconds, and then remains at about 1.

When a probing vehicle meets and senses an abnormal event, it will generate a sensing report and disseminate it to all the vehicles in the crowd of this event. Thus, we evaluate the delay of this dissemination from the first vehicle to all the vehicles in the crowd. Figure 6(c) shows the delay of the dissemination to the crowd. The IDs of the sensing reports denote the different reports generated by separate vehicles. Due to the mobility of the vehicles, the delays of the different reports are varied. We notice that the smaller size of the crowd (300 m) has lower delay than the bigger size (500 m).

4.4. Optimization Problem. The feedback delay for a querying user (t) should be satisfied by the condition that the utility is no less than the predefined threshold δ, as follows:

$$U(t) \geq \delta. \tag{9}$$

Depending on the demand of the quality (denoted by Q_D), we formulate the optimal feedback time for a querying user (t) with the maximal utility as an optimization problem as follows:

$$\begin{aligned} \max_{t} \quad & U(t) = Q(t) \cdot S(t) \\ \text{s.t.} \quad & t \leq T_{\max} \\ & Q(t) \geq Q_D \\ & U(t) \geq \delta. \end{aligned} \tag{10}$$

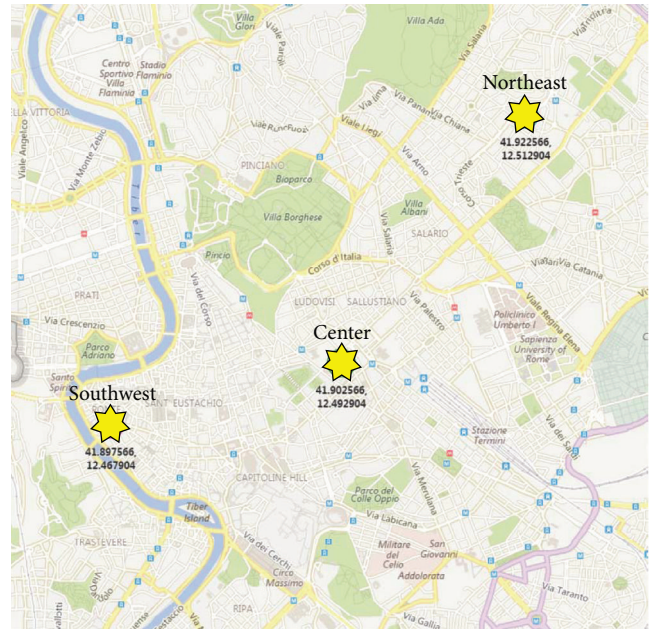


Figure 7: The scenario of Roma taxi.

In such optimization problem, $Q(t)$ can be estimated by the historical statistics with the help of (8). The system harvests the local crowd information and the remote cloud information and calculates $Q(t)$ as the historical records. For a new local crowd, the system chooses the record which is the geographically nearest during the same period of a day as a reference to estimate its $Q(t)$. $S(t)$ is a linear function of the time t. By considering the requirements on the data delivery delay ($T_{\max}$) and the quality of the truth discovery (Q_D), we can find the optimal time (t) with the maximal utility, which will be further discussed in the next section.

5. Simulation Results and Discussions

In this section, we evaluate the performance of the local crowd. Our experiments are based on the dataset consisting of real vehicular traces. We evaluate the number of vehicles in the local crowd, the number of sensing reports, the ratio of the vehicles with reports, and the average delay of the reports. All the parameters are listed in Table 3.

5.1. Taxi-ROMA Dataset. When a vehicle moves into a local crowd, it will join in this crowd. Oppositely, when a vehicle moves out of a local crowd, it will leave this crowd. We do experiments on the Taxi-ROMA dataset [21]. This dataset contains real mobility traces of taxi cabs in Rome, Italy. It contains GPS coordinates of approximately 320 taxis collected over 30 days. We select the dataset of the traces collected on February 5, 2014, which contains 172 taxis. The traces cover the area with the range of 66 km × 59 km.

As shown in Figure 7, we set three events happening at different places with the yellow marks, which are termed as the northeast event, the southwest event, and the center event.

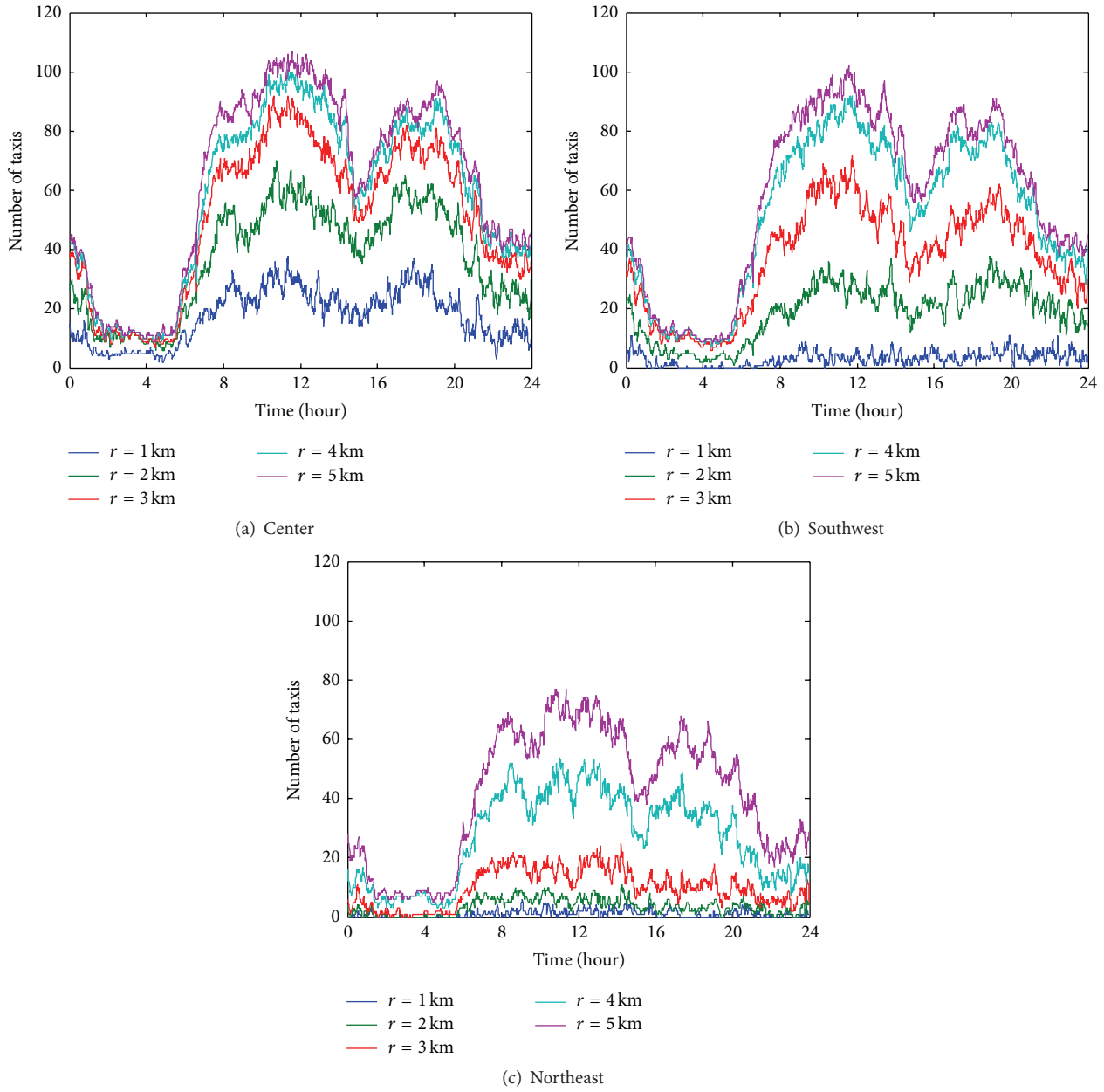


Figure 8: Number of the vehicles in a local crowd with different predefined ranges.

Table 3: Parameters of Roma taxi.

Parameter	Value
Day of the dataset	February 5, 2014
Number of the taxis	172
Range of the area (km^2)	66×59
Number of the events	3
Communication range (m)	300
Sensing range (m)	200
Range of the local crowd (km)	1, 2, 3, 4, and 5

The traffic density near the center event is the highest and that near the northeast event is the lowest.

The communication range of each vehicle is 300 m. In our experiments, the vehicles can communicate with each other or sense the events, only in the local crowd of each event. When the vehicle moves out of the crowd, it cannot transmit the reports of the event with other vehicles, but the reports in its buffer will not be dropped. The sensing range of an event is defined as the range from this event that a vehicle can sense it. In our experiment, we set the sensing range as 200 m.

5.2. Number of Vehicles in the Local Crowd. We evaluate the number of the vehicles in a local crowd with different predefined ranges (r) from 1 km to 5 km as a function of time during a day. Figure 8 shows the results of the events

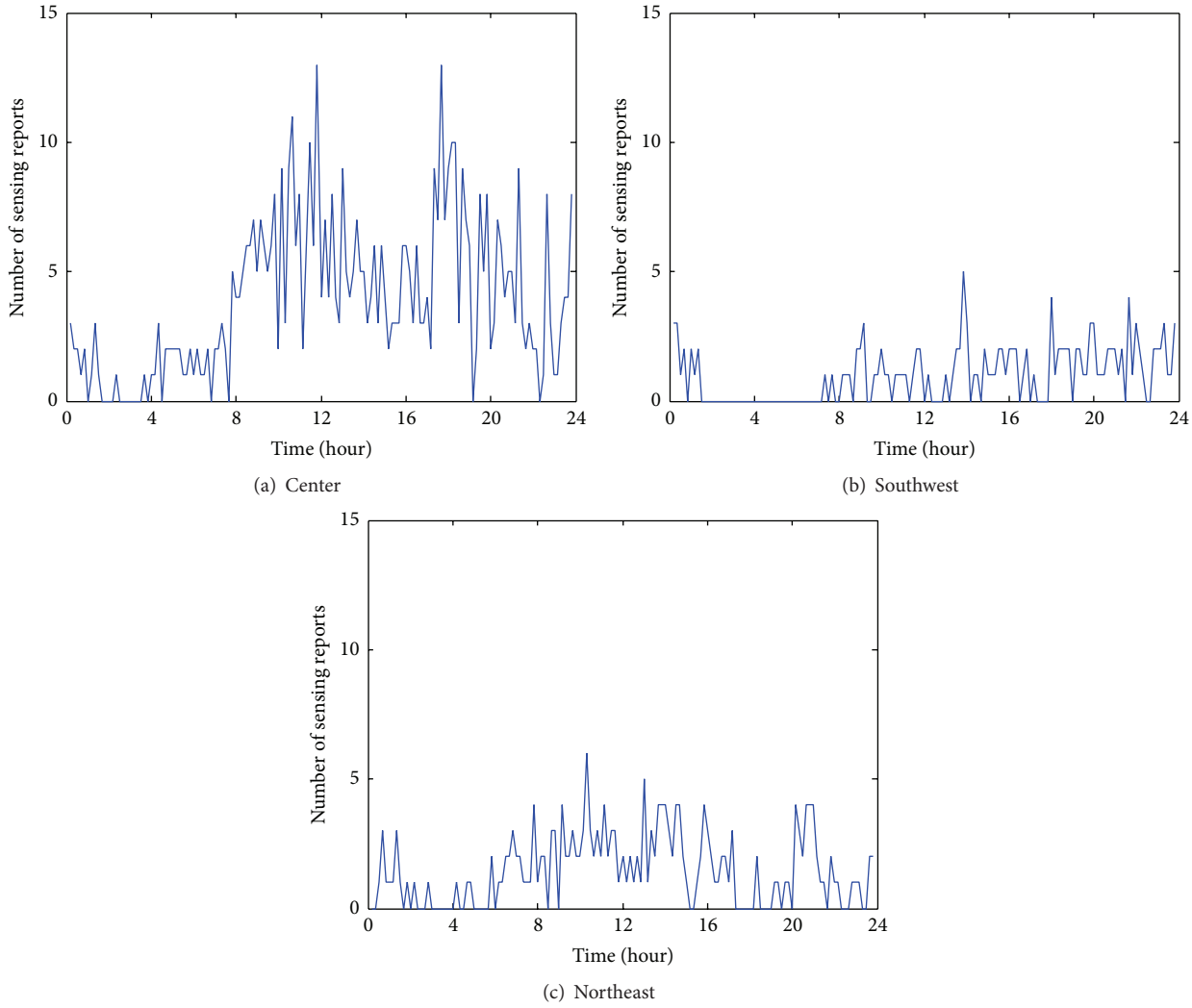


Figure 9: Number of the sensing reports in a local crowd.

which happen in different places. The size of the window for sampling is 60 seconds.

We notice that the number of vehicles in the local crowd of the center event is the largest, due to the high traffic density. In contrast, the number of vehicles in the local crowd of the northeast event is the smallest, due to the low traffic density.

While the range of the local crowd is increasing, the number of vehicles in the local crowd is also increasing. The local crowd with the range of 5 km has the largest number of vehicles for each event at different places.

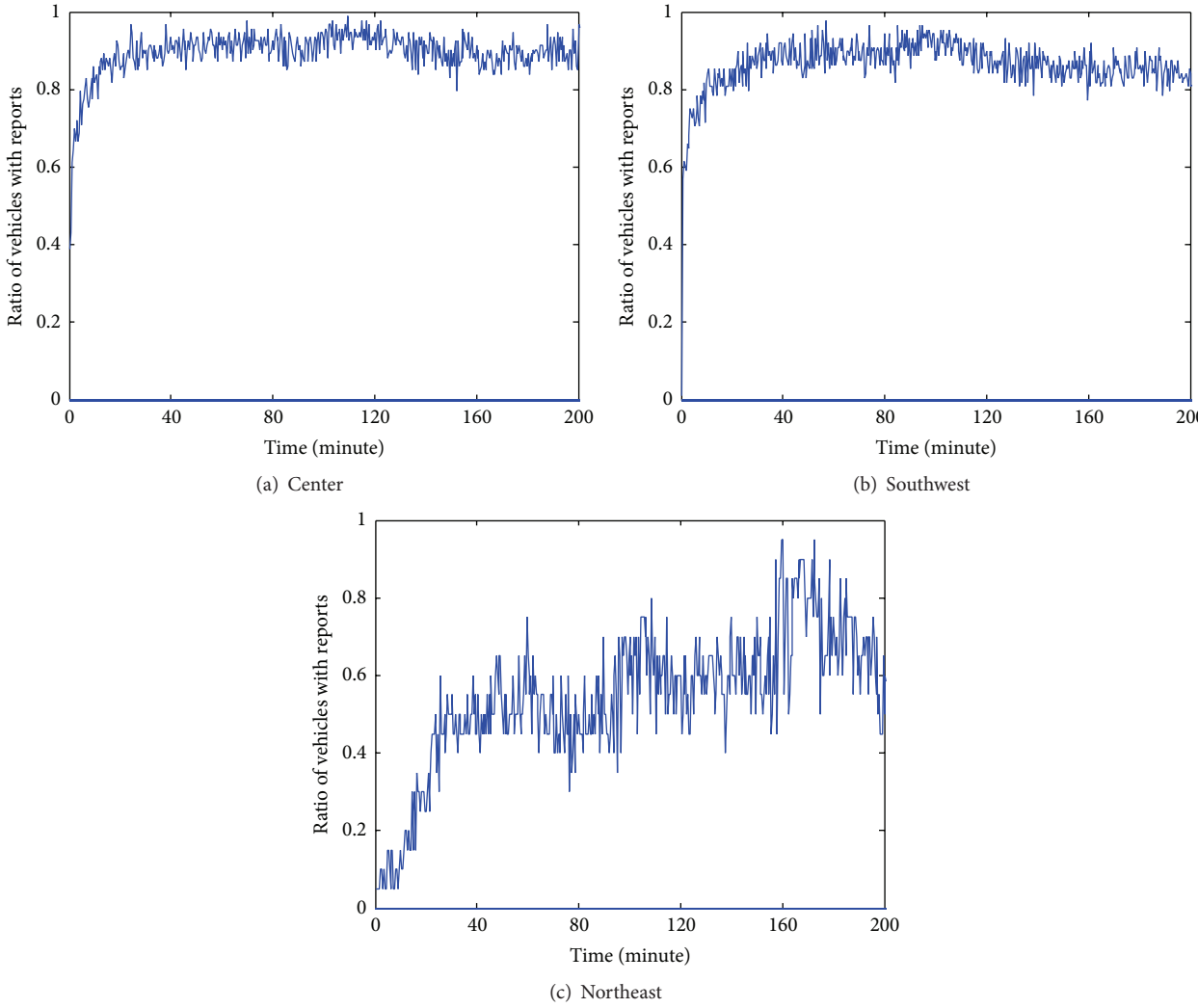
The number of vehicles in the local crowd is changed at different times during the whole day. In particular, the number of vehicles between 2 am and 5 am is the smallest and that between 10 am and 12 am is the largest. That is also caused by the traffic density at different times.

5.3. Number of the Sensing Reports. When a vehicle moves into the sensing range of an event, it will generate a report for it. We evaluate the number of the sensing reports in a local crowd as a function of time during a day. Figure 9 shows the results of the events which happen in different places. The range of the local crowd is set as 3 km.

We notice that the number of the sensing reports from the center event is the largest, due to the high traffic density. In contrast, the number of the sensing reports from the northeast event is the smallest, due to the low traffic density.

The number of the sensing reports is changed at different times during the whole day. In particular, the number of the sensing reports from the center event between 2 am and 4 am is the smallest. The number of the sensing reports from the southwest event between 2 am and 7 am is the smallest. The number of the sensing reports from the northeast event between 2 am and 5 am is the smallest.

The numbers of the sensing reports from the center event at about 12 am and 5 pm are the largest. The number of the sensing reports from the southwest event at about 1p is the largest. The number of the sensing reports from the northeast event at about 10 am is the largest. That is also caused by the traffic density at different times.



(a) Center

(b) Southwest

(c) Northeast

Figure 10: Ratio of the vehicles with the reports.

5.4. Ratio of the Vehicles with Reports. After generating a report by a vehicle, this report will be disseminated in the local crowd of this event. The ratio of the vehicles with the reports is defined as the ratio of the number of vehicles which has the reports to that of vehicles in the local crowd. We evaluate the changes of the ratio of the vehicles with the reports during 200 minutes after the first report is generated. Figure 10 shows the results of the events which happen in different places. The range of the local crowd is set as 3 km. Initially, the ratio is the lowest, since only the first vehicle in the local crowd has the report. We notice that with the increasing time, the ratio is also increasing, because more and more vehicles will receive the reports. The local crowd of the center event has the highest ratio, due to the high traffic density. In contrast, the local crowd of the northeast event has the lowest ratio, due to the low traffic density. Because there are some new vehicles without any reports moving into the crowd, the ratio cannot reach 1.

5.5. Quality and Utility of Local Crowd. The quality of the truth discovery at local crowd is evaluated by (8), which compares the sensing data harvested by the local crowd with those harvested by the remote cloud. We evaluate the quality of the local crowd in different places, and the results are shown in Figure 11. The parameter $Q_{\max}$ is set by 1. The range of the local crowd is set as 3 km. As in the aforementioned introduction, the sensing data from the same event are stochastic. In this simulation, the sensing data from the event follow a Poisson distribution. At the beginning, the quality is the lowest, since the vehicles in the local crowd have few sensing reports. We notice that with the increasing time the quality is also increasing, because the vehicles will receive more and more sensing reports. Finally, the quality will be approximated to $Q_{\max}$. Among the three places, the local crowd of the northeast event has the lowest quality, due to the low traffic density. Moreover, we notice that, for center and southwest regions, the quality of the local crowd is reaching $Q_{\max}$ after about 20 minutes. In Figure 10, we also notice that, after about 20 minutes, the ratio of the vehicles with the reports for center and southwest regions is about 90%. This is because when the average number of the received reports in the local crowd is close to the total number at remote cloud,

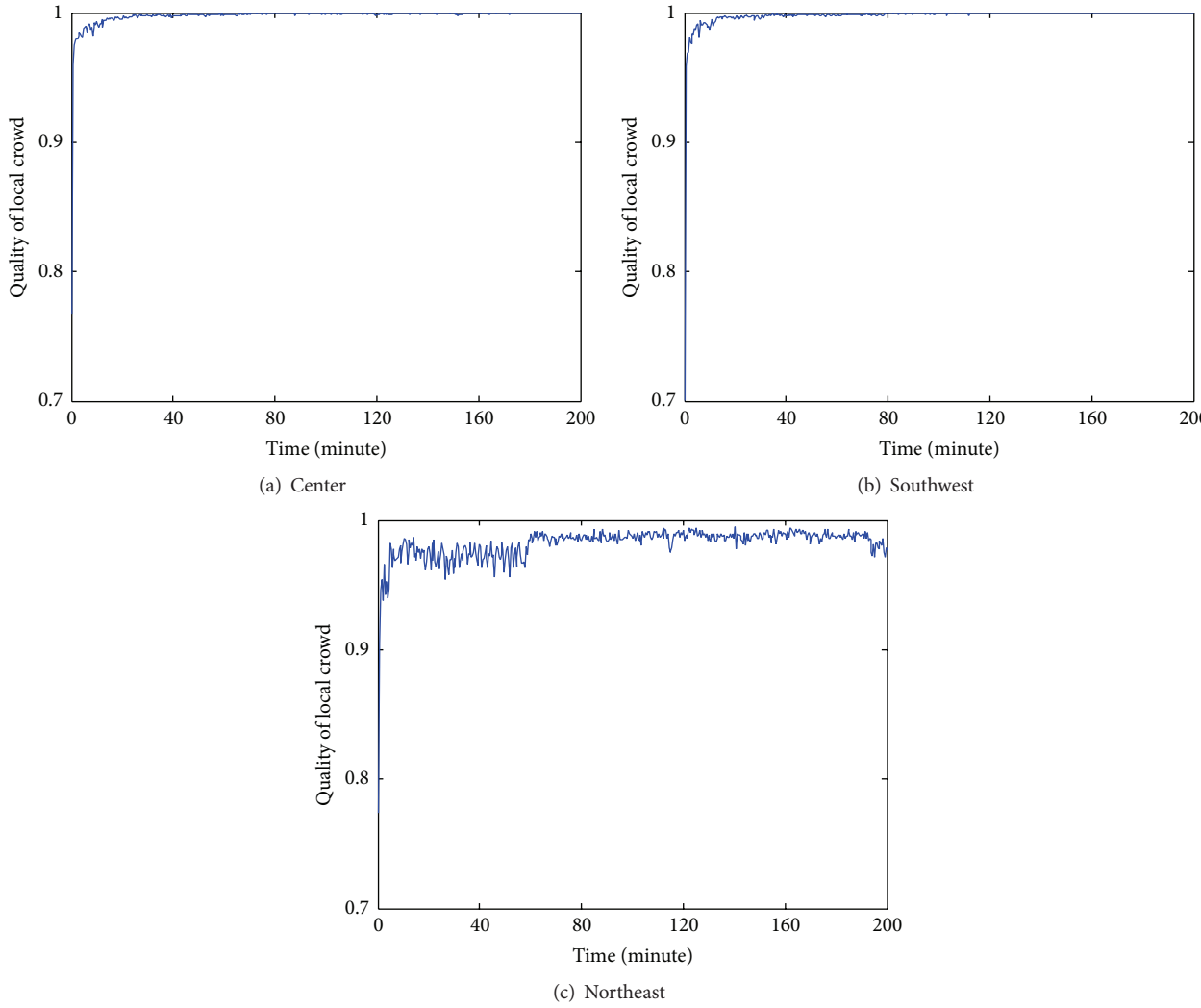


Figure 11: Quality of the local crowd.

there is much less difference between the performances of the local crowd and the remote cloud.

With the help of (10), we can find the optimal time with the maximal utility by considering the requirements on the data delivery delay and the quality of the truth discovery. We evaluate the utility of the local crowd in different places during the period of 10 minutes ($T_{\max}$) from the beginning, and the results are shown in Figure 12. The range of the local crowd is set as 3 km. Among the three places, the local crowd of the northeast event has the lowest utility, due to the low traffic density. We notice that when the time is equal to 2 minutes, the query user has the maximal utility.

5.6. Average Delay of Disseminating the Reports. We define the delay of disseminating a report from the time when the report is generated to the time when its copy is received by another vehicle. We evaluate the average delays of disseminating the reports from the events at different places as a function of different ranges of the local crowds from 1 km to 5 km. The lifetime of the report is 800 seconds, so the report will be dropped by the mobile device after that time. Figure 13 shows the average delays of disseminating the reports from the events which happen in different places.

We find that the average delay of disseminating the reports from the center event is the lowest, due to the high traffic density. In contrast, the average delay of disseminating the reports from the northeast event is the highest, due to the low traffic density.

However, in the local crowd with the shortest range, the average delay is high, because the number of the vehicles is not enough to disseminate the reports. In the local crowd with the longest range, the average delay is also high, because of the long distance among the vehicles.

6. Conclusion

In mobile crowdsensing system, the cloud (or back-end server) harvests the crowdsensing data from the mobile devices and then aggregates the data to the feedback results for the querying users. Offloading the crowdsensing data

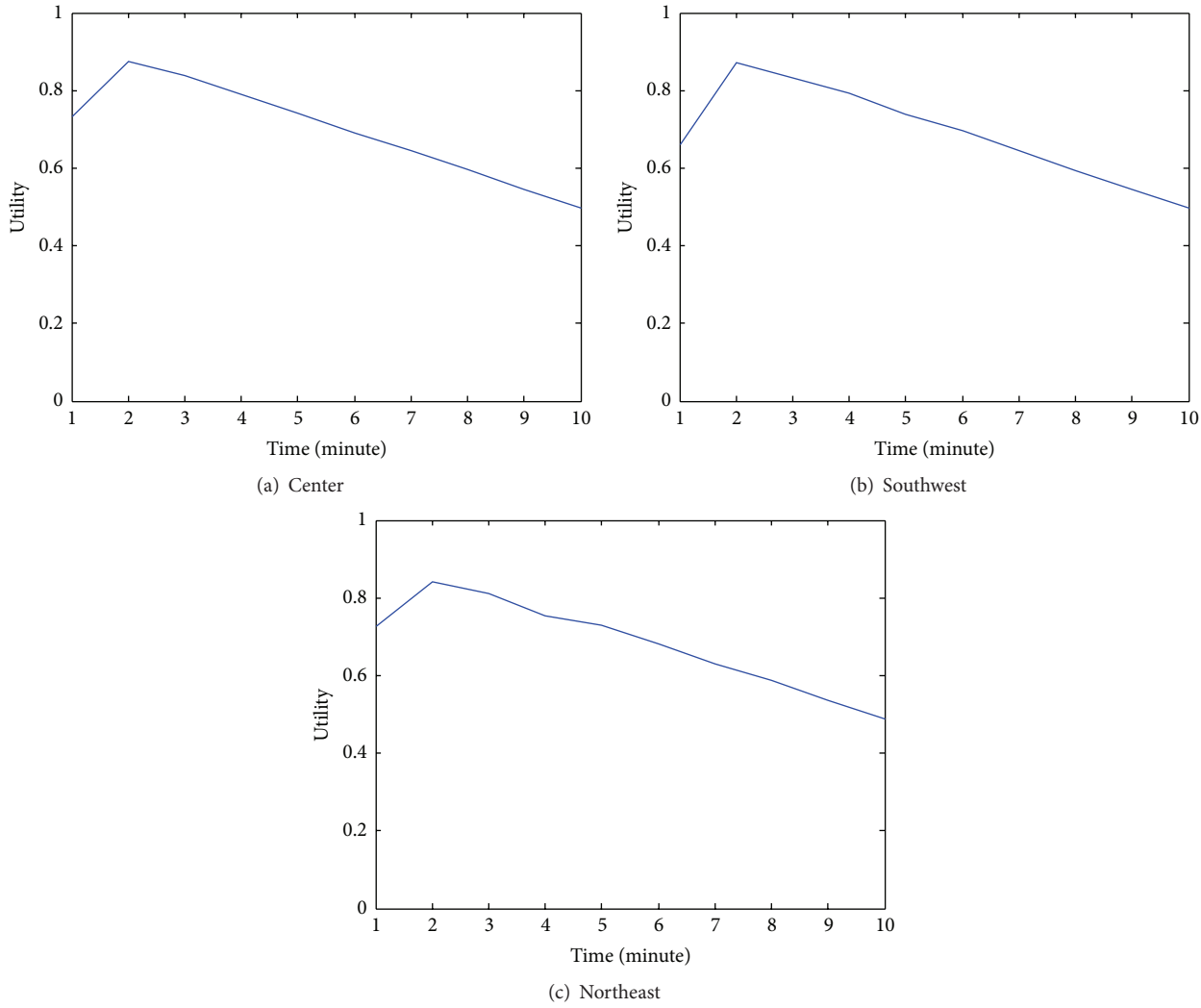


Figure 12: Utility of the local crowd.

to the cloud has less cost and energy consumption than the way of 3G/4G, but it has longer data delivery delay. Compared with offloading data via WiFi APs, we investigate the communication and sharing of crowdsensing data by vehicles near the sensing event, termed as a local crowd. The crowd-based approach has a lower delay than the offloading-based approach, by considering the quality of truth discovery. We define a utility function related to the crowdsensing data shared by the local crowd, in order to quantify the trade-off between the quality of the truth discovery and the user satisfaction. Our extensional simulations verify the effectiveness of our proposed schemes.

Notations

r: Predefined range of the local crowd
d: Delivery delay of a single sensing data
t: Feedback delay for a querying user
$U(t)$: Utility of the crowdsensing data in the local crowd at time t

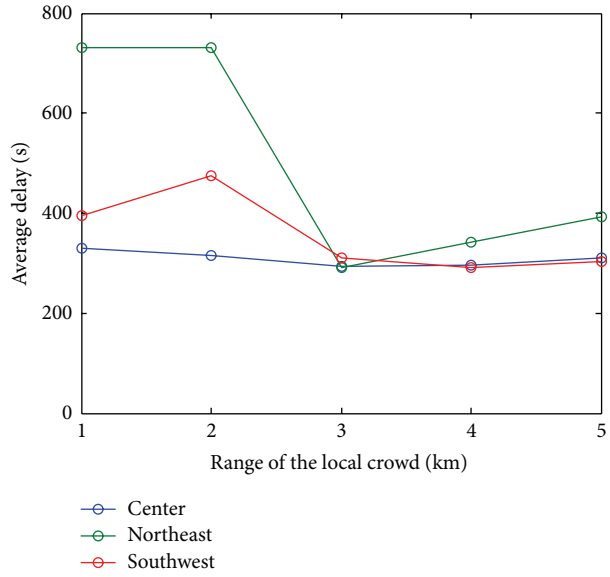


Figure 13: Average delay in the local crowd.

$S(t)$: Satisfaction of feedback from the local crowd at time t
$Q(t)$: Quality of the feedback results from the local crowd at time t
$T_{\max}$: The maximum delay tolerance of the user
$R(t)$: The discrete probability distributions of the crowdsensing data in the remote cloud at time t
$L(t)$: The discrete probability distributions of the crowdsensing data in the local crowd at time t
δ: Threshold of the utility
Q_D: Demand of the quality of the truth discovery.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by NSFC Grants nos. 61170256, 61103226, 61173172, 61272526, and 61370204, the Fundamental Research Funds for the Central Universities (nos. ZYGX2013J077 and ZYGX2013J067), and the Applied Basic Program of Sichuan Province of China (no. 2014JY0192).

References

[1] "List of countries by number of mobile phones in use," http://en.wikipedia.org/wiki/.

[2] E. Miluzzo, N. D. Lane, K. Fodor et al., "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)*, pp. 337–350, November 2008.

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (ACM MobiSys '08)*, pp. 29–39, Breckenridge, Colo, USA, June 2008.

[4] R. W. Ouyang, M. Srivastava, A. Toniolo, and T. J. Norman, "Truth discovery in crowdsourced detection of spatial events," in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM '14)*, pp. 461–470, ACM, Shanghai, China, November 2014.

[5] F. Mehmeti and T. Spyropoulos, "Is it worth to be patient? Analysis and optimization of delayed mobile data offloading," in *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM '14)*, pp. 2364–2372, Toronto, Canada, April 2014.

[6] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 269–280, Istanbul, Turkey, August 2012.

[7] P. Zhou, Y. Zheng, and M. Li, "Demo: how long to wait?: predicting bus arrival time with mobile phone based participatory sensing," in *Proceedings of the ACM 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, pp. 459–460, 2012.

[8] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 173–184, Istanbul, Turkey, August 2012.

[9] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *Proceedings of the 33rd IEEE Conference on Computer Communications (IEEE INFOCOM '14)*, pp. 745–753, Ontario, Canada, May 2014.

[10] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," *Proceedings of the ACM 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*, pp. 209–222, 2010.

[11] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: how much can wifi deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, 2013.

[12] D. Zhang and C. K. Yeo, "Optimal handing-back point in mobile data offloading," in *Proceedings of the IEEE Vehicular Networking Conference (VNC '12)*, pp. 219–225, Seoul, South Korea, November 2012.

[13] N. Ristanovic, J.-Y. Le Boudec, A. Chaintreau, and V. Erramilli, "Energy efficient offloading of 3G networks," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '11)*, pp. 202–211, Valencia, Spain, October 2011.

[14] T. Luo and C.-K. Tham, "Fairness and social welfare in incentivizing participatory sensing," in *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '12)*, pp. 425–433, Seoul, Republic of Korea, June 2012.

[15] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM '12)*, pp. 2140–2148, Orlando, Fla, USA, March 2012.

[16] J. Ahn, Y. Wang, B. Yu, F. Bai, and B. Krishnamachari, "RISA: distributed road information sharing architecture," in *Proceedings of the IEEE INFOCOM*, pp. 1494–1502, Orlando, Fla, USA, March 2012.

[17] J. Zhao and G. Cao, "Vadd: vehicle-assisted data delivery in vehicular ad hoc networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (IEEE INFOCOM '06)*, Barcelona, Spain, April 2006.

[18] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[19] A. Vahdat and D. Becker, *Epidemic routing for partially-connected ad hoc networks [M.S. thesis]*, 2000.

[20] S. Kullback, "Information theory and statistics," *American Mathematical Monthly*, vol. 504, no. 3, p. 301, 1968.

[21] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD data set roma/taxi (v. 2014-07-17)," July 2014, http://crawdad.org/roma/taxi/.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 8793025, 14 pages
http://dx.doi.org/10.1155/2016/8793025

Research Article

Privacy Leakage in Mobile Sensing: Your Unlock Passwords Can Be Leaked through Wireless Hotspot Functionality

Jie Zhang,[1] Xiaolong Zheng,[2] Zhanyong Tang,[1] Tianzhang Xing,[1] Xiaojiang Chen,[1] Dingyi Fang,[1] Rong Li,[1] Xiaoqing Gong,[1] and Feng Chen[1]

[1]*School of Information Science and Technology, Northwest University, Xi'an 710127, China*
[2]*School of Software and TNLIST, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Zhanyong Tang; zytang@nwu.edu.cn

Received 11 December 2015; Accepted 6 March 2016

Academic Editor: Tingting Chen

Mobile sensing has become a new style of applications and most of the smart devices are equipped with varieties of sensors or functionalities to enhance sensing capabilities. Current sensing systems concentrate on how to enhance sensing capabilities; however, the sensors or functionalities may lead to the leakage of users' privacy. In this paper, we present WiPass, a way to leverage the wireless hotspot functionality on the smart devices to snoop the unlock passwords/patterns without the support of additional hardware. The attacker can "see" your unlock passwords/patterns even one meter away. WiPass leverages the impacts of finger motions on the wireless signals during the unlocking period to analyze the passwords/patterns. To practically implement WiPass, we are facing the difficult feature extraction and complex unlock passwords matching, making the analysis of the finger motions challenging. To conquer the challenges, we use DCASW to extract feature and hierarchical DTW to do unlock passwords matching. Besides, the combination of amplitude and phase information is used to accurately recognize the passwords/patterns. We implement a prototype of WiPass and evaluate its performance under various environments. The experimental results show that WiPass achieves the detection accuracy of 85.6% and 74.7% for passwords/patterns detection in LOS and in NLOS scenarios, respectively.

1. Introduction

With the boom of mobile smart devices, mobile sensing on smart devices has become a new style of applications and more and more people rely on the smart devices since the rich functionalities and enhanced computing power conveniently provide intelligent service for peoples' daily lives. Most of the smart devices are equipped with a variety of sensors and kinds of functionalities to enhance sensing capabilities, such as detecting the vehicle steering maneuvers using gyroscope and accelerometer [1]. However, current researches have paid much attention on how to process the sensing data 4Vs (Volume, Velocity, Variety, Veracity) to enhance sensing capabilities; the security of mobile smart devices themselves has not received much attention. The sensors or functionalities on the smart devices may leak the users' privacy, since the smart devices are carrying much sensitive personal information, such as personal photos, credit card numbers, and passwords. Once the smart devices are attacked, the sensitive personal information is prone to leak, bringing the privacy leakage and even financial loss.

Previous studies have shown that the accelerometer and gyroscope can track users [2], and the accelerometers on the devices can recognize the unlock passwords of touch-enabled screen devices [3]. However, previous sensor attacks against unlock passwords [3–5] just aim at digital unlock passwords and successfully decode the digital unlock passwords; for graphical unlock passwords, as shown in Figure 1, it has not been mentioned. Besides, it is known that the sensors on the smart devices may lead to the leakage of users' privacy; however, can the functionalities of the smart devices leak the users' privacy?

In this paper, we present WiPass, a snooping method that does not require attacker close to the target or have control of the device. Only the wireless hotspot functionality is used in WiPass to recognize the graphical unlock passwords. WiPass

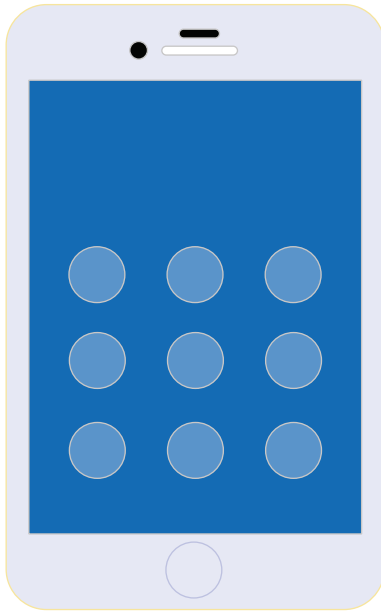


Figure 1: Graphical unlock passwords in the screen of current smart mobile devices.

can "see" your passwords/patterns through the impacts of finger motions on wireless signals even in NLOS (Nonline of Sight) scenarios.

Many existing works have already demonstrated the feasibility of leveraging the impacts of body motions on wireless signals to do localization [6], gesture recognition [7, 8], and even keystroke detection [9, 10]. However, most of existing methods are not suitable to recognize the unlock passwords/patterns. Most existing recognition methods are used in control systems. The user in a control system tends to comply and performs predefined gestures near the devices. However, in WiPass, the attacker cannot access the target devices. The impacts of finger motions on wireless signals from the devices not close to the target are not easy to extract since the impacts are easily overwhelmed by the significant noise.

The differences between attack and control systems bring new challenges. First, it is nontrivial to extract the influenced signal traces among the sampled sequence with intrinsic noise. Second, recognizing the finger motions under the serious noisy environment is challenging. Existing methods usually leverage the amplitude information which is suspected to be corrupted under the noisy environments, decreasing the detection accuracy significantly. Third, the similarity of many unlock patterns significantly increases the difficulty of accurate recognition.

To cope with those challenges, WiPass constructs *finger motion profiles* for the influenced signal traces of different unlock passwords. First, a common method to extract the influenced signal traces is a sliding window. However, in general, a threshold is needed for a sliding window and the threshold is obtained through abundant experiments; it will be time-consuming. Besides, there are lots of different unlock passwords, different unlock passwords correspond to different influenced signal traces, and different influenced signal traces correspond to different amplitude information of wireless signals; thus, different thresholds need to be set for different unlock passwords. Thus, a new efficient method needs to be considered to extract the influenced signal traces. In this paper, DCASW (the difference of cumulative amplitude of the sliding window) is used to extract the influenced signal traces and the max value of the difference can be seen as the beginning of the unlock passwords (where the user starts to unlock the device).

Inspired by time-series data matching method, a well-established technique—Dynamic Time Warping (DTW)—is used to recognize the unlock passwords. However, there are lots of unlock passwords; the matching will be time-consuming and cost large computational overhead; thus, a hierarchical approach is used to reduce time and computational overhead. Given that there are many similar graphical unlock passwords and amplitude information is suspected to be corrupted, phase information can be used with amplitude information together to recognize the unlock passwords and improve the recognition accuracy.

We implement a prototype of WiPass on commercial wireless devices and evaluate its performance under various environments. The experimental results show that, for those unlock passwords with great difference, the recognition accuracy can achieve 70% when using amplitude only. But for those similar unlock passwords, the recognition accuracy can only achieve 37%. The results also show that combining the amplitude information and phase information together can effectively improve the recognition accuracy of similar unlock passwords to 58%. Besides, in LOS scenario, the recognition accuracy of 25 tested graphical unlock passwords can achieve 85.6% within three attempts and 74.7% in NLOS scenario.

Contributions. This paper makes the following contributions:

(i) WiPass is an unlock passwords recognition system, in which a mobile smart phone with wireless hotspot functionality is used as a transmitter to transmit wireless signals, and it exposes a serious threat for mobile device users.

(ii) WiPass exploits the impacts of finger motions on wireless signals to achieve unlock passwords recognition. As a result, the design delivers 74.7% accuracy even in NLOS scenario.

(iii) WiPass uses DCASW to extract the influenced traces and the basic idea can be extended to other systems when different thresholds are needed according to different conditions.

(iv) WiPass also demonstrates the capability of dynamic time warping to recognize the unlock passwords, and a hierarchical approach is used to reduce the time and computational overhead.

The rest of this paper is organized as follows. Section 2 presents the related work about attack against unlock passwords/patterns. Section 3 introduces the overview of

the system, followed by designs in Section 4. Hierarchical approach for unlock passwords recognition is presented in Section 5. Implementation and microbenchmark are introduced in Section 6 and evaluation of the recognition accuracy is presented in Section 7. Section 8 discusses the defense strategies; Section 9 introduces discussions and limitations. Then conclusion is introduced in Section 10.

2. Related Work

Currently, attackers try to attack the unlock passwords to obtain the users' privacy, and there are four main ways that the attackers usually use.

(1) Shoulder Surfing Attack. Mobile devices are often used in public places where shoulder surfing attacks [11, 12] often happen and the unlock passwords are easy to be obtained. It is the most simple way to snoop the unlock passwords and does not need any support of additional hardware. However, shoulder surfing attack only can be done when the attacker and the user are very close and the attacker looks unsuspected. If the users are careful enough during unlocking period, the shoulder surfing attack will not succeed.

(2) Finger Print Attack [13]/Smudge Attack [14]. In fingerprint attack, fingerprint powder is needed to dust the touch screen to reveal fingerprints left from tapping fingers and then the fingerprints are sharpened to obtain the unlock passwords [13]. In smudge attack, the attack is done under a variety of lighting and camera conditions [14]. So, fingerprint attack/smudge attack needs the support of additional hardware (e.g., fingerprint powder/camera). Zhang et al. [13] also suggest that a randomized software keyboard is a feasible solution to prevent the unlock passwords from being obtained.

(3) Video Attack. Shukia et al. [15] introduce one kind of side-channel attacks, and the attack can successfully decode the passwords after several attempts. However, cameras are needed to obtain a video and the success rate is related to the camera configurations. Yue et al. [16] present another side-channel attack, in which webcam or a phone camera is needed instead of a camera. They also design randomized virtual keyboards to defeat the attacks.

(4) Sensors Attack. Sensors are exploited to infer touched keys of touch-enabled screen devices, including orientation sensor, accelerometer, and motion sensors [3–5]. They also point out that the defense strategy is to force every application to declare their intention when accessing the sensors and then inform users about dangerous combinations of permissions.

However, some of the defense strategies mentioned above do not protect the devices completely. For example, the randomized virtual keyboards mentioned above are only put forward to defeat the attacks against digital unlock passwords, and it cannot defeat the attacks against graphical unlock passwords. Besides, most touch-enabled devices such as smart phones have not implemented that functionality. For the defense strategy that aims at sensors attack, it has not been achieved in current touch-enabled screen devices because of the friendly interactive interfaces and many other reasons.

The attack against unlock passwords using wireless signals is always neglected by people, and the attack is similar to gesture recognition system based on wireless signals. However, the attack is different from gesture recognition system, because gesture recognition system [7, 17] can only detect more notable motions because of the limited frequency of the wireless transmission, and those tiny motions cannot be detected. Besides, previous gesture recognition studies used machine learning to recognize the gesture because of the few number of gestures in the control system. However, for unlock passwords, there are a large number of unlock passwords and the influenced signal traces are also different when different people unlock the same kind of devices with the same unlock passwords. Given that, learning-based approach is not appropriate for unlock passwords recognition. This paper introduces an attack against unlock passwords using wireless signals, which is immune to those defense strategies. The attacker can turn on the wireless hotspot functionality of their smart devices and the smart device with hotspot functionality can be used as a transmitter; once the signal reflections from the users' finger motions during the unlocking period are collected by an attacker, the users' unlock passwords will be leaked.

3. System Overview

WiPass is an unlock passwords recognition system that enables mobile smart devices with wireless hotspot functionality to "see" the unlock passwords if the influenced signal traces from finger motions during unlocking period are collected by attackers.

Following a common practice in gesture recognition system, WiPass leverages a wireless transmitter to transmit wireless signals. The difference of WiPass and gesture recognition system is that the transmitter of WiPass is a smart mobile device with wireless hotspot functionality instead of a wireless router. In WiPass, one antenna is enough for receiver to capture signal reflections, and current mobile devices with two omnidirectional antennas can be used as the receiver. Figure 2 illustrates the framework of WiPass. It consists of a transmitter and a receiver. The transmitter transmits wireless signals and the receiver extracts signal reflections from finger motions.

To recognize an unlock password, at a high level WiPass goes through the following steps:

(i) WiPass collects the signal reflection information when there exists an unlock password.

(ii) WiPass removes the noise from the collected signal reflection information using Symlet filter, and the details are introduced in Section 4.2.

(iii) WiPass extracts the influenced signal traces from the noise-removal signal reflection information using DCASW, and the details are introduced in Section 4.3.

(iv) By comparing and matching the desired unlock password's *finger motion profile* with the reference

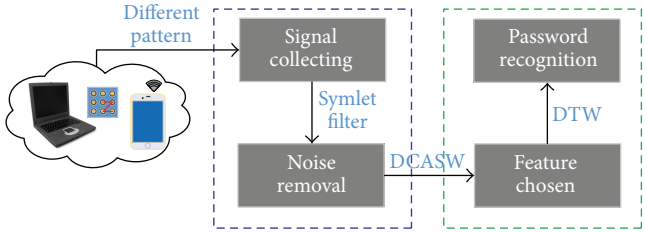


Figure 2: The overview of WiPass.

unlock passwords' *finger motion profiles*, as described in Section 4.5, WiPass identifies the desired unlock password.

In unlock passwords recognition, there are a large number of reference unlock passwords. It is difficult and time-consuming to do unlock passwords matching using DTW, as described in Section 4.5. In Section 5, we describe a hierarchical approach to recognize the desired unlock password.

The next few sections elaborate on the above steps, providing the technical details.

4. Designs

4.1. Signal Collecting. In experiments, the data starts to be collected before the user starts to unlock the device and ends being collected after the user ends unlocking the device. The purpose of such collection is that we need to make sure that the collected data contains the influenced signal traces during the unlocking period. What we have collected is a sequence of CSI data and each CSI represents the phases and amplitudes on a group of 30 OFDM subcarriers.

4.2. Noise Removal. After obtaining the signal, noise needs to be removed from obtained signal, because when the signal is collected, it is unavoidable that the noise in the environment is also collected. For example, additive white Gaussian noise is common in the environment, and the collected signal always contains such noise. In this paper, discrete wavelet decomposition is used to remove noise from obtained signals [18]. Using wavelet decomposition has the following twofold advantages:

(1) It facilitates signal analysis on both time and frequency domain. This attribute can be leveraged in WiPass for analysing the finger motions in varied frequency domains. It can also help WiPass locate the start time for finger motions when one unlock password happens.

(2) It achieves fine-grained multiscale analysis. In WiPass, the finger motions share a lot in common when the unlock passwords of touch-enabled screen devices are similar, such as the "Z" in the top left corner and the "Z" in the bottom right corner, and it makes them difficult to be distinguished. By applying discrete wavelet packet transform to the original signals y_i that contains noise, the tiny differences can be figured out among the similar unlock passwords.

The steps of noise removal using discrete wavelet decomposition are usually as follows.

4.2.1. Forward Wavelet Transform. Generally, a discrete signal $f[n]$ is approximated by the following equation [18]:

$$f[n] = \frac{1}{\sqrt{M}} \sum_{k} W_{\phi}[j_0, k]\, \phi_{j_0,k}[n] + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_{k} W_{\psi}[j, k]\, \psi_{j,k}[n], \tag{1}$$

where $f[n]$ represents the original discrete signals, and it is defined in $[0, M-1]$ while there are totally M points. $\phi_{j_0,k}[n]$ and $\psi_{j,k}[n]$ are both discrete functions, which are defined in $[0, M-1]$, and they are called wavelet basis. In general, the basis sets $\phi_{j_0,k}[n]_{k\in Z}$ and $\psi_{j,k}[n]_{(j,k)\in Z^2, j\geq j_0}$ are chosen to be orthogonal to each other in order to obtain the wavelet coefficients conveniently in the decomposition process.

During the decomposition process, first the original signals are divided into approximation coefficients (i.e., $W_{\phi}[j_0, k]$) and detail coefficients (i.e., $W_{\psi}[j, k]$). Then the approximation coefficients and detail coefficients are both iteratively divided into approximation coefficients and detail coefficients, just as the strategy in the division. The division is an iterative step and the times of iteration depend on the level of decomposition, as shown in Figure 3. The approximation coefficients $W_{\phi}[j_0, k]$ and detail coefficients $W_{\psi}[j, k]$ in each level can be computed as the following equations when $j \geq j_0$:

$$W_{\phi}[j_0, k] = \frac{1}{\sqrt{M}} \sum_{n} f[n]\, \phi_{j_0,k}[n], \qquad W_{\psi}[j, k] = \frac{1}{\sqrt{M}} \sum_{n} f[n]\, \psi_{j,k}[n]. \tag{2}$$

Given the distortion of signals, we apply a two-level decomposition in this paper.

4.2.2. Threshold Quantification. The threshold plays a very important role in denosing process. A small threshold value will still retain the noisy coefficients while a large threshold value will lose the coefficients that may contain the useful information of the influenced signals. There are two types of threshold, and they are separately soft threshold and hard threshold. For hard threshold, set the smaller coefficients to zero while keeping the larger coefficients. For soft threshold, set the smaller coefficients to zero while shrinking the large coefficients towards zero. Based on that and the effectiveness and simplicity of soft threshold and its frequency of use in literature [19, 20], soft threshold is used in this paper.

4.2.3. Inverse Wavelet Transform. Through the above two steps, the original signals experience n-level decomposition, and the numbers of approximation coefficients and detail coefficients are both 2^{n-1}, so the next step is using the coefficients to reconstruct the signal to achieve noise removal.

However, the reconstruction efficiency relies on the selection of wavelet basis. There are 15 kinds of wavelet basis that

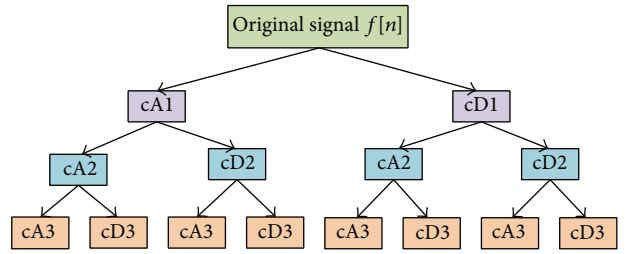


Figure 3: An example of 3-level discrete wavelet packet decomposition, where in the figure cA and cD separately represent the approximation coefficients and detail coefficients.

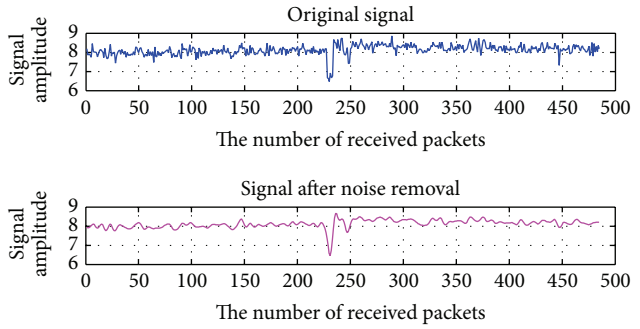


Figure 4: The comparison between original signals and signals after noise removal.

Matlab can support and the most commonly used are the three following families: Daubechies, Coiflets, and Symlets [20]. An ideal wavelet basis should contain the following features: orthogonality, short support, symmetry, smoothness, and high order of vanishing matrix [21]. However, Symlet is an improvement of Daubechies, and the symmetry of Coiflets is higher than that of Daubechies. So Symlets or Coiflets can be used to achieve noise removal, but which kind of wavelet basis is better? Actually, after the wavelet transform, what we have obtained are the coefficients and the coefficients reflect the main information of original signal, so when the signal that is reconstructed by coefficients is more similar to the original signal, the reconstructed signals will not lose useful information of the original signal. Compared with Symlets and Coiflets, the constructed signal of Symlets is more similar to the original signal; besides Wang et al. [18] and Chavan et al. [22] also use Symlets to achieve noise removal. Thus, in this paper, a two-level Symlets wavelet filter is applied to remove noise and the signal after noise removal is as shown in Figure 4.

4.3. Feature Extraction. Feature extraction is important for *finger motion profile* construction. In this paper, we define the influenced signal traces as *features*, and thus the *features* just reflect the unlock passwords. If the extracted *features* are too little, the extracted *features* will not fully reflect the unlock passwords, and if the extracted *features* are too many, redundant information about the signal will be stored and that will lead to a waste of the space and large computational overhead. So how can we automatically extract the *features* and the extracted *features* just reflect the unlock passwords?

Inspired by the sliding window *feature extraction* [23], the cumulative amplitude of the sliding window can be used to extract the *features*. However, for the cumulative amplitude of the sliding window, the *features* are usually extracted according to thresholds and the thresholds are generally obtained after many attempts in actual experiments, the process is time-consuming. Besides, there are many unlock passwords for touch-enabled screen devices and the impacts of finger motions on wireless signals for different unlock passwords are different; thus, for different unlock passwords, different thresholds need to be set to extract *features*. Thus, a new efficient method to extract *features* is needed to be considered.

In this paper, difference of the cumulative amplitude of the sliding window (DCASW) is used to extract the feature. DCASW needs no threshold; thus, it reduces the time overhead. The accumulated amplitude of the sliding window can be calculated by the following equation:

$$F_i = \left|\mathrm{Sum}_i - \mathrm{Sum}_{i-\tau}\right|, \tag{3}$$

where τ is the size of the sliding window and Sum_i is the cumulative amplitude of the sliding window, which can be computed as follows:

$$\begin{aligned} \mathrm{Sum}_i &= \mathrm{Sum}_{i-1} + A_i; \\ \mathrm{Sum}_0 &= 0, \\ \mathrm{Sum}_1 &= A_1, \end{aligned} \tag{4}$$

where A_i represents the amplitude of ith received packets. Then the difference of cumulative amplitude of the sliding window is computed to extract the *feature*, and the computation is as follows:

$$D_i = F_i - F_{i-1}. \tag{5}$$

The max value of the difference can be seen as the beginning of the unlock passwords (where the user starts to unlock the device). That is because when the unlock password begins, the signals begin to fluctuate while the signals keep stable when there is no unlock password, as shown in Figure 4. So the max value of the difference can be thought to be the beginning of the unlock passwords. When the unlock passwords end, the signal will return to keep stable, and the min value of the difference that occurs after the max value can be thought to be the ending of the unlock passwords. The result of the feature extraction using DCASW is shown in Figure 5.

4.4. Finger Motion Profile Construction. After removing the noise from the collected original signals and extracting *features*, what we have obtained is a sequence of cleaned CSIs. Each CSI represents the phases and amplitudes on a group of 30 OFDM subcarriers. Since the noise has been removed from the signals, there would be little dramatic fluctuation caused by interference or noise [24]. Thus, the cleaned CSIs

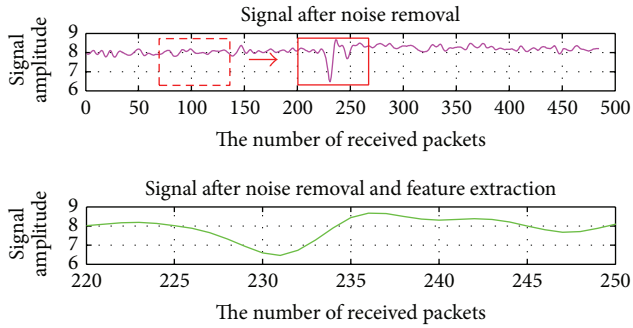


Figure 5: The comparison between signals after noise removal and signals after noise removal and feature extraction.

can represent the influenced signal traces caused by finger motions (*features*), and we define the cleaned CSIs as a *finger motion profile*.

4.5. Unlock Passwords Recognition. After building the *finger motion profiles*, the next work is how can we recognize the different *finger motion profiles* and further recognize the unlock passwords. Having recognized the similarity between time-series data matching and unlock passwords recognition, we borrow the technique dynamic time warping (DTW) from time-series data matching to recognize the unlock passwords.

Dynamic time warping is widely used in time-series data matching, and it is used to quantify the similarity of two time-series data sets. However, our work is quantifying the similarity of two signals; they have something in common. Besides, Rath and Manmatha [25] exploit the potentiality of dynamic time warping to match word image, Wang and Katabi [26] evaluate the similarity between the multipath profiles of the desired tag and the multipath profiles of the reference tags by using dynamic time warping, and many others [18] also leverage dynamic time warping to achieve the evaluation of similarity between two series. Thus, we can use dynamic time warping to quantify the similarity between the signals of two different unlock passwords.

The input of DTW is two signals, one is reference signal and another is desired signal, and the output of DTW is a calculated distance. When given a desired signal, what we want to know is which reference signal is the most similar to the desired signal. The only measurement index is the calculated distance, and the reference signal whose calculated distance with desired signal is the minimum can be thought to be the most similar to the desired signal [27, 28].

5. Hierarchical Approach for Unlock Passwords Recognition

After computing the distances between the desired signals and reference signals, WiPass will identify the unlock passwords. However, there exist plenty of reference signals. When the distances are computed between desired signal and all those reference signals, that will cost a lot of time and computational complexity will be high. Hence, in order to keep the cost and computational complexity low, WiPass recognizes the unlock passwords hierarchically using the protocol below.

Protocol. In stage 1, several *finger motion profiles* of each type of unlock passwords are chosen as the reference signals. Then DTW will compute the distances between the desired signal and reference signals. In the computed distances, there will exist a type of the unlock passwords whose distance is much smaller than other types, and the desired unlock password is thought to belong to that type.

In stage 2, the unlock passwords with similar shape can be thought to belong to one kind, and the different kinds of unlock passwords are chosen as the reference signals. Similar to stage 1, the unlock passwords will belong to one kind of the unlock passwords with similar shape.

In stage 3, the unlock passwords will be matched with the kind of unlock passwords and finally the desired unlock password will be recognized.

Computational Complexity. The complexity of WiPass comes from the number of the reference signals and the length of the features of desired signals and reference signals. Let N be the total number of reference signals, let L_1 be the length of the feature of desired signals, and let L_2 be the length of the feature of the reference signals. Thus, recognizing the desired unlock password has a complexity of $O(NL_1L_2)$. Using hierarchical approach can reduce the complexity to $O(nL_1L_2)$, where n is total number of reference signals that are matched with the desired signal, and $n \ll N$.

The runtime of unlock passwords matching is 37.131518 seconds when the system computes the calculated distances between one desired unlock password and 25 reference unlock passwords (the length of the unlock passwords is more than 300 packets). The runtime of unlock passwords matching is 9.668004 seconds when the system computes the calculated distances between one desired unlock password and 5 reference unlock passwords (the length of the unlock passwords is more than 300 packets). The runtime of *finger motion profile* matching is 0.243151 seconds when the system computes the calculated distances between one desired *finger motion profile* and 5 reference *finger motion profiles* (the lengths of the *finger motion profiles* are 60 packets). The experiments are done using MATLAB R2012b on a 64-bit machine with Intel Core i3-4150 Quad-Core processor and 8 G memory. The actual runtime experiments demonstrate that the complexity of WiPass is positively correlated to the number of the reference signals and the length of the features.

6. Implementation and Microbenchmark

We implement WiPass on current mobile smart devices with the wireless hotspot functionality and evaluate its performance in typical indoor scenarios.

6.1. Hardware and Scenarios. A smart device with wireless hotspot functionality is used as the transmitter (IPhone 6 plus), and a desktop equipped with Intel 5300 NIC (Network Interface Controller) is used as the receiver. The transmitter operates in IEEE 802.11n. The receiver has 3 working antennas

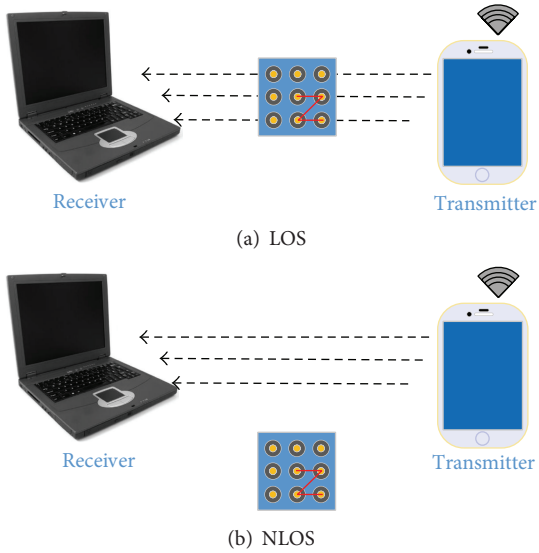


Figure 6: Experiment scenarios.

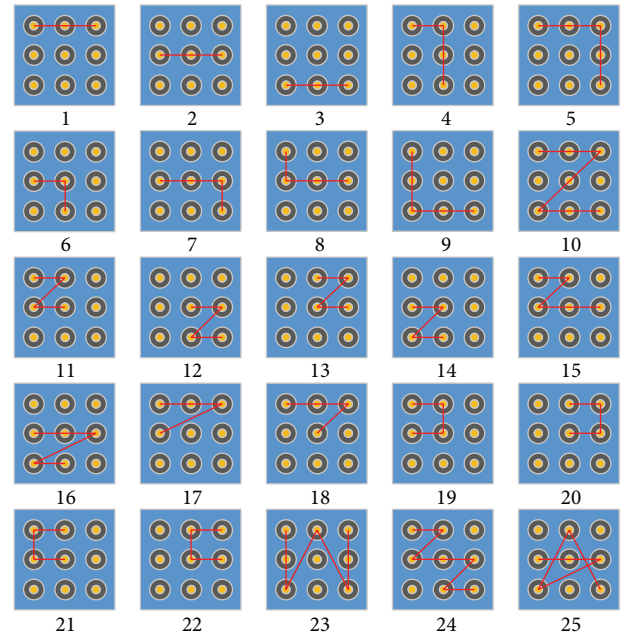


Figure 7: 25 tested unlock passwords in the experiment.

and the firmware is modified to report CSIs to upper layers. During the measurement, the receiver continuously pings packets from the smart devices at the rate of 5 packets per second. The collected CSIs are stored and processed at the receiver. The tested mobile phone is SAMSUNG Galaxy Note 3.

The experiments are conducted in a typical indoor office whose area is 3.6 m × 6.6 m. To evaluate WiPass's performance, the experiments are done in two scenarios, LOS scenario and NLOS scenario.

LOS Scenario: Line of Sight. The target person is just on the straight line between transmitter and receiver and is within the radio range of the transmitter, as shown in Figure 6(a).

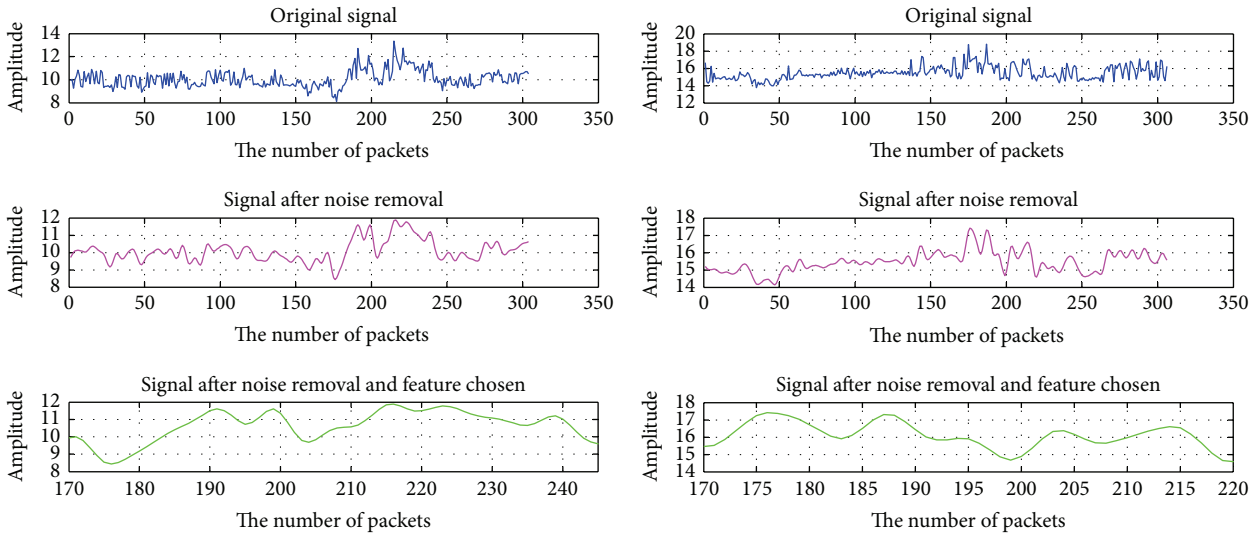
NLOS Scenario: Nonline of Sight. The target person is not on the straight line between transmitter and receiver but also is within the radio range of the transmitter, as shown in Figure 6(b).

6.2. Unlock Passwords Vocabulary. The unlock passwords are divided into four types, some of each type of unlock passwords are chosen as the tested unlock passwords, and 25 unlock passwords are chosen randomly as the tested unlock passwords. All the tested graphical unlock passwords can be divided into four types, and one type of the unlock passwords is that there is no inflection points in the unlock passwords, one is one inflection point in the unlock passwords, one is two inflection points in the unlock passwords, and the last is three or more than three inflection points in the unlock passwords. As shown in Figure 7, unlock passwords pattern 1, pattern 2, and pattern 3 can be thought to be the first type of unlock passwords, and unlock passwords pattern 4 and pattern 5 can be thought to be the second type, unlock passwords pattern 10 and pattern 11 can be thought to be the third type, and unlock passwords pattern 23, pattern 24, and pattern 25 can be thought to be the last type. In the second type, there are three kinds of unlock passwords, pattern 4 to pattern 7 belong to the first kind, pattern 8 and pattern 9 belong to the second kind, and pattern 17 and pattern 18 belong to the third kind. The first three types of unlock passwords can be thought to be simple unlock passwords, and the last can be thought to be complex unlock passwords.

The lengths of extracted features (the number of received packets for the influenced signal traces) of different types of unlock passwords are different, since the time spent on unlocking for simple unlock passwords and complex unlock passwords is different when the same group of persons unlock the same-size touch-enabled devices. The impacts of finger motions of different kinds of unlock passwords are also different; thus, the hierarchical approach is feasible theoretically.

6.3. Microbenchmark Experiment. We start with a microbenchmark experiment to provide insights into the working of WiPass. In order to better understand how unlock passwords influence the wireless signals, we conduct a simple experiment of two different unlock passwords. The experiments are conducted in the conditions that there are no surrounding people in the environment and the user does not move while unlocking the devices.

Figure 8 shows the signals under different conditions for two different graphical unlock passwords when the transmitter is current smart mobile phone with wireless hotspot functionality. As Figure 8 shows, the impacts of finger motions of different unlock passwords on wireless signals are different. When there is no surrounding people in the environment and there exist no unlock passwords, the collected signals keep relatively stable. When the user starts to unlock the device, the collected signals will fluctuate, and



(a) Graphical unlock password pattern 25

(b) Graphical unlock password pattern 5

Figure 8: Microbenchmark experiments of two different unlock passwords.

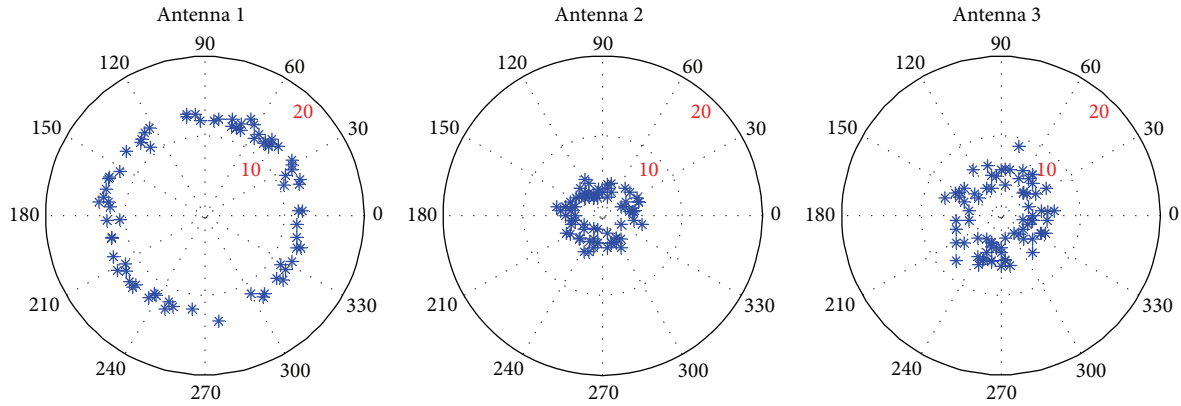
when the user ends unlocking the device, the signals will return to be relatively stable. Thus, the unlock passwords can be recognized.

6.3.1. Recognition among Unlock Passwords with Great Difference. We can see from Figure 8 that, after noise removal and feature chosen, the signals are different in amplitude. In Figure 8(a), after noise removal and feature chosen, the signal amplitude of the unlock password pattern 25 is between 8 and 12 while, in Figure 8(b), the signal amplitude of unlock password pattern 5 is between 14 and 18. Besides, when there exists great difference among the unlock passwords, the lengths of the features of unlock passwords are different. For example, in Figure 8(a), the length of the feature of unlock password pattern 25 is 75 while the length of the feature of unlock password pattern 5 is 50. That is because, for different types of unlock passwords, the spent time is usually different, and, after noise removal and feature chosen, the lengths of the features are different, as shown in Figure 8. Thus, using the signal amplitude and the length of the feature can distinguish the unlock passwords with great difference. In addition, there is another information that can be obtained from CSIs except amplitude and it is phase. Using phase can also recognize the unlock passwords with great difference successfully. Figure 9 can demonstrate it. Figures 9(a) and 9(b) separately represent the relationships between amplitude and phase of two different unlock passwords with great difference. We can see from Figure 9 that the relationships between phase and amplitude are different no matter what in terms of one antenna or in terms of three antennas. So, for those unlock passwords with great difference, the unlock passwords can be recognized using amplitude and the length of features or using amplitude, phase, and the length of features.

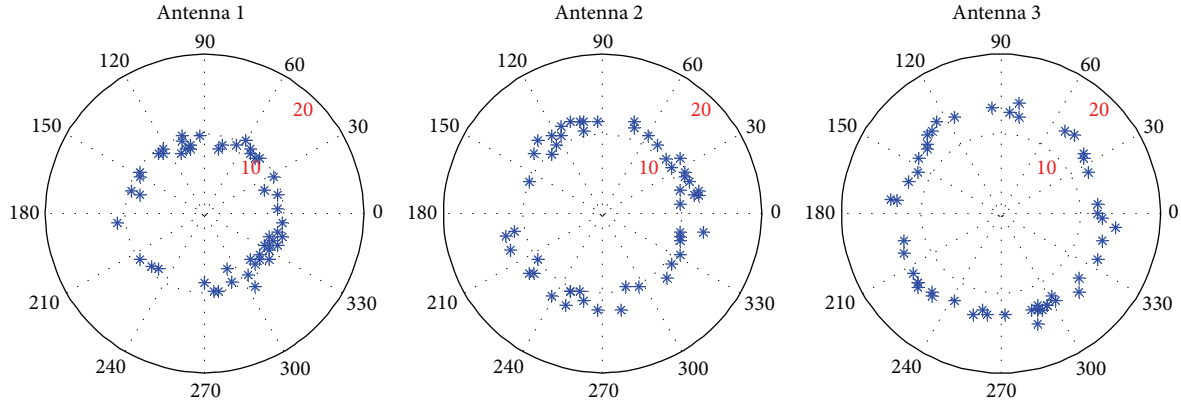
6.3.2. Recognition among Similar Unlock Passwords. For those graphical unlock passwords with great difference, the representations of signals are different in amplitude in time domain, as shown in Figure 8. However, for those similar unlock passwords, such as unlock passwords pattern 11, pattern 12, pattern 13, and pattern 14, the representations of signals are similar in amplitude in time domain and the lengths of the features are also the same. Besides, the relationships between amplitude and phase are also similar, as shown in Figure 10. Figures 10(a) and 10(b) represent separately the relationships between amplitude and phase of unlock passwords pattern 11 and pattern 13. We can see from Figure 10 that, for those similar unlock passwords, the relationships between amplitude and phase are also similar no matter what in view of one antenna or three antennas. So how can we recognize those similar unlock passwords? It is known that phase is another information that can be obtained from CSIs, and it can be expressed in time domain, as shown in Figure 11. We can see from Figure 11 that the phases of pattern 11 and pattern 13 are different in time domain no matter what in view of one antenna or three antennas. Thus, for those similar unlock passwords, when their signal amplitudes are similar, the lengths of features are the same, and the relationships between amplitude and phase are also similar, the phase information can be used to recognize the unlock passwords successfully. So, in this paper, amplitude and phase are used together to recognize the similar unlock passwords.

7. Evaluation

In this section, the recognition accuracy of graphical unlock passwords when using amplitude only and using amplitude and phase together is computed. This section also compared



(a) Graphical unlock password pattern 25



(b) Graphical unlock password pattern 5

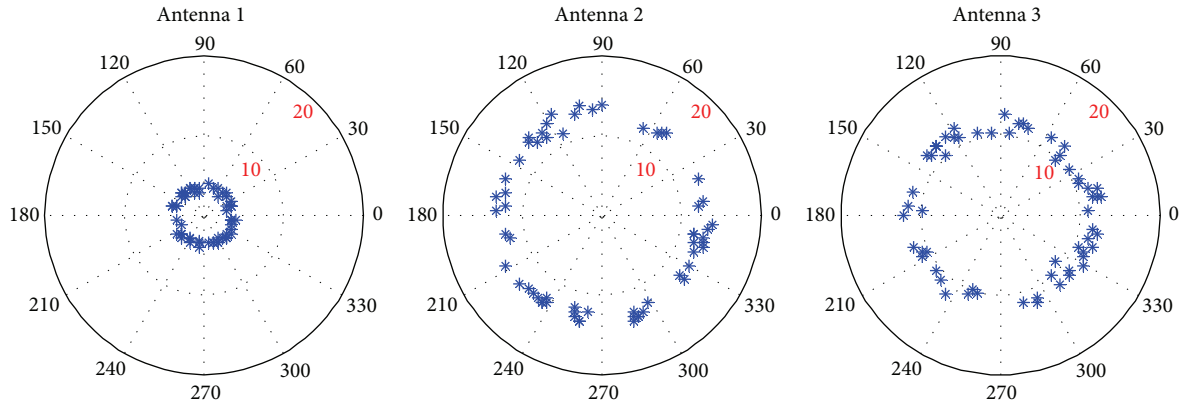
Figure 9: Comparison of the relationships between phase and amplitude of two graphical unlock passwords with great difference after noise removal and feature chosen.

the complexity of feature chosen when using AP (Access Point) as the transmitter and when using a smart mobile phone with wireless hotspot functionality as the transmitter.

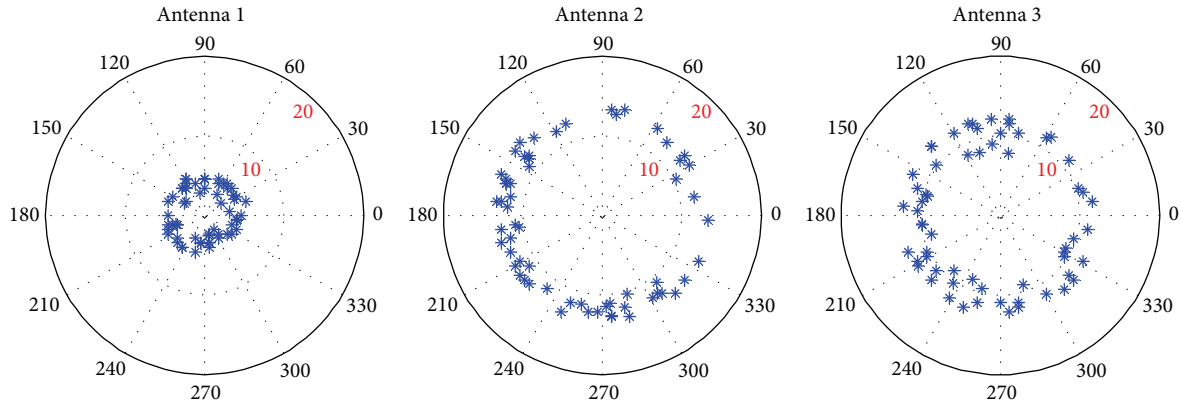
7.1. Wireless Device Diversity: Router versus Smart Phones. We can see from Figure 12 that when the transmitter is TP-Link wireless router and there exist no unlock passwords, the signals after noise removal keep stable, as shown in Figure 12, when the number of received packets is between 0 and 160. However, when the transmitter is the smart phone with the wireless hotspot functionality, the signals after noise removal just keep relatively stable, as shown in Figure 8. That is because, for TP-Link wireless router, there is only one kind of antenna, and it is WiFi antenna and it is used to transmit the wireless signals while, for smart phones with the wireless hotspot functionality, there are also other antennas besides WLAN antenna, such as communication antenna, GPS antenna, Bluetooth antenna, and NFC antenna. When there exist no unlock passwords and the environment is stable, there is no other interferences that influence the signals; thus, the signals after noise removal keep stable when the transmitter is TP-Link wireless router, while when the transmitter is the smart phone with wireless hotspot functionality, there will exist other interferences coming from other antennas that influence the signals; thus, the signals after noise removal just keep relatively stable. Thus, when the transmitter is TP-Link wireless router, the feature chosen is easier than when the transmitter is a smart phone with wireless hotspot functionality. In this paper, the smart phone with wireless hotspot functionality is used as the transmitter just because, for those places where there does not exist a wireless router (e.g., in the bus), the attack against unlock passwords cannot occur; however, it should be a warning for mobile phone user that the attack can occur when the attacker has a smart mobile phone with wireless hotspot functionality regardless of the place where the attacker is.

7.2. Graphical Unlock Password Accuracy. In this section, the accuracy of graphical unlock passwords is tested. In order to test the accuracy of similar unlock passwords, some of the unlock passwords of each type are tested. As shown in Figure 7, 25 unlock passwords are tested and each unlock password is tested 20 times.

7.2.1. Recognition Using Amplitude Only. For those unlock passwords with great difference, the lengths of features and the amplitude of the signals are usually different; thus, using amplitude can recognize the unlock passwords successfully. In order to demonstrate it, 6 unlock passwords are chosen as the tested unlock passwords, and they are separately unlock



(a) Graphical unlock password pattern 11



(b) Graphical unlock password pattern 13

FIGURE 10: Comparison of the relationships between phase and amplitude of two similar graphical unlock passwords after noise removal and feature chosen.

passwords pattern 2, pattern 5, pattern 9, pattern 10, pattern 21, and pattern 23. The results are shown in Figure 13. We can see from Figure 13 that the recognition accuracy can achieve 60% at least, and the recognition accuracy is between 60% and 80%. The average recognition accuracy of the six unlock passwords is 70%; thus, for those unlock passwords with great difference, using amplitude only can recognize the unlock passwords. However, for those similar unlock passwords, using amplitude only cannot recognize the unlock passwords successfully. To further demonstrate it, three groups of similar unlock passwords are tested, and one group is unlock passwords pattern 4, pattern 5, pattern 6, and pattern 7, one group is unlock passwords pattern 10, pattern 11, pattern 12, pattern 13, and pattern 14, and the last group is unlock passwords pattern 19, pattern 20, pattern 21, and pattern 22. The results of recognition accuracy are shown in Figure 14. As shown in Figure 14, the recognition accuracy is low and most of the accuracy are between 20% and 50%, except pattern 10 and pattern 21. The average recognition accuracy of the three groups of similar unlock passwords is 37%. Thus, using amplitude only cannot recognize those similar graphical unlock passwords successfully.

7.2.2. Recognition Using Amplitude and Phase. We know from Figure 14 and the analysis of Section 6.3.2 that when amplitude cannot recognize the similar unlock passwords successfully, phase information can help to distinguish them. To demonstrate it, the amplitude information and phase information of the three groups of similar unlock passwords are leveraged together to recognize them. The results are shown in Figure 15. Comparing with Figure 14, we know that the recognition accuracy improved significantly and the average recognition accuracy can achieve 58%. Thus, amplitude information and phase information can be used together to improve the recognition accuracy of similar graphical unlock passwords.

Figure 16 shows the results of recognizing the 25 tested unlock passwords with one attempt. We can see from Figure 16 that the recognition accuracy of most unlock passwords is above 60%, and the average recognition accuracy can achieve 66%. However, because DTW computes distances between the desired unlock password and reference unlock passwords and if the computed minimum distance is not matched with the desired unlock passwords, the unlock passwords can be matched with the second minimum distance, third minimum distance. That means we can try to unlock the device with two attempts or with three attempts. After three attempts or less than three attempts, the recognition accuracy of most unlock passwords can achieve above 80%, and the average accuracy is 85.6%, as shown in Figure 17.

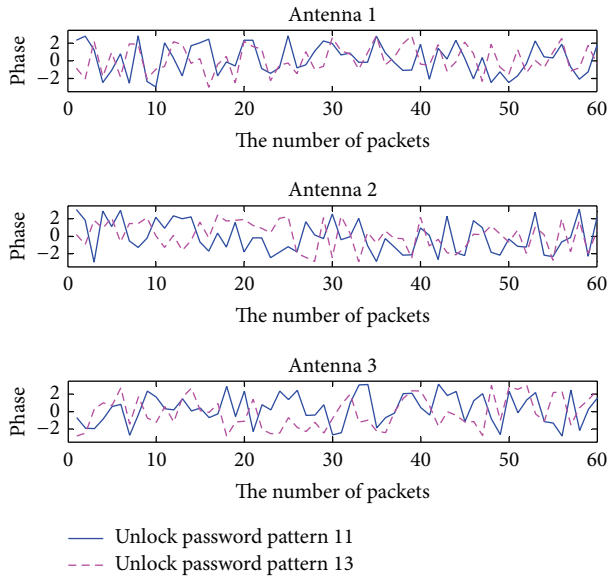


Figure 11: The comparison of phase after noise removal and feature chosen between the graphical unlock password pattern 11 and graphical unlock password pattern 13.

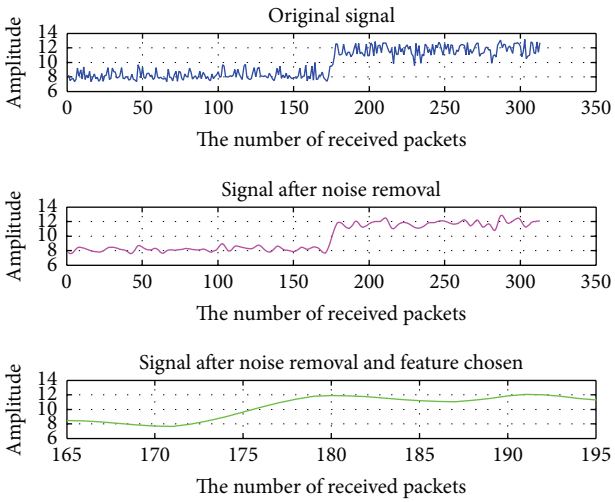


Figure 12: The signals of graphical unlock password pattern 5 when the transmitter is TL-WR740N wireless router.

In Figures 16 and 17, the successful recognition accuracy of similar unlock passwords is relatively low; for example, the successful recognition accuracy of unlock passwords pattern 11, pattern 12, and pattern 13 in Figure 16 are just separately 40%, 30%, and 40%. However, after several attempts of similar unlock passwords, the accuracy will be improved and, after enough attempts of similar unlock passwords, the desired unlock password will be recognized.

7.2.3. Recognition in NLOS Scenario. The above experiments are done in LOS scenario. In most cases, the attack occurs in NLOS scenario; thus, the recognition accuracy in NLOS scenario is also needed to be considered. Figure 18 shows the

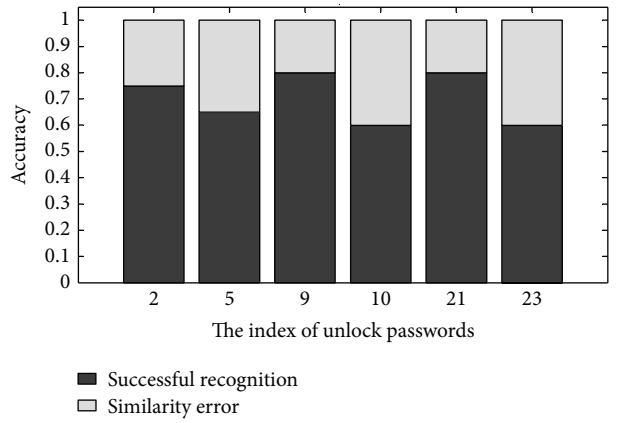


Figure 13: The accuracy of 6 tested graphical unlock passwords with great difference when amplitude is used only.

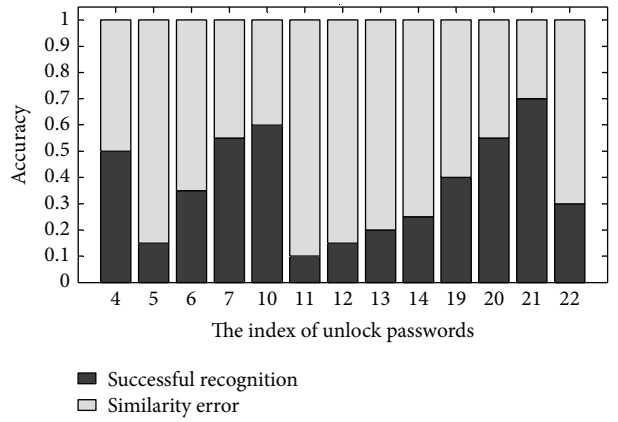


Figure 14: The accuracy of 13 tested similar graphical unlock passwords when amplitude is used only.

recognition results of 25 tested graphical unlock passwords within three attempts in NLOS scenario. We can see from Figure 18 that most of recognition accuracy of 25 tested graphical unlock passwords is between 50% and 80% and the average recognition accuracy can achieve 74.7%. Thus, in NLOS scenario, the unlock passwords can be recognized successfully within three attempts. Comparing the recognition accuracy in LOS scenario with that in NLOS scenario, the recognition accuracy in NLOS scenario is lower than that in LOS scenario. That is because, in NLOS scenario, the signal reflections from finger motions are weaker than that in LOS scenario; thus, the accuracy is lower in NLOS scenario than that in LOS scenario.

8. Defense Strategies

Unlock passwords are vulnerable to various attacks, including the attack using wireless signals. In this section, we discuss a few strategies to improve the security and protect the privacy of touch-enabled screen device users.

A few strategies are available to mitigate video attack, sensors attack, and the attack using wireless signals for

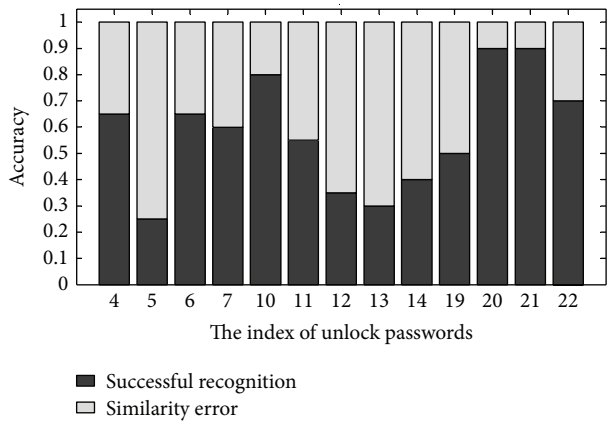


Figure 15: The accuracy of 13 tested similar graphical unlock passwords when amplitude and phase are used.

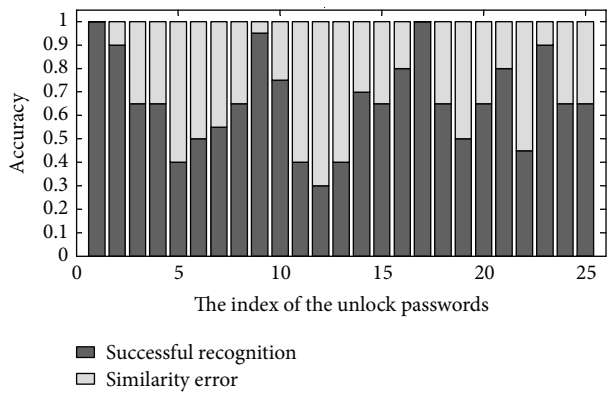


Figure 16: The accuracy of 25 tested graphical unlock passwords when amplitude and phase are used.

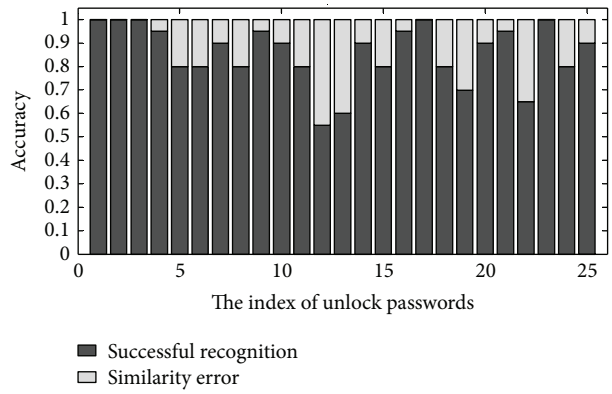


Figure 17: The accuracy of 25 tested graphical unlock passwords within three attempts when amplitude and phase are used.

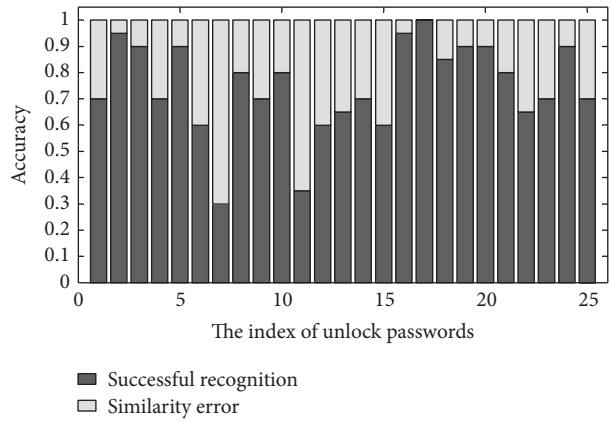


Figure 18: The accuracy of 25 tested graphical unlock passwords within three attempts in NLOS scenario.

touch-enabled screen devices without modifying the devices. The first one is setting complex passwords. Setting complex unlock passwords can defeat the attacks to some extent; however, if the complex unlock passwords are common and it can be thought of by the attacker in advance, the unlock passwords will be decoded. Besides, setting complex unlock passwords is inconvenient for users, especially for those who input passwords frequently.

Another defense strategy is that in public places, especially when there are persons near to you, we try not to unlock the devices so that the unlock passwords cannot be obtained by the attacker to the maximum possible extent. However, it is very troublesome for people, because people need to be careful when there are people near to them.

One more defense strategy is unlocking the smart devices using fingerprints, and it is the most safe unlocking. Current touch-enabled screen devices should go ahead for that fingerprints unlocking direction.

9. Discussions and Limitations

In this section, we discuss the limitations of our implementation.

(1) User Movement. In this paper, the unlock passwords can be recognized when the target person does not move. It is possible that the person unlocks the device while walking. However, the device-free localization techniques [29] can achieve real-time tracking, so combining with WiPass, we can achieve the recognition of unlock passwords when the target person moves.

(2) Impact of Surroundings. Movements of surrounding people sometimes can reflect the wireless signals more significantly than finger motions do. In this paper, we assume that there are no surrounding moving people when the target person is unlocking the devices. It is possible that there are no surrounding moving people in a silent coffee shop or library. However, there will always exist surrounding moving people in most public places, and that can be solved by MIMO technique [30]. Then MIMO beamforming will be leveraged to focus on the targets' fingers to reduce irrelevant multipath effects.

(3) Devices Diversity. There are many kinds of smart touch-enabled screen devices. For each kind of touch-enabled screen devices, the size of touch screen is different; thus, the

positions of each keypads are different. In this paper, just one kind of mobile phone is tested to demonstrate that the unlock passwords can be recognized using wireless signals and, in future work, more experiments on other kinds of smart devices will be conducted and the similar smart devices can be classified into one group (the size of touch screen and the positions of keypads are the same) to recognize the unlock passwords in order to be time-saving and labor-saving.

(4) The Diversity of Unlocking Speeds. For different people, the unlocking speeds are different. There are four user groups for smart devices, and one is teenager, one is the young, one is the middle-aged, and the last is the old. However, the young is the main user group and, in this paper, the experiments are conducted with the young people. The future work of this paper will analyse the impacts of finger motions on the signals when the unlocking speeds are different and the unlocking passwords of different speeds can be classified into different groups.

(5) The Size of Patterns. There are a great number of different unlock passwords. In this paper, only 25 patterns are considered to demonstrate that your unlock passwords can be leaked through wireless hotspot functionality. It should be a warning for current mobile device users. When the desired unlock passwords are not in the 25 patterns, the desired unlock passwords will not be recognized successfully. However, the performance can be improved by a continuously learning-based approach, where the model keeps evolving using examples collected in the end-users environments, and when a user unlocks the device using the unknown unlock passwords, the unknown unlock passwords will be put into the size of patterns. That will be a continuous process to enlarge the size of patterns and improve the recognition accuracy.

10. Conclusion

This paper presents WiPass, a novel system that enables wireless signals, which are transmitted by a smart device with wireless hotspot functionality, to "see" the unlock passwords. WiPass is easily implemented by current smart devices and does not need any support of additional hardware. To achieve the unlock passwords recognition, WiPass first removes the noise from collected signals using a two-level Symlet filter and then uses DCASW (the difference of the cumulative amplitude of the sliding window) to extract the features to build the *finger motion profiles* and then uses a hierarchical dynamic time warping (DTW) approach to recognize the unlock passwords. The experiment results demonstrate that WiPass can achieve recognition accuracy of 85.6% for graphical unlock passwords in line of sight (LOS), 74.7% in nonline of sight (NLOS). The results also demonstrate that the recognition accuracy can be improved by using amplitude information and phase information together and by adding the times of attempts. We believe that this paper exposes a serious threat for current touch-enabled screen devices users, and such attack can happen in public places where the attacker looks unsuspected.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (no. 61170218, no. 61202198, no. 61202393, no. 61272461, no. 61373177, and no. 61572402), the Key Project of Chinese Ministry of Education (no. 211181), the International Cooperation Foundation of Shaanxi Province, China (no. 2013KW01-02 and no. 2015KW-003), and the China Postdoctoral Science Foundation (Grant no. 2012M521797).

References

[1] D. Chen, K. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible sensing of vehicle steering with smartphones," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 1–13, ACM, Florence, Italy, May 2015.

[2] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "Accelprint: imperfections of accelerometers make smartphones trackable," in *Proceedings of the Network and Distributed System Security Symposium (NDSS '14)*, San Diego, Calif, USA, February 2014.

[3] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: password inference using accelerometers on smartphones," in *Proceedings of the 12th Workshop on Mobile Computing Systems & Applications (HotMobile '12)*, p. 9, ACM, San Diego, Calif, USA, February 2012.

[4] L. Cai and H. Chen, "TouchLogger: inferring keystrokes on touch screen from smartphone motion," in *Proceedings of the 6th USENIX Conference on Hot Topics in Security (HotSec '11)*, p. 9, 2011.

[5] Z. Xu, K. Bai, and S. Zhu, "TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC '12)*, pp. 113–124, ACM, Tucson, Ariz, USA, April 2012.

[6] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, "Nuzzer: a large-scale device-free passive localization system for wireless environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1321–1334, 2013.

[7] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking (MobiCom '13)*, pp. 27–38, ACM, Miami, Fla, USA, October 2013.

[8] H. Abdelnasser, K. A. Harras, and M. Youssef, "WiGest demo: a ubiquitous WiFi-based gesture recognition system," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '15)*, pp. 17–18, IEEE, Hong Kong, May 2015.

[9] B. Chen, V. Yenamandra, and K. Srinivasan, "Tracking keystrokes using wireless signals," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 31–44, ACM, Florence, Italy, May 2015.

[10] K. Ali, A. Liu X, W. Wang et al., "Keystroke recognition using WiFi signals," in *Proceedings of the 21st Annual International*

Conference on Mobile Computing and Networking, pp. 90–102, ACM, Paris, France, September 2015.

[11] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS '07)*, pp. 13–19, ACM, Pittsburgh, Pa, USA, July 2007.

[12] A. Habibi Lashkari, S. Farmand, O. B. Zakaria, and R. Saleh, "Shoulder Surfing attack in graphical password authentication," *International Journal of Computer Science and Information Security*, vol. 6, no. 2, pp. 145–154, 2009.

[13] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu, "Fingerprint attack against touch-enabled devices," in *Proceedings of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*, pp. 57–68, Raleigh, NC, USA, October 2012.

[14] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies (WOOT '10)*, pp. 1–7, USENIX Association, Washington, DC, USA, August 2010.

[15] D. Shukia, R. Kumar, V. V. Phoha, and A. Serwadda, "Beware, your hands reveal your secrets!," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 904–917, ACM, Scottsdale, Ariz, USA, November 2014.

[16] Q. Yue, Z. Ling, B. Liu, W. Fu, and W. Zhao, "Blind recognition of touched keys: attack andcountermeasures," http://arxiv.org/abs/1403.4829.

[17] F. Adib and D. Katabi, "See through walls with WiFi!," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 75–86, 2013.

[18] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with Wi-Fi!," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom '14)*, pp. 593–604, ACM, 2014.

[19] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[20] R. Singh, R. E. Vasquez, and R. Singh, "Comparison of daubechies, coiflet, and symlet for edge detection," in *Visual Information Processing VI*, vol. 3074 of *Proceedings of SPIE*, pp. 151–159, International Society for Optics and Photonics, July 1997.

[21] T. D. Bui and G. Chen, "Translation-invariant denoising using multiwavelets," *IEEE Transactions on Signal Processing*, vol. 46, no. 12, pp. 3414–3420, 1998.

[22] M. S. Chavan, N. Mastorakis, M. N. Chavan et al., "Implementation of SYMLET wavelets to removal of Gaussian additive noise from speech signal," in *Proceedings of the 10th International Conference on Recent Researches in Communications, Automation, Signal Processing, Nanotechnology, Astronomy and Nuclear Physics*, pp. 37–41, February 2011.

[23] D. Chendong, H. Zhengjia, and J. Hongkai, "A sliding window feature extraction method for rotating machinery based on the lifting scheme," *Journal of Sound and Vibration*, vol. 299, no. 4-5, pp. 774–785, 2007.

[24] G. Cohn, D. Morris, S. N. Patel, and D. S. Tan, "Humantenna: using the body as an antenna for real-time whole-body interaction," in *Proceedings of the 30th ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 1901–1910, ACM, May 2012.

[25] T. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II-521–II-527, IEEE, June 2003.

[26] J. Wang and D. Katabi, "Dude, wheres my card? RFID positioning that works with multipath and non-line of sight," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 51–62, 2013.

[27] S. Salvador and P. Chan, "FastDTW: toward accurate dynamic time warping in linear time and space," in *Proceedings of the 3rd SIGKDD Workshop on Mining Temporal and Sequential Data (KDD/TDM '04)*, Seattle, Wash, USA, December 2004.

[28] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*, pp. 69–84, 2007.

[29] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, "Pilot: passive device-free indoor localization using channel state information," in *Proceedings of the IEEE 33rd International Conference on Distributed Computing Systems (ICDCS '13)*, pp. 236–245, IEEE, Philadelphia, Pa, USA, July 2013.

[30] A. L. Moustakas, S. H. Simon, and A. M. Sengupta, "MIMO capacity through correlated channels in the presence of correlated interferers and noise: a (not so) large N analysis," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2545–2561, 2003.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 8686945, 10 pages
http://dx.doi.org/10.1155/2016/8686945

Research Article

Cooperation Dynamics on Mobile Crowd Networks of Device-to-Device Communications

Yong Deng, Guiyi Wei, Mande Xie, and Jun Shao

School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 300018, China

Correspondence should be addressed to Guiyi Wei; weiguiyi@gmail.com

Received 8 January 2016; Accepted 23 February 2016

Academic Editor: Tingting Chen

The explosive use of smart devices enabled the emergence of collective resource sharing among mobile individuals. Mobile users need to cooperate with each other to improve the whole network's quality of service. By modeling the cooperative behaviors in a mobile crowd into an evolutionary Prisoner's dilemma game, we investigate the relationships between cooperation rate and some main influence factors, including crowd density, communication range, temptation to defect, and mobility attributes. Using evolutionary game theory, our analysis on the cooperative behaviors of mobile takes a deep insight into the cooperation promotion in a dynamical network with selfish autonomous users. The experiment results show that mobile user's features, including speed, moving probability, and reaction radius, have an obvious influence on the formation of a cooperative mobile social network. We also found some optimal status when the crowd's cooperation rate reaches the best. These findings are important if we want to establish a mobile social network with a good performance.

1. Introduction

With the rapid development of wireless communication technology, communications between smart devices (such as smart phones, Pads, wearable devices, and smart vehicles) for information dissemination and network resource sharing are becoming more and more common. Numerous researches have been done to solve the design challenges of wireless device-to-device (D2D) communication. Many new approaches have been proposed to make the formation of D2D communication networks possible [1–4]. As an emerging and innovative technology in next-generation cellular networks, LTE-A is capable of making the most of topology in networking its nodes. It enables wireless nodes to communicate directly with each other without support from the network infrastructure. Another enabling technology is WiFi direct, which supports devices to connect with each other for file transfer without requiring a wireless access point. Both of LTE-A and WiFi direct technologies make geography-aware information sharing and collaborative task solving within a crowd easier in our daily life.

The handheld smart devices connect with each other via direct wireless D2D communication when they come across and develop the whole crowd into a self-organized network. We call such network *mobile crowd network* (MCN). In general, there is no central control mechanism to manage the communications between nodes in MCNs. Each node decides to participate in or get out of the network independently by itself. A node shares its resources and cooperates with other nodes voluntarily when it is in the network. Communication links between nodes are usually temporal and volatile due to nodes' movement. Nodes' mobility attributes (including moving probability, velocity, direction, and trajectory) are hard to predict; that is, the topology of a MCN is fully dynamic.

With resource sharing and cooperation between nodes, MCN brings us higher network coverage and bigger network capacity [1, 2]. MCN also allows its nodes to experience additional benefits in terms of smaller communication latency, increased data rate, and reduced energy consumption under well-designed cooperation and incentive mechanism [5]. Furthermore, MCN provides a new paradigm of mobile crowd applications, such as community dynamics monitoring in public safety, traffic monitoring and planning in cities, geography-aware and participatory data collection in environment and weather sensing, and other collaborative

complex task solving on mobile crowd, also known as mobile crowdsensing/crowdsourcing.

D2D Communications in MCNs can be treated as a type of cooperation behavior between nodes that are willing to transport or relay data for the nodes around them by some types of interdependency and reciprocity. It is the collaborative communications that significantly improve the whole network quality and throughput capacity [6]. However, there are some obstacles in promoting the cooperation behaviors that can improve the network performance of MCNs.

First, the MCN is a type of fully distributed and self-organized wireless network. In MCNs, unlike wireless ad hoc networks and wireless sensor networks, there are no network operators or coordinator nodes that work on the whole network level, have rich information of the whole network, and take charge of network performance management. Therefore, it is hard to design and deploy a centralized control mechanism for improving the collaborative communications in MCNs.

Second, most nodes are usually handheld smart devices. The resource sharing at application level or system level may be turned on/off any time anywhere by the device holders. The collaborative communications between these autonomous nodes fully depend on human-centric communication will. Due to the fact that all the mobile devices are carried by different people who are irrelevant, a natural problem is whether the device holders are willing to cooperate when they communicate with their neighbors. As we know, altruistic behaviors in D2D communications, that is, sharing one's own resources with others, induce extra energy consumption and utility loss in storage, computing, and communication. Therefore, it is reasonable that MCN nodes tend to make selfish decisions without effective incentive mechanisms; that is, they do not share their resources with others. Selfish behaviors will destroy the attempt of cooperation promotion in MCNs. Therefore, the incentive mechanisms designed for data-centric self-organized networks, such as mobile ad hoc networks and vehicular networks, can not work effectively in the participatory human-centric networks like MCNs, due to possible frequent human intervention.

Third, mobility attributes of nodes, such as moving possibility, velocity, and direction, also substantially affect the cooperation between nodes. Mobility attributes affect the meeting possibility and communication connectivity between nodes. Ephemeral linkages between nodes make the topology of a MCN highly variable. Thus, topology-based incentive mechanisms can not be directly used to promote the cooperation in MCNs, such as tit-for-tat, reputation schemes, and auction scheme [7, 8] proposed for structured networks. Before designing an effective incentive mechanism for MCNs, we should know how nodes' mobility attributes affect the resource sharing behaviors (cooperation or defection) in a MCN. However, to the best of our knowledge, there is still no comprehensive work on MCNs that investigate the relationship between the nodes' mobility and the cooperation rate of a MCN.

In this paper, our objective is not to design an effective incentive mechanism to promote the cooperation rate of a specified MCN but to explore the cooperation dynamics on generalized MCNs, to investigate the factors that influence the cooperation rate of MCNs, and, more importantly, to analyze the relationships between the cooperation rate and the main influence factors in MCNs. To tackle the above obstacles, we classify the main influence factors on the cooperation rate of MCNs into three categories as below.

(i) Device Factor. The device factor is double-edged [9]. The advantageous aspect of device factor is that a device may benefit from the cooperation with other nodes, such as increasing network performance, enlarging its sensing coverage, and promoting application specific performance. However, the main disadvantageous aspect of device factor is the supernumerary resource consumption incurred by altruistic resource sharing in cooperation, including energy consumption, bandwidth loss, additional CPU cycles, and memory occupation.

(ii) Network Factor. The network factor mainly comes from nodes' mobility, including moving probability, moving speed, and direction changing. Suppose each smart device can communicate with multiple neighbored nodes, simultaneously (in this paper, we analyze D2D communication in the application layer. We consider a node can communicate with multiple other nodes, simultaneously; the implementation of multiple-to-multiple D2D communication in the physical layer is not the focus of this paper). The length of wireless communication link (or radius) and the density of the nodes (or population) in a network also affect the cooperation rate in MCNs.

(iii) Human Will. The smart device holders make the final decision, sharing or not. Suppose all participants in a MCN are rational. Therefore, they all make cost-efficient decisions. As a result, we always need appropriate rewards to incentivize more cooperative participants [10]; thus, we can build a satisfying MSN. According to the most prestigious work on social behaviors, such as [7, 8, 11–13], reciprocity mechanism leads to high cooperation rate in social networks. These works show that incentive and punishment play important roles in cooperation behaviors.

2. Contribution

With considerations of three influence factors and possible social dilemma mentioned in Section 1, our objective is to find out how and to what extent the cooperation rate is influenced and changes. Without loss of generality, we study the cooperation dynamics on the mobile crowds of unstructured population that locate, move, and communicate with each other within a fixed-size region. To highlight the influence from nodes' mobility, under the same population size and region size, we compare the cooperation dynamics on mobile crowds with that on the crowds of static spatial structures. Furthermore, to analyze the situation of more reality, we also explore the cooperation dynamics on the MCN that contains a proportion of special nodes, which is designedly placed into the crowd to promote or degrade the cooperation rate of the MCN. In this paper, we use Prisoner's

Dilemma Game (PDG) model to feature the cooperation behaviors in MCNs. Then, from an evolutionary game theoretic perspective, we construct an evolutionary PDG model to explore the relationships between the cooperation rate and the main influence factors on it. The proposed game model takes full account of the three types of influence factors. We think the quantitative findings are valuable for designing effective incentive mechanisms in MCNs.

Our main contributions in this work are as follows:

(i) We model the cooperation dynamics on MCNs of D2D communications into an evolutionary dynamic PDG model, in which smart devices are treated as rational game players/agents. To the best of our knowledge, this is a novel work that analyzes the cooperation dynamics on MCNs taking into consideration the above three categories of influence factors simultaneously. In our model, the device factor and the human will are featured by the parameterised payoff function and evolutionary rule. The agents (handheld smart devices) are supposed to be selfish and rational. Meanwhile, they also expect to construct reciprocity relationships with other agents. Therefore, the network factor is mainly featured by dynamic network topology and agents' mobility attributes. Comprehensive experiments and simulations on the unstructured crowd population show the cooperation rate of MCNs presents no-trivial changing when the influence factors change by time. The experimental results also quantitatively indicate the relationships between the cooperation rate and the main influence factors, including density of agents, communication range, temptation to defect, and mobility attributes.

(ii) We fully investigate the impact on the cooperation rate of MCNs from agents' mobility. The mobility attributes we mainly focus on in this paper include moving probability and velocity. Consider that the nodes in the MCN do not change their places all the time. We novelly introduce moving probability which can be viewed as the stability of the crowd and then systematically investigate the cooperation rate under different moving probability, speed, communication radius, and temptation to defect. We find too fast and too slow moving speed both lead to low cooperation rate by fixing other parameters. For a given crowd, it can be observed that there is an optimal average moving speed to achieve the highest cooperation rate. Furthermore, for a given average moving speed, the crowd's cooperation rate increases when the average moving probability increases from static situation; and the cooperation rate decreases when the average moving probability continuously increases from a certain average moving probability. There is an optimal moving probability for achieving the highest cooperation rate.

(iii) Based on our experiment results, we find that the cooperation rate of a mobile crowd is low when the crowd population is both too scattered and too dense. And the highest cooperation level is achieved with a moderate crowd density. We also find that the highest cooperation rate occurs under agent's moderate communication radius. In particular, when agents' communication radius is larger or smaller than the moderate value range of the given configuration, the cooperation rate of the mobile crowd declines significantly.

(iv) Simulations on the square plane indicate that the cooperation rate on a mobile crowd has no-trivial dependence on the individuals behaviors including moving probability, speed, communication radius, and temptation to defect.

3. Related Work

The formation of cooperative D2D communication between the mobile nodes is of great importance to an efficient cooperative wireless network, in which individuals share data with their neighbors to substantially boost the whole network performance. Numerous researches have been done to solve the design challenges of a D2D network, and many new approaches have been proposed to make the formation of a D2D network possible [1, 3, 4].

Nevertheless, except for the design challenges, since the fact that all the mobile devices are carried by different people who are irrelevant, a natural problem is whether the device holders are willing to cooperate when they communicate with their neighbors [14].

Given the fact of explosive growth of online social networks such as Facebook, Twitter, and Weibo, some researches novelly leverage the social relationship to promote a cooperative D2D network [5]. There are also some papers trying to investigate the cooperation rate in a Public Goods Game (HPGG) and Prisoner's Dilemma Game (PDG) modeled collaborative social networks (CSNs) [15, 16].

As we know, with the popularity of D2D likely communication model, more and more crowdsensing and crowdsourcing applications will be deployed in the flocking population. The realization of such apps is deeply dependent on the cooperation of the mobile nodes. Most existing works place their nodes on a given spatial structure and demonstrate appearance of cooperative nodes in such spatial structure from the view of evolutionary mechanism [17–19].

Recently, research about the statistical properties of human motion has attracted much attention. Compared with agents on some specific network topology, we think this meaningful extension is natural and more close to our real social network, given the fact that motion is a fundamental property of individuals and agents may change their place in a typical crowdsensing or crowdsourcing environment. Nevertheless, most researches about this just assume that each node will move all the time during the simulation [20–22]. In our paper, we assume that the nodes locate on a square plane and each node may move with a probability.

Prisoner's Dilemma (PD) game has long been one of the most popular game models to simulate the biological systems and human behavior [11, 21]. Like most researches about the CSN applications, we also use the PDG to model

the cooperation D2D dynamics on the networks. In a PDG modeled D2D communication, each node can choose to cooperate or defect simultaneously and then get their payoff. When a node is willing to sacrifice itself to transfer data for its neighbors, we say that it chooses to cooperate, and we call it a cooperator; otherwise, we consider it a defector. A defector means someone who only gets the data they need from the others and sacrifice nothing to help the others.

Such social dilemma is commonly observed throughout the human societies. The emergence and existence of cooperation remain an open question in biology and social science [12]. Tones of works have been done to study the showing and persistence of cooperation in a population composed of selfish individuals [23–25]. In a broad range of disciplines, evolutionary game theory is becoming one of the most fruitful frameworks to investigate this dilemma.

Since Nowak and May have first shown us that the evolutionary PD on a simple spatial structure can induce remarkable cooperation emergence [13], numerous researches have been done to understand the interplay between network topology and evolutionary game. Some natural behaviors in the real world have also been introduced at the same time and many interesting phenomena have been observed. We have already known that social relationship such as kin selection, direct reciprocity, indirect reciprocity, network reciprocity, and group selection has some potential to lead to cooperation [7]. In order to stimulate cooperation among D2D communications, some researches try to leverage human social relationship to solve the cooperation problem in the collaborating D2D network [5, 26]. Recently, research about the statistical properties of human motion has attracted much attention. Recently, many researches extended the research on evolutionary games to the systems consisting of mobile agents which are randomly located on a square plane and play games with agents around them [11].

4. Mobile Crowd Network Model

The mobile crowd network we study in this paper was formed by some moving nodes. The nodes meet, connect, and share resources with each other opportunistically. Each node is an autonomous and rational agent in the cooperations with others. Many features of the nodes affect their cooperation behaviors, mainly including crowd density, moving probability, speed, communication radius, and temptation to defect. For the easy understanding of the MCN model, we list some important notions as follows:

N: the number of players in a game.

L: the length of the square plane.

ρ: the density of the crowd.

$v_i(t)$: node i's moving speed at moment t.

$\theta_i(t)$: node i's moving direction at moment t.

f_c: crowd's cooperation rate.

p: moving probability.

v: moving speed.

$\text{pos}_i(t)$: node i's position in round t.

$\text{pof}_i(t)$: node i's payoff in round t.

$s_i(t)$: node i's strategy in round t.

$k_i(t)$: the number of neighbors of i at round t.

b: temptation to defect.

r: reaction radius.

4.1. Mobile Crowd Network. Suppose a mobile crowd consists of N nodes (or smart devices) moving within a $L \times L$ square plane with periodic boundary conditions. Initially, the N nodes randomly locate on the plane and randomly move. They meet and communicate with each other via D2D communication for reciprocal resource sharing. Here, *two nodes meet* means they both locate inside the other's communication coverage range. The momentary connections between the N nodes make them dynamically form a mobile crowd network, MCN.

Here, we use $\rho = N/(L \times L)$ to represent the density of the crowd. In the MCN, each node i ($i \in [1, \ldots, N]$) has a fixed identical communication radius r. It is indicated that the mean degree of all nodes is $\rho\pi r^2$. We use $v_i(t)$, and $\theta_i \in [0, 2\pi)$ denotes i's moving speed and direction at moment t, respectively. At any time, i can just contact its neighbor nodes, that is, the set of nodes locating within i's communication coverage. At time t, i's neighbor set is denoted as $w_i(t)$, where $w_i(t) = \{j \mid E(i, j) < r_i,\ j \in N,\ j \neq i\}$. Here, $E(i, j)$ represents the Euclidean distance between i and j.

In this MCN, the cooperation behaviors between nodes are constrained by nodes' D2D communication radius and their cooperation willingness. When $j \in w_i(t) \wedge i \notin w_j(t)$, we assume that i can just transfer data to j and j can not receive data from i. This means i and j form a unidirectional communication. When $j \in w_i(t) \wedge i \in w_j(t)$, i and j can transfer data to and receive data from each other based on their cooperation willingness, which is a bidirectional communication.

4.2. Prisoner's Dilemma Game on MCNs. The cooperation behaviors on the MCN described above can be modeled by a Prisoner's Dilemma Game (PDG), in which the N independent nodes are agents of the game. Each node plays PDG with its neighbors round by round with fixed period of time for a round when any two nodes meet each other. All nodes synchronously choose their strategies and compute their payoffs in a round.

A node autonomously makes decision of cooperation or defection that means sharing or not sharing resource to its neighbor nodes. Each node can just choose its strategy: cooperate (C) or defect (D). C strategy means sharing while D means not sharing. A node with a strategy C is a cooperator; otherwise, it is a defector. A cooperator will have to sacrifice its own resources; as a consequence, cooperator will create a benefit to each of its neighbors with its own resource consumption. On the contrary, a defector will not lose anything but only benefit from its cooperative neighbors. The payoff matrix of a PDG can be expressed as $\left(\begin{smallmatrix} R & S \\ T & P \end{smallmatrix}\right)$.

Two nodes meet and both get a payoff R for mutual *cooperate* and P for mutual *defect*. When two cooperators meet together, they will choose to sacrifice themselves to benefit each other, so they get the same payoff R equal to their benefits from cooperative neighbors minus their consumption by choosing to be a cooperator. However, two defectors will do nothing for each other and get no reception from each other, which means that their payoff $P = 0$. Things will change when a cooperator meets a defector; cooperator will choose to sacrifice itself to share its information with the defector, yet the defector will do nothing but only get the benefits from its cooperator neighbors. Thus, cooperator's payoff by playing with a defective neighbor S will be less than 0, because it will sacrifice its resources but get nothing from the defector; on the other hand, the defector's payoff T will be the largest, because it gets what it needs without sacrificing anything. It is easy to deduce that $T > R > P > S$ for all nodes. Therefore, defector always outperforms the cooperator since it will get more no matter what its neighbor is. However, when all nodes choose D strategy, they will all get nothing from others and the network is out of action. The situation that the whole network suffers from the problem of low cooperation rate is called social dilemma (the Prisoner's Dilemma Game model first illustrates the conflict of interests between what is best for the individual and what is best for the group and creates the social dilemma [27]).

To investigate the cooperation dynamics on MCNs, we model the long-term interactions among mobile nodes into an evolutionary PDG. At the beginning of the evolution process, each node randomly chooses its strategy (C or D) at its initial location. Each agent plays a PDG with their neighbors synchronously. In this multiround (or multigeneration) evolutionary game, one round is divided into two phases. At the first phase, all nodes compute and get payoffs according to their strategies. At the second phase, all nodes move and choose their next round strategies. Since a node locates at different position and meets different neighbors, its strategy changes in different round. It is obvious that the multiround game is different from iterated PDGs, since a node's game opponents are changing in different rounds.

By adopting the common practice of the configuration in evolutionary PDGs, such as the work [11, 21, 22], we let $T = b$, $R = 1$, and $P = S = 0$ in the payoff matrix. $T = b$ can be considered as the temptation to defect, $1 < b < 2$. Thus, we rescale the payoff matrix as $\begin{pmatrix} 1 & 0 \\ b & 0 \end{pmatrix}$.

In the interval from round t to $t + 1$, agent i moves with velocity $v_i(t)$ and direction $\theta_i(t)$. Here, $\theta_i(t)$ is a variable randomly chosen at the second phase of round t, $\theta_i(t) \in [0, 2\pi)$. This means that topology of the MCN will change irregularly. Different from most related work in mobile crowd/social networks, we novelly introduce a new parameter p to describe a node's moving probability realistically. When all nodes obtained their payoffs in round t, each node should choose its strategy for the next round $t + 1$. Let $\text{pof}_i(t)$ denote node i's payoff and $s_i(t)$ denote i's strategy in round t. In this evolutionary game, a node changes its strategy according to the following mechanism:

(1) Agent i does not need to change its strategy when i has no neighbor node in round t; that is, $s_i(t + 1) = s_i(t)$.

(2) Agent i randomly chooses a neighbor j and compares $\text{pof}_i(t)$ with $\text{pof}_j(t)$. If $\text{pof}_i(t) > \text{pof}_j(t)$, i will keep its strategy not changing in the next round; that is, $s_i(t + 1) = s_i(t)$.

(3) Agent i randomly chooses a neighbor j and compares $\text{pof}_i(t)$ with $\text{pof}_j(t)$. If $\text{pof}_i(t) \leq \text{pof}_j(t)$, i adopts j's t round strategy, that is, $s_i(t + 1) = s_j(t)$, with a probability $(\text{pof}_j(t) - \text{pof}_i(t))/(\max\{k_j(t), k_i(t)\} \cdot b)$. The probability is proportional to the payoff difference [22]. Here, $k_i(t)$ and $k_j(t)$ are the number of neighbors of i and j at round t.

It is notable that this mechanism is important for the multiround evolution process because a rational agent tends to change its strategy when it is not satisfied with its payoff in previous round. All agents carry out the strategy update process synchronously. In round t, the cooperation rate of the MCN is f_c, where $f_c = N_C/N$; here $N_C = \text{count}\{i \mid p_i(t) = C\}$ for $i = 1, \ldots, N$.

5. Experiments and Discussions

In this section, Using evolutionary PDG model, we carry out simulations on the MCN to investigate the cooperation rate changing by the moving probability (p), speed (v), radius (r), and temptation to defect (b). In our experiments, the MCN consists of 1000 mobile nodes. The crowd move on a 50×50 square. To avoid the border effects, we assume the square plane is toroidal. Under a given p, we give agents' radius different distributions. We also compare the cooperation dynamics of the mobile crowd with different radius distributions, including uniform, normal, exponential, and power-law. In order to figure out the influence of the others, we configured the other three parameters as identical features for all agents, and then changed them accordingly to get different cooperation dynamics.

Suppose the position of agent i at time (or round) t is $\text{pos}_i(t) = (x_i(t), y_i(t))$. At round t, $v_i(t)$ refers to agent i's moving speed and $\theta_i(t)$ means agent i's moving direction. Then, i's position at next round $t + 1$ is $\text{pos}_i(t + 1) = (x_i(t + 1), y_i(t + 1))$ which can be easily computed on a toroidal surface. To simplify the computing, we let time interval from round t to $t + 1$ be 1.

In the simulations, to guarantee accuracy, we collect data traces of the last 3,000 generations (rounds) from 50,000 generations. And each piece of data is an average of 100 runs under the same configuration.

5.1. Moving Probability with Different Moving Velocity. Moving probability determines the stability of the nodes and affects the cooperation between agents. To analyze the affection on cooperation rate (f_c) from agents' moving probabilities (p), in the experiment, we calculate the f_c by changing agents' moving probability p and fixing the other parameters, including velocity (v), radius (r), crowd density (ρ), and the temptation to defect (b). The result of this experiment is shown as Figure 1.

Compared with related work [11] which demonstrated that the cooperation rate can be enhanced under a moderate value of b and v, in Figure 1, we find the cooperation rate of the whole population is also greatly improved with a moderate

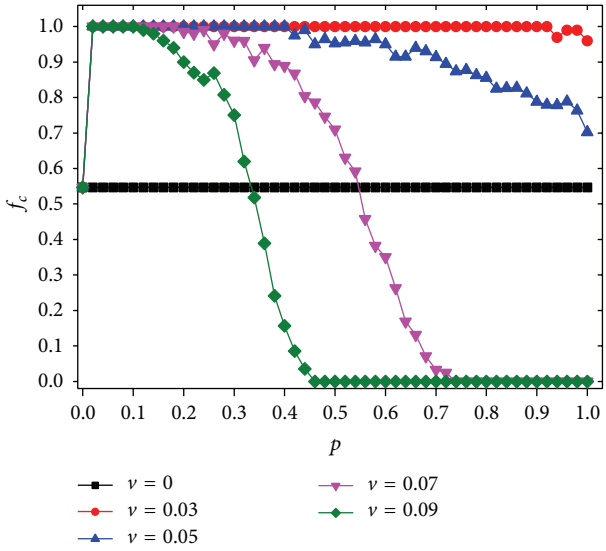


FIGURE 1: (Color online) The cooperation rate (f_c) under different agents' moving probability p and velocity (v). Here, we fix $L = 50$, $\rho = 1.3$, $b = 1.1$, and $r = 1$. The black line means the crowd is static ($v = 0$).

p and v. Compared with the static network model ($v = 0$) denoted by the black line in Figure 1, when letting crowd nodes move with some moderate probability, cooperation rate improves significantly in the crowd.

It is obvious that the cooperation rate decreases monotonously with p increasing, but is irrelevant to the changing of v. When p is fixed, we find that f_c is also decreasing while v is increasing. It is notable that the decline curve of different velocity ($v = 0.03$, $v = 0.05$, $v = 0.07$, and $v = 0.09$) has different amplitude. When v is equal to 0.03, f_c remains almost the same and only a tiny decrease can be observed when p is increasing to almost 1. This means the influence of p is limited with a moderate v. But when v comes to the larger situation, the influence of p appears, which is demonstrated by the curves of $v = 0.05$, $v = 0.07$, and $v = 0.09$. Looking at these curves, it can be found that the decline extent of f_c becomes quicker when agents' moving speed increases. And the dominance of mobility can be only observed in a relatively smaller region of p. Based on this discovery, we find that the influence of p is also partly determined by agents' moving speed. The influence on cooperation rate from agents' moving probability is more significant under a relatively larger moving speed.

In Figure 1, we also realize that when the whole mobile crowd move fast, a perfect cooperation network can be formed with a moderate moving probability, which means that individuals of the crowd are relatively stable. On the other hand, when the speed of the crowd is relatively slow, the cooperation rate achieves a high level even though the crowd is not stable enough, which means that the moving probability has relatively larger impact on cooperation rate than moving speed.

5.2. Crowd Evolutionary Dynamics. Capturing a series of snapshots from the evolutionary process, we find some major evolution characteristics of the mobile crowd network system. Figure 2 shows the snapshots at eight times for a simulation with $v = 0.07$, which indicates how a mobile crowd evolve to a full cooperation network with some favorable conditions. At time 0 [Figure 2(a)], both cooperator [green dots] and defector [red dots] players are randomly located on the squared plane, with the same fraction (0.5). Because agents move with a probability, the number of connections for each player will change. Thus, it is essentially different from the static status. Not too long after the initial state, Figure 2(b) shows the state of the system at $t = 100$. We can see a quick decrease of the cooperators. Most individuals turn to a defector for a better payoff, except for a small cluster of cooperators on the top right corner of the square.

Many existing related works had revealed that the cooperators can enforce their success only by forming clusters in a mobile environment [21]. In our simulation, one can also find that the small cooperator cluster becomes expanding as time grows. From Figures 2(c)–2(h), we can see the cooperators slowly expand to the whole square based on the small cooperator region. Then, at about 3000 time steps after $t = 20000$, the system finally evolves to a cooperative network.

Figure 2 has shown how cooperators form clusters and then attract the defect nodes that are lying around the boundary to ensure the success of cooperation. In order to qualitatively explain the results generated in Figure 1, we evaluate the mean payoffs of cooperators and defectors lying around the boundary when $v = 0.07$. Because all nodes are rational, their strategies will change according to how much payoff they can receive. As Figure 3 shows, the average payoff of the defectors (P_d) around the boundary is 0 when the moving probability (p) is less than 0.38; on the contrary, the payoffs of the cooperators (P_c) on the boundary are at the highest level. So, at this region of p, we can know that defectors have no place to live ($f_c = 0$), while cooperators flourish in the crowd and get their highest payoff. When p is bigger than 0.38, P_c begins to decrease and P_d begins to increase. We are able to see the decreasing of cooperators and the defectors appear and keep increasing. Interestingly, when p is more than 0.62, both P_c and P_d tend to decline. This may be owing to the lasting growth of the defectors. When the number of defectors is big enough, then defectors will have more chances to communicate with the ones that are also defectors. Compared with the case that most nodes lying around the defective one are cooperator, as a matter of fact, this will definitely lower their payoffs. Then, at $p = 0.74$, we can find both cooperators and defectors get a payoff of 0, which means the crowd falls into a defection state, and defectors can never gain payoffs by exploiting their cooperator neighbors.

5.3. Moving Probability under Various Defect Temptation. In order to analyze the role of moving probability, given the condition that the temptation to defect (b) is changing, Figure 4 shows us the cooperation level curves of our simulation results for (f_c) as a function of the temptation to defect b. Each curve indicates different moving probabilities (p), here

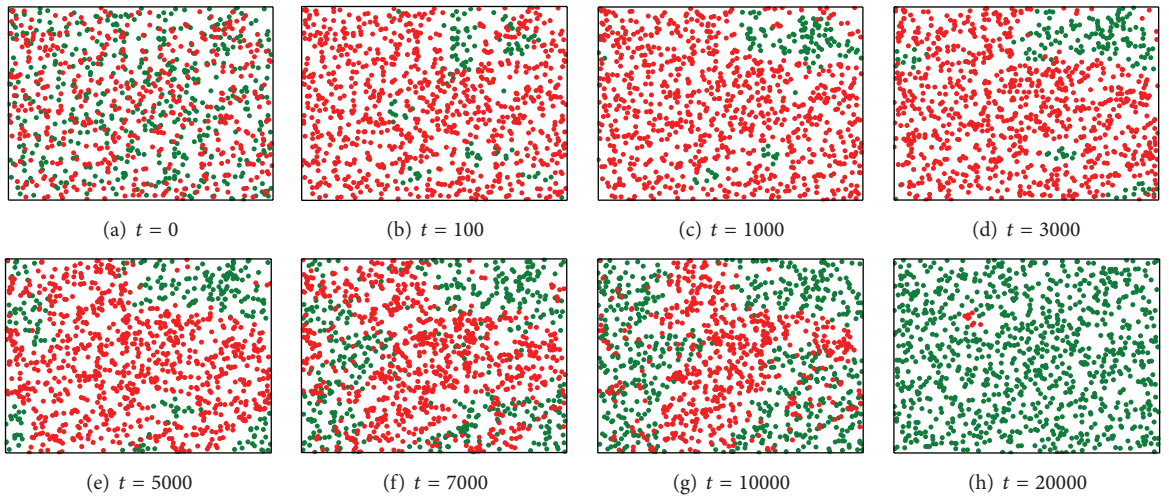


FIGURE 2: (Color online) Spatiotemporal evolution of a mobile crowd (each agent moves with a probability) where $p = 0.1$. This specific simulation has been performed for a population in which $N = 1000$, $\rho = 1.3$, $b = 1.1$, and $v = 0.03$. Cooperators are denoted by green (light gray) and defectors are denoted by red (dark gray) dots correspondingly. Each picture depicts a snapshot of different time of evolution process.

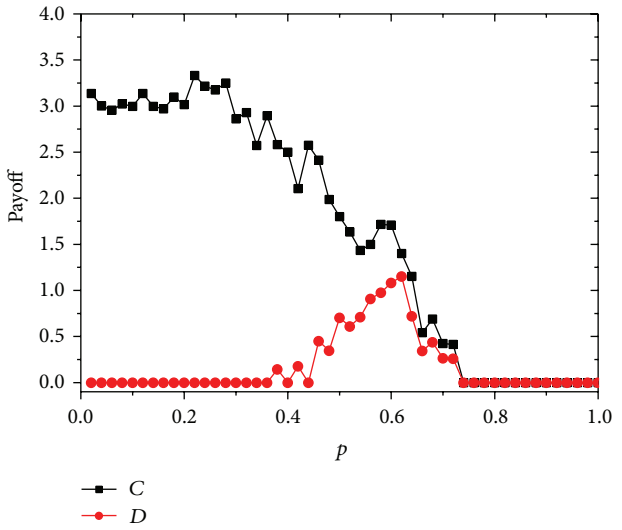


FIGURE 3: The mean payoffs of cooperators and defectors who are lying around the boundary. Here, we fix $L = 50$, $\rho = 1.3$, $b = 1.1$, $r = 1$, and $v = 0.07$.

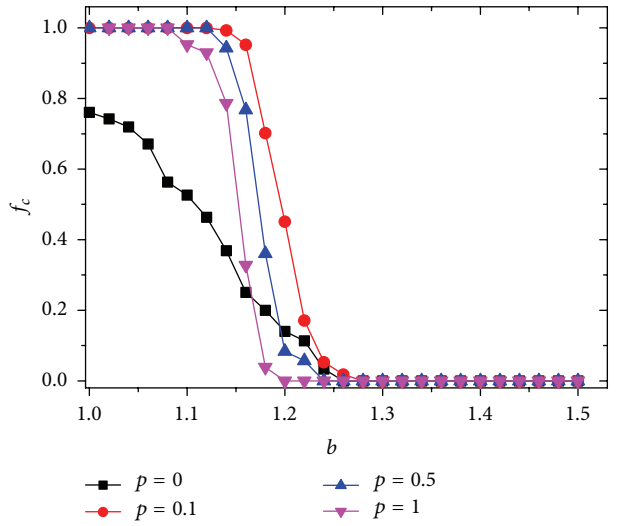


FIGURE 4: The relationship between the cooperator level (f_c) and temptation to defect (b) under various values of p. Black line indicates a static crowd ($p = 0$). Here, we set $N = 1000$, $\rho = 1.3$, and $v = 0.03$.

we have four curves indicated as $p = 0$, $p = 0.1$, $p = 0.5$, and $p = 0.9$. Not surprising, like most classic studies about the relationship between defect temptation and f_c in the evolutionary game theory, it is notable that f_c decreases monotonously with b increasing up to a threshold where cooperation vanishes, in both static and dynamic status. This is reasonable because larger b means stronger temptation to defect. All nodes of the crowd will tend to defect when the temptation is big enough. But we can find various threshold under different p. When p is not 0, it is clear that the cooperation level decreases while the moving probability increases. By fixing b, we find that f_c is inversely proportional to the moving probability (p). When b is relatively small (less than about 1.3), f_c is dramatically promoted when individuals choose to move with some probability. As we see in Figure 3, f_c for $p = 0.1$ always dominates f_c for the static status ($p = 0$). But, for a higher moving probability ($p = 0.5$, $p = 0.9$), compared with the static status, cooperation rate is promoted in a relatively smaller region of b. It is about 0–1.19 when $p = 0.5$ and about 0–1.16 when $p = 0.9$.

From Figure 4, we find that when given a moderate temptation to defect, a relatively stable (p is small) crowd helps to form a cooperation network. We can also find that f_c declines when the mobile crowd system becomes more erratic. It is also important to notice that a relatively small p

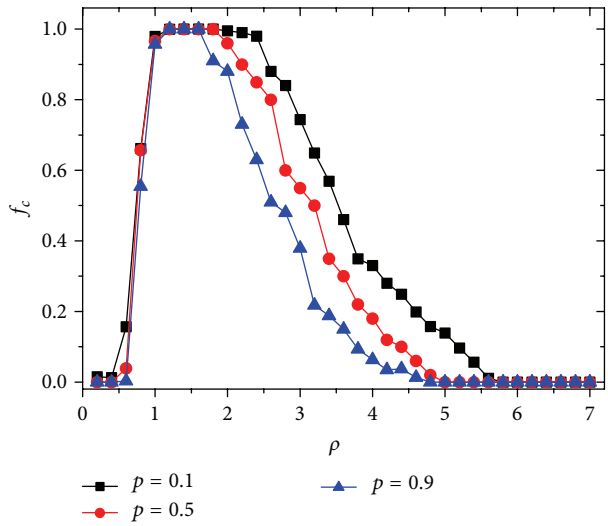


FIGURE 5: (Color online) The cooperator frequency (f_c) versus crowd density (ρ) under different moving probability (p). Here, we fix $L = 50$, $v = 0.03$, $b = 1.1$, and $r = 1$.

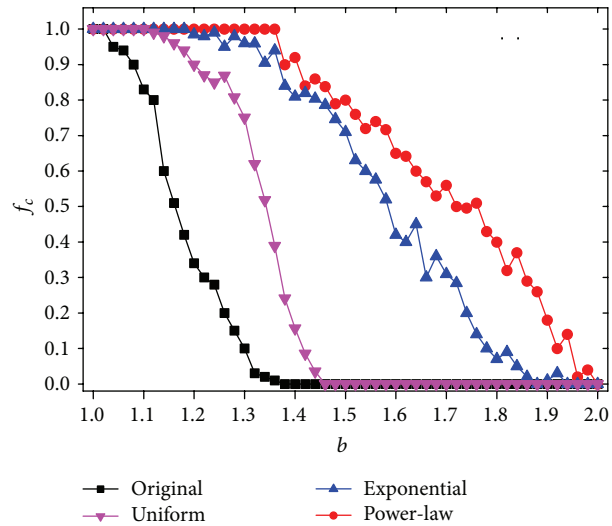


FIGURE 6: The cooperator frequency (f_c) versus r with different distributions. Here, we fix $L = 50$, $\rho = 1.3$, $v = 0.03$, and $p = 0.5$.

means a larger region of b that can make sure the cooperation is successful.

5.4. Crowd Density and Communication Radius. Similar to most related work, it is interesting to investigate the effects of ρ (crowd density), r (node's communication radius), and v (moving speed), respectively, on the evolution of cooperation after we give the node a probability to move.

With fixed r and v, we investigate the effect of ρ on three different moving probabilities (p) in Figure 5. In order to get various ρ, we change the crowd population N. Obviously, smaller N means lower crowd density. Compared with the work [11, 21, 25], we find an ideal region of the density of the players which is favorable to form a cooperation crowd under three different p. From Figure 4, we can see that the cooperation level (f_c) tends to be low when the population is both too spread (ρ is small) and too dense (ρ is large) in three different p. For this reason, these two conditions are hard for cooperators to form clusters, the only mechanism that can enforce their success. When it comes to different p, the figure shows that different p obtains a different ideal density region. Obviously, $p = 0.1$ gets a biggest range of ideal density area that is good for the evolution of cooperation. Interestingly, when fixing ρ, smaller moving probability ($p = 0.1$) also dominates the larger ones ($p = 0.5$ and $p = 0.9$). It indicates that a relatively smaller individual's moving probability can get a more cooperative network. At the same time, when ρ is too large or small, the crowd can only get a full defective result, which means f_c is equal to 0.

Each node only communicates with the nodes within its radius, which means node's radius (r) has an important role in the evolutionary game dynamic. Figure 6 shows mobile crowd's cooperation dynamics with four different view radii when b is changing. It is obvious that f_c monotonically decreases with the increase of b. However, different radius distributions can induce different cooperation levels. And we

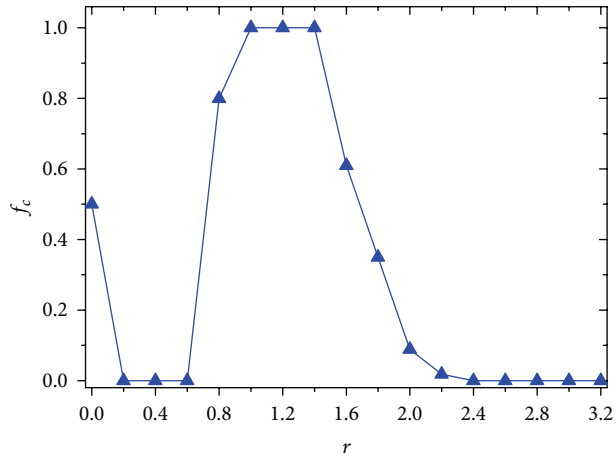


FIGURE 7: The cooperator frequency (f_c) versus r. Here, we fix $L = 50$, $\rho = 1.3$, $b = 1.1$, $v = 0.03$, and $p = 0.5$.

can sort it out easily from Figure 6. We can know that crowd will get a lowest cooperation level when all the nodes have the same radius. However, the cooperation rate will be improved remarkably when the crowd has a more heterogeneous distribution. We can see the power-law distribution, which is the most heterogeneous one, gets the highest cooperation level. Previous studies have revealed that radii have some nontrivial connection with nodes' moving property [21]. From Figure 6, we infer that the influence of different radius distribution still exists under a moving probability p.

In Figure 7, we fix other parameters in order to investigate the effect of the view radius changing. We can find that the highest cooperator frequency can be obtained under some moderate values of r (r is within 1.0–1.6). When r is larger or smaller than that value region, we observe an apparent decline of the crowd's cooperation level. This is in accordance with [21, 22]. Evidently, nodes have fewer interaction neighbors with a smaller r, and cooperators

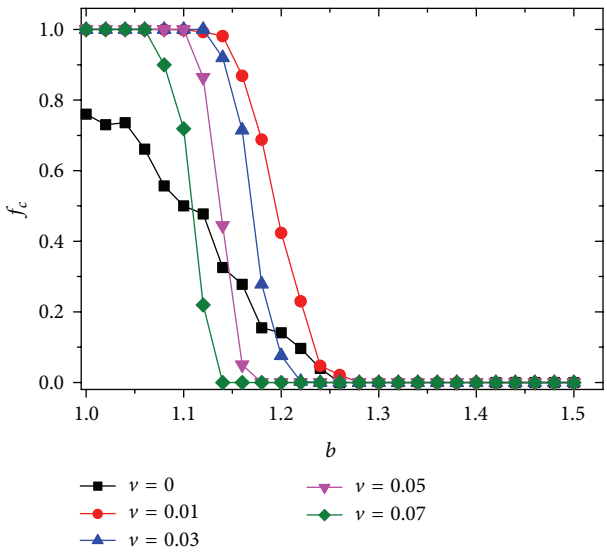


FIGURE 8: The relationship between the cooperator level (f_c) and temptation to defect (b) under various values of v. Black line indicates a static crowd ($p = 0$). Here, we set $N = 1000$, $\rho = 1.3$, and $p = 0.5$.

cannot form clusters to resist the invasion of defectors. On the contrary, with a large r, nodes of the crowd tend to be mixed enough; more or less everyone interacts with each other, so we inevitably have a low cooperation level.

5.5. Moving Speed. Next, assume node will move with a probability ($p = 0.5$), we fix other parameters to investigate the effect of the velocity v in Figure 8. As in Figure 4 even with different speed, cooperation level declines when we increase the temptation of defect (b). Compared with the static case ($v = 0$, $p = 0$), it is worth noting that cooperation is greatly enhanced when players are allowed to move with a low velocity (e.g., $v = 0.01$). Besides, different moving speed can induce different cooperation levels, and one can easily sort out the cooperation levels: $v = 0.01 > v = 0.03 > v = 0.05 > v = 0.07$. Similar to Figure 4, we also observe that a low velocity can improve the cooperator in a larger region of b, indicating that lower velocity can tend to be of more benefit to the survival of cooperator.

6. Conclusion

In this paper, we study the cooperation dynamics on a typical mobile crowd network of D2D communication. Using evolutionary game theory, our simulation and analysis on the cooperative behaviors of mobile users take a deep insight into the cooperation promotion in such a dynamical network with selfish autonomous users. The experiment results show that mobile user's features, including speed, moving probability, and reaction radius, have an obvious influence on the formation of a cooperative MCN. We also found some optimal status when the crowd's cooperation rate reaches the best. (1) The crowd can reach a good cooperation rate with a moderate moving speed and probability, which is no more than 0.5; (2) we found the best reaction radius of a node in MCN, which is about 0.8 to 1.5 in our simulation; (3) the ideal crowd is also shown in our experiment, which is about 1 to 3; (4) the crowd can research a higher cooperation rate when the temptation to defect is higher, which is reasonable. These regularities are useful for a network designer to design a MCN with good performance.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by China NSF Grants 61472365, 61472364, and 61572435 and ZJNSF Grant LZ16F020001.

References

[1] G. Fodor, E. Dahlman, G. Mildh et al., "Design aspects of network assisted device-to-device communications," *IEEE Communications Magazine*, vol. 50, no. 3, pp. 170–177, 2012.

[2] A. Antonopoulos, E. Kartsakli, and C. Verikoukis, "Game theoretic D2D content dissemination in 4G cellular networks," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 125–132, 2014.

[3] S. Shalmashi and S. Ben Slimane, "Cooperative device-to-device communications in the downlink of cellular networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '14)*, pp. 2265–2270, Istanbul, Turkey, April 2014.

[4] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 96–104, 2012.

[5] X. Chen, B. Proulx, X. Gong, and J. Zhang, "Social trust and social reciprocity based cooperative D2D communications," in *Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '13)*, pp. 187–196, ACM, Bangalore, India, August 2013.

[6] I.-R. Chen, J. Guo, F. Bao, and J.-H. Cho, "Trust management in mobile ad hoc networks for bias minimization and application performance maximization," *Ad Hoc Networks*, vol. 19, pp. 59–74, 2014.

[7] M. A. Nowak, "Five rules for the evolution of cooperation," *Science*, vol. 314, no. 5805, pp. 1560–1563, 2006.

[8] D. R. Amor and J. Fort, "Effects of punishment in a mobile population playing the prisoner's dilemma game," *Physical Review E*, vol. 84, no. 6, Article ID 066115, 2011.

[9] Z. Han, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, Cambridge University Press, Cambridge, UK, 2012.

[10] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: a quality based incentive mechanism for crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '15)*, pp. 177–186, ACM, Hangzhou, China, June 2015.

[11] S. Meloni, A. Buscarino, L. Fortuna et al., "Effects of mobility in a population of prisoner's dilemma players," *Physical Review E*, vol. 79, no. 6, Article ID 067101, 2009.

[12] M. A. Nowak, *Evolutionary Dynamics*, Harvard University Press, 2006.

[13] M. A. Nowak and R. M. May, "Evolutionary games and spatial chaos," *Nature*, vol. 359, no. 6398, pp. 826–829, 1992.

[14] C. P. Roca and D. Helbing, "Emergence of social cohesion in a model society of greedy, mobile individuals," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 28, pp. 11370–11374, 2011.

[15] C. Hauert, S. D. Monte, J. Hofbauer, and K. Sigmund, "Volunteering as Red Queen mechanism for cooperation in public goods games," *Science*, vol. 296, no. 5570, pp. 1129–1132, 2002.

[16] G. Wei, P. Zhu, A. V. Vasilakos, Y. Mao, J. Luo, and Y. Ling, "Cooperation dynamics on collaborative social networks of heterogeneous population," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1135–1146, 2013.

[17] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device communication," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 136–144, 2014.

[18] B. Di, T. Wang, L. Song, and Z. Han, "Incentive mechanism for collaborative smartphone sensing using overlapping coalition formation games," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '13)*, pp. 1705–1710, Atlanta, Ga, USA, December 2013.

[19] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *The Journal of Supercomputing*, vol. 54, no. 2, pp. 252–269, 2010.

[20] H. Lin, D.-P. Yang, and J.-W. Shuai, "Cooperation among mobile individuals with payoff expectations in the spatial prisoner's dilemma game," *Chaos, Solitons and Fractals*, vol. 44, no. 1–3, pp. 153–159, 2011.

[21] J. Zhang, W.-Y. Wang, W.-B. Du, and X.-B. Cao, "Evolution of cooperation among mobile agents with heterogenous view radii," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 12, pp. 2251–2257, 2011.

[22] Y.-T. Lin, H.-X. Yang, Z.-X. Wu, and B.-H. Wang, "Promotion of cooperation by aspiration-induced migration," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 1, pp. 77–82, 2011.

[23] F. C. Santos, M. D. Santos, and J. M. Pacheco, "Social diversity promotes the emergence of cooperation in public goods games," *Nature*, vol. 454, no. 7201, pp. 213–216, 2008.

[24] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak, "A simple rule for the evolution of cooperation on graphs and social networks," *Nature*, vol. 441, no. 7092, pp. 502–505, 2006.

[25] Z.-X. Wu and P. Holme, "Effects of strategy-migration direction and noise in the evolutionary spatial prisoner's dilemma," *Physical Review E*, vol. 80, no. 2, Article ID 026108, 2009.

[26] F. Wu, K. Gong, T. Zhang, G. Chen, and C. Qiao, "COMO: a game-theoretic approach for joint multirate opportunistic routing and forwarding in non-cooperative wireless networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 948–959, 2015.

[27] A. Rapoport and A. M. Chammah, *Prisoner's Dilemma: A Study in Conflict and Cooperation*, University of Michigan Press, Ann Arbor, Mich, USA, 1965.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 7908328, 11 pages
http://dx.doi.org/10.1155/2016/7908328

Research Article

ODMBP: Behavior Forwarding for Multiple Property Destinations in Mobile Social Networks

Jia Xu,[1,2,3,4] Jin Xin Xiang,[1] Xiang Chen,[1] Fang Bin Liu,[4] and Jing Jie Yu[4]

[1]*School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Jiangsu, Nanjing 210003, China*
[2]*Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Jiangsu, Nanjing 210003, China*
[3]*Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Jiangsu, Nanjing 210003, China*
[4]*Department of Information Technology, Nanjing General Hospital of Nanjing Military Command, Jiangsu, Nanjing 210002, China*

Correspondence should be addressed to Jia Xu; xujia@njupt.edu.cn

Received 2 November 2015; Accepted 3 January 2016

Academic Editor: Tingting Chen

The smartphones are widely available in recent years. Wireless networks and personalized mobile devices are deeply integrated and embedded in our lives. The behavior based forwarding has become a new transmission paradigm for supporting many novel applications. However, the commodities, services, and individuals usually have multiple properties of their interests and behaviors. In this paper, we profile these multiple properties and propose an Opportunistic Dissemination Protocol based on Multiple Behavior Profile, ODMBP, in mobile social networks. We first map the interest space to the behavior space and extract the multiple behavior profiles from the behavior space. Then, we propose the correlation computing model based on the principle of BM25 to calculate the correlation metric of multiple behavior profiles. The correlation metric is used to forward the message to the users who are more similar to the target in our protocol. ODMBP consists of three stages: user initialization, gradient ascent, and group spread. Through extensive simulations, we demonstrate that the proposed multiple behavior profile and correlation computing model are correct and efficient. Compared to other classical routing protocols, ODMBP can significantly improve the performance in the aspect of delivery ratio, delay, and overhead ratio.

1. Introduction

In recent years, the smartphones have increased rapidly. According to the data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, the vendors shipped a total of 334.4 million smartphones worldwide in the first quarter of 2015 [1]. Wireless mobile networks are evolving and integrating with many aspects of our lives since we can read news, watch videos, listen to music, communicate with others, send and receive emails, browse and search the web, share contents to Internet and trade online, and so forth, through the smartphones conveniently. The wide spread smartphones promote the combination of the online social network and the mobile smart terminals, accelerating the development of the mobile social network (MSN) [2]. MSN involves the interactions between participants with similar interests and objectives through their mobile devices within virtual social networks.

Due to the dynamic and volatile nature of MSN, opportunistic networks operate under a completely new networking paradigm where traditional routing protocols cannot be applied [2]. Opportunistic networks are wireless mobile self-organizing networks in which the topology is extremely dynamic and unstable. Thus, in most cases, there might not exist the complete link from the source to the destination simultaneously. There have been many research efforts on opportunistic forwarding. However, most of them deliver the message based on IP or device address, which is not effective in many interest-aware or behavior-aware MSN applications.

The unprecedented tight coupling between mobile devices and their users provides new approaches to infer users' behavior and interest from mobile devices. The mobile devices can now act as distributed behavioral sensors of users to capture their interests and enable implicit interest profiling [3].

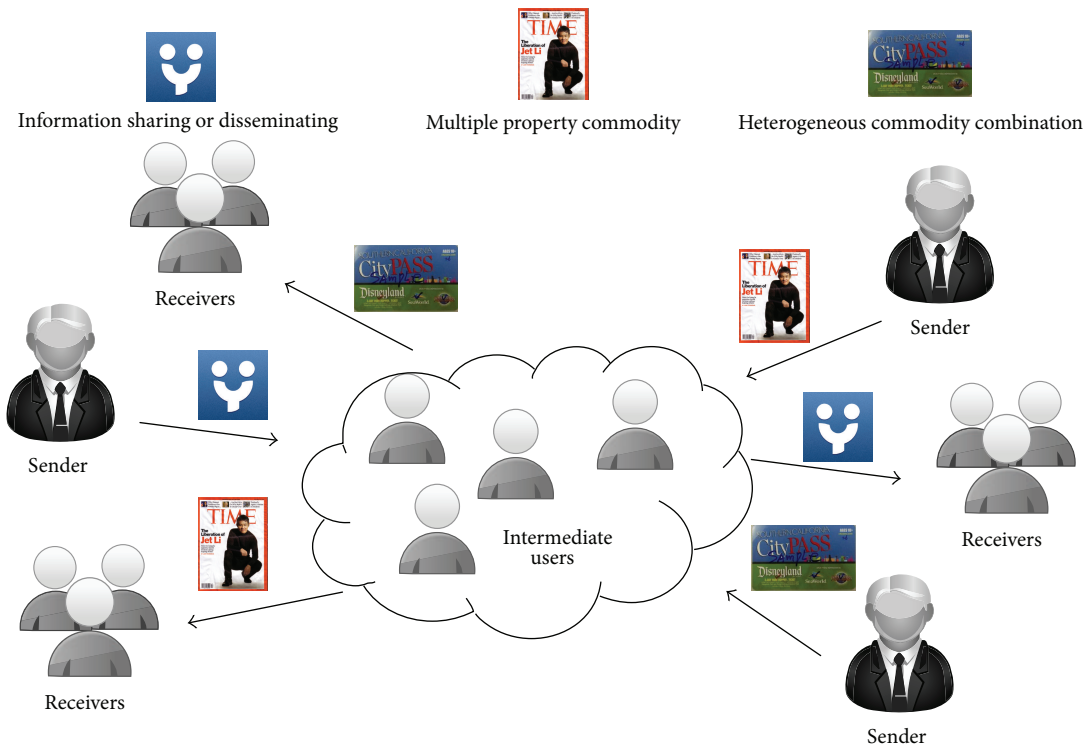


Figure 1: Multiple interests oriented opportunistic dissemination.

There are many popular location based applications in MSN. For instance, the location based services can help mobile users to find friends who are currently in their vicinity. Another example is Contact Recommendation Mechanism [4], which can efficiently select contacts in order to address them as a social group, so as to ease the initialization of group interactions.

The basic idea of these applications is extracting the interest profiles or relationship profiles from the social behavior. However, few research efforts consider the multiple behavior properties of MSN users. In fact, many MSN applications deal with the objects with multiple properties. Moreover, the human being has multiple interests naturally. There are some types of typical scenarios:

(i) sharing or disseminating the information to the people with similar multiple interest profiles: as an example, Bob is a student, and he wants to find a roommate who is in the same university. He also hopes that the roommate likes swimming, just like himself. Now, Bob wants to push the message to the persons who have the great possibility to be the roommate;

(ii) recommendation of commodities or services with multiple properties: for example, the editorial office wants to recommend a new magazine, which includes multiple topics, such as pop music, clothing, and bodybuilding. The editorial staff need to disseminate the advertisement to the potential readers who are interested in most of the topics;

(iii) recommendation of the combination of heterogeneous commodities or services: for example, the merchant wants to publicize a discount combo of films and snacks and to send the information to the people who like both of them.

In the aforementioned scenarios, the message sender has a set of target interests, which can represent the commodities, services, or individuals with multiple properties. As shown in Figure 1, the message sender wants to send the message to the receivers who have the same or similar interests to the target interests.

In this paper, we focus on extracting multiple behavior properties from the daily traces of users and exploring the Opportunistic Dissemination Protocol based on the Multiple Behavior Profile in mobile social network. The key contributions of our work are summarized as follows.

(i) We aim to deal with the data dissemination in a class of ubiquitous application scenarios, where the multiple properties of objects or the multiple interests of people need to be considered.

(ii) We map the multiple properties or the multiple interests to the behavior space and profile the multiple behavior properties. Moreover, we propose the correlation computing model based on the principle of BM25 [5] for multiple behavior profiles.

(iii) We design an Opportunistic Dissemination Protocol based on Multiple Behavior Profile (ODMBP) in mobile social networks.

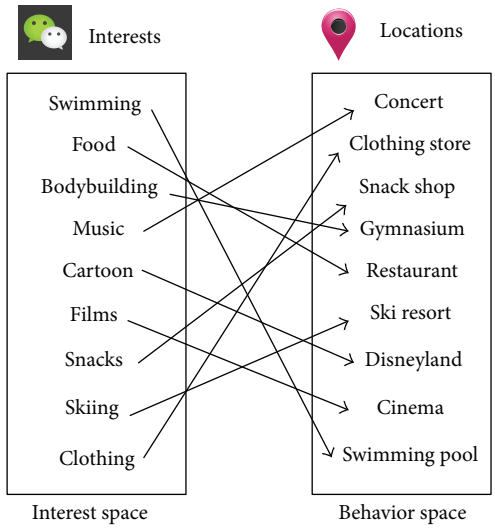


Figure 2: Mapping the interest space to the behavior space.

(iv) The extensive simulations show that the proposed multiple behavior profiles and correlation computing model are correct and efficient. Compared to other classical routing protocols, ODMBP achieves high delivery ratio and low delay in the scenarios of multiple property data dissemination.

The remainder of this paper is organized as follows. Section 2 presents the challenges and the rationale of designed protocol. Section 3 introduces the multiple behavior profile and the correlation computing model. We present our opportunistic dissemination protocol in Section 4. The performance evaluation is presented in Section 5. We review the related work in Section 6. We conclude the paper in Section 7.

2. Challenges and Rationale

There are some challenges to design the opportunistic dissemination protocol for multiple property objects. First, we need to give a computable expression of interests, from which the multiple properties can be obtained. Second, for multiple properties, we need to consider the number of matched properties; that is, the designed protocol should try its best to match more properties in appointed properties. Thus, it might be inefficient to summate the similarity value of each property straightforwardly. Moreover, due to the energy limited devices and the intermittent link of opportunistic network, the designed protocol should meet the desired properties of distributed design, low computation complexity, low overhead, and high expandability.

As can be seen in Figure 2, many interests are closely related to the individual's daily trace and can be represented by the specific locations in the trace. It is shown in [6] that social relationships can explain about 10% to 30% of all human movements based on an analysis of different kinds of location datasets. On the other side, a large body of research has demonstrated that people show striking persistence in their mobility profiles. For example, in [7], the authors state that the similarity of the mobility profile of a given user to its future profile is high, above 0.75 for eight days, and remains above 0.6 for five weeks. The observations demonstrate that the mobility profile is indeed an intrinsic property and a valid representation of the user, even if only a short history of mobility profile is used. Therefore, in this work, we assume that the locations can represent the user's interests; moreover, the longer the time in one location is, the stronger the corresponding interest is.

The basic idea of ODMBP is mapping the interest space to the behavior space and extracting the similarity between users' multiple behavior profiles and the target profiles. ODMBP uses the locations and the corresponding time spent at the locations to reflect users' preferences. The multiple behavior profile of each user is extracted from the quantized behavior space. Further, a reasonable correlation computing model should be applied to calculate the correlation metric. The designed opportunistic dissemination protocol then takes the correlation metric based forwarding strategy as the basic principle.

3. Multiple Behavior Profile and Correlation Computing Model

In this section, we will introduce the multiple behavior profile and the correlation computing model. The behavior profile should reflect multiple behavior properties, respectively, according to multiple interests. The correlation computing model should quantize the correlation for each user and should match as many behaviors as possible in the set of appointed behaviors.

3.1. Multiple Behavior Profile. Assume that there are a set $U = \{1, 2, \ldots, n\}$ of users and a set $L = \{1, 2, \ldots, m\}$ of locations, where $n \geq 2$ and $m \geq 2$. Each location j in behavior space represents the corresponding interest. Each user $i \in U$ has a user multiple behavior profile $\mathrm{UMBP}_i = (t_{i1}, t_{i2}, \ldots, t_{im})$, where t_{ij}, $j \in L$, is the total time that user i spent at location j.

Note that t_{ij} is a cumulative time based on current trace of user i, and the value would be changed when time goes on. The time that user spent at the specific location can be measured through different ways. A widely used method is sensing the location information continuously through GPS sensors, which are integrated universally in the smartphones. Alternatively, the connection log of WiFi or switches in specific location can also help to obtain the time that the user spent. This work does not involve the specific persistent sensing, and the energy consumption can be very low.

The user multiple behavior profiles can be expressed as $\mathrm{UMBP} = (\mathrm{UMBP}_1, \mathrm{UMBP}_2, \ldots, \mathrm{UMBP}_n)$. It can be viewed as $n \times m$ behavior matrix with element t_{ij}. An example of UMBP is given in Figure 3. In most cases, the behavior matrix is a sparse matrix since most users only stay at a small fraction of all m locations. Thus, some specific data structures such as triple table can be used to reduce the space and time

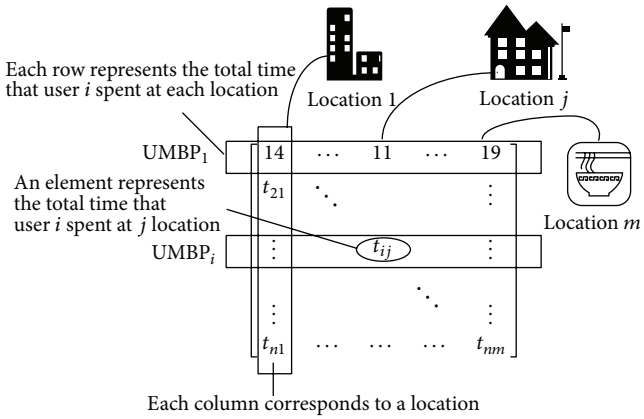


Figure 3: Illustration of user multiple behavior profiles.

complexity. Each element t_{ij} in UMBP is associated with a behavior indicator x_{ij}, where

$$x_{ij} = \begin{cases} 1, & \text{if } t_{ij} \neq 0 \\ 0, & \text{if } t_{ij} = 0. \end{cases} \tag{1}$$

There is a target multiple behavior profile TMBP $= (tx_1, tx_2, \ldots, tx_m)$ for each specific data dissemination application, where $tx_j = 1$, $1 \leq j \leq m$, if the message sender hopes that the receivers have the behavior property associated with location j; else, $tx_j = 0$. We assume that TMBP, which can be obtained through mapping the interests to the corresponding locations, is known in advance. We further denote the number of behavior properties in TMBP as $r = \sum_{j=1}^{m} tx_j$.

3.2. Correlation Computing Model. To find the potential receivers, we need a computing model to calculate the correlation between the target multiple behavior profile and user i's multiple behavior profile, $1 \leq i \leq n$. The correlation is quantized by metric $\text{Score}_i(\text{TMBP}, \text{UMBP})$. We use the principle of the ranking function, named BM25 [8], to calculate this metric. BM25 uses the ideas of Robertson-Sparck-Jones (RSJ) probability model [9] and is a ranking function used by search engines to rank matching documents according to their relevance to a given search query. So far, it is the most successful model for calculating the correlation [10–12].

We first define the behavior factor of user i to location j in TMBP as

$$\text{BF}_{ij} = \frac{t_{ij}(K+1)}{t_{ij} + K}, \quad \text{s.t. } 1 \leq i \leq n,\ 1 \leq j \leq m,\ j = \arg\limits_{1 \leq k \leq m} tx_k = 1, \tag{2}$$

where K is an empirical parameter, which represents the importance of behavior factor in $\text{Score}_i(\text{TMBP}, \text{UMBP})$.

The behavior factor BF_{ij} measures the total time user i spent at location j in TMBP. BF_{ij} provides a basic correlation evaluation. However, it might not meet the requirement of matching as many behaviors as possible in TMBP since the behavior factor does not consider the distinction of user distribution at different locations. Actually, there might be some locations where few people stay in general. Thus, the behavior factor, which only considers the time the user spent at the location, might lose sight of these sparsely populated locations. To balance the sparsely populated locations, we introduce w_j, the weight of location j,

$$w_j = \log \frac{n - q_j + 0.5}{q_j + 0.5}, \tag{3}$$

where $q_j = \sum_{i=1}^{n} x_{ij}$ is the number of users at location j.

Note that the greater the value of q_j is, the smaller the value of w_j is. The weight of location j reflects the distinction of user distribution at different locations and can promote the importance of sparsely populated locations in TMBP.

Now we can give the ultimate formula to calculate the correlation metric:

$$\text{Score}_i(\text{TMBP}, \text{UMBP}) = \sum_{j=1}^{m} w_j \times \text{BF}_{ij}, \quad \text{s.t. } 1 \leq i \leq n,\ 1 \leq j \leq m,\ j = \arg\limits_{1 \leq k \leq m} tx_k = 1. \tag{4}$$

Note that the Cosine similarity [13, 14] is another method to compute this metric as well. The Cosine similarity is widely used for computing the similarity of the text. It is not difficult to use Cosine similarity based on our user multiple behavior profiles. However, the Cosine similarity does not consider the distinction of user distribution at different locations. Further analyses and evaluation will be given in Section 5.

4. Opportunistic Dissemination Protocol Design

In this section, we attempt to design an opportunistic dissemination protocol for the services with multiple property objects. According to the principle of small world [15], people have high clustering property, and the users with similar behavior property have high probability of encounter. ODMBP disseminates the messages based on users' multiple behavior profiles and corresponding correlation metric.

The UMBP of each user will change with elapsed time, and it should be updated in distributed way. In ODMBP, each user i stores UMBP in his mobile device. UMBP_i can be updated by itself through position sensor or network connection log, while UMBP_j, $j \neq i$, will be updated when user i encounters user j.

As shown in Algorithm 1, ODMBP consists of three stages: user initialization, gradient ascent, and group spread. In the user initialization stage, for each encountered user j, the message sender i matches the UMBP_j with the unmatched target multiple behavior profile TMBP'. If there is at least one matched location k, that is, $t_{jk} = tx'_k = 1$, the message sender i sends the message to user j. Once all locations in target multiple behavior profile are matched, that is, $\text{TMBP}' = 0$, the message sender i deletes the message. By this way, the user

```
(1) TMBP′ ← TMBP;
(2) foreach j encountered do
(3)   update UMBP for i and j;
(4)   if i is a message sender then
(5)     if TMBP′ ≠ 0 then // Stage 1: User Initialization
(6)         foreach k ∈ {1, 2, ..., m} do
(7)             if t_jk ≠ 0 and tx′_k ≠ 0 then
(8)                 send message to j;
(9)                 tx′_k ← 0;
(10)                break;
(11)    else
(12)        delete message in i;
(13)  else if δ > Score_j(TMBP, UMBP) > Score_i(TMBP, UMBP) then // Stage 2: Gradient Ascent
(14)    send message to j;
(15)    delete message in i;
(16)  else if Score_j(TMBP, UMBP) > Score_i(TMBP, UMBP) ≥ δ then // Stage 3: Group Spread
(17)    send message to j;
```

Algorithm 1: ODMBP.

initialization stage can parallelize the dissemination process and decrease the delay efficiently.

Then, in the gradient ascent stage, the message holder forwards the message to the users with higher correlation score. The gradient ascent stage is derived from the fact that the multiple behavior profiles of the users with higher correlation score are more similar to the target receivers.

In the group spread stage, if the correlation score of the message holder is higher than threshold δ, where δ is a parameter of ODMBP, the message holder copies the message to the users with higher correlation score. This means ODMBP considers all user i satisfying $\text{Score}_i(\text{TMBP}, \text{UMBP}) \geq \delta$ as the receivers.

5. Performance Evaluation

5.1. Methodology and Settings. In this section, we conduct thorough simulations to investigate the performance of ODMBP. We use the real trace dataset StudentLife [16], which contains the sensor data, EMA data, survey responses, and educational data. For our simulations, we adopted a part of this dataset, named Wifi-Location, which contains the data of 49 volunteers moved around 92 buildings in Dartmouth College within a month. The Wifi-Location, which contains nearly 0.192 million mobility records, acquires WiFi AP deployment information from Dartmouth Network Services and records participants' on-campus rough locations and unix time stamp. As an example, the record (1364359102, in (Kemeny)) indicates a volunteer moved in the building called Kemeny at the unix time 1364359102. The buildings can be seen as the locations in UMBPs and TMBP. We removed the interference items in the real movement trace such as the duplicate data and the invalid users. Figure 4 describes the number of locations of each user of the processed data and this number mostly falls in the interval [5, 100].

All the simulations were run on ONE simulator [17]; it is an opportunistic network environment simulator which provides a powerful tool for generating mobility traces, running

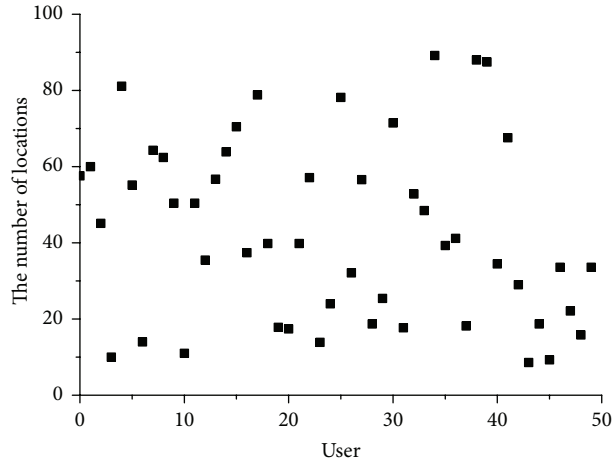


Figure 4: Distribution of location number.

DTN messaging simulations with different routing protocols. All the results are averaged over 1000 runs. The settings of the ONE simulator have been listed in Table 1. We first integrate the continuous records with the same location into a new record in order to compute their time difference. We also need to remove some interference items such as duplicate data and invalid user. We take this final output results as the external events connection data for the simulator. The number of hosts and the number of locations are 49 and 92, respectively, which are equal to the number of volunteers and buildings in Wifi-Location trace.

We use the time-location pairs to structure UMBP_i of any user i. As the simulator time goes on, the time spent in specific location can be obtained through calculating the elapsed time from the time stamp of user i's current mobility record. By this way, we can obtain UMBP_i through accumulating all such elapsed time for each location. The UMBP is privacy information for each user and is calculated and updated dynamically with the simulator time. Moreover, the

Table 1: Simulation parameter setting.

Parameter	Value
Scenario name	ODMBP/ODMBP-Cos/EP/S & W
Number of hosts	49
Message time to live (Message TTL)	500 s
Simulation connection	False
Buffer size of nodes	1 M
Movement model warm-up time	1000 s
Simulation duration	960000 s
Number of locations	92
External events connection	True

users can connect with the users who are in the same location simultaneously. So, we can structure the external event connection data dynamically for the ONE simulator based on the above processing method. In our simulations, the behavior locations in TMBP are selected randomly among 92 buildings.

In our simulations, we first reveal the impacts of the key parameters on delivery ratio and delay of ODMBP. Moreover, we evaluate ODMBP further by comparing it with other protocols: Epidemic routing [18], Spray and Wait [19], and ODMBP-Cos.

To explore the differences between Cosine and BM25, we apply the Cosine similarity to our system model, and the correlation score function based on Cosine similarity is

$$\text{CosSim}_i\,(\text{TMBP}, \text{UMBP}) = \frac{x_{ij} \cdot tx_k}{\left\|x_{ij}\right\| \times \left\|tx_k\right\|}, \quad \text{s.t. } 1 \le i \le n,\ 1 \le j \le m,\ j = \underset{1 \le k \le m}{\arg}\, tx_k = 1, \tag{5}$$

where $\cdot$ is the vector product and $\|x\|$ is the Euclidean norm of x; that is, $\sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}$.

We substitute the correlation score function with CosSim (TMBP, UMBP) in stage two and stage three of Algorithm 1, respectively. We call the protocol using Cosine similarity as ODMBP-Cos.

5.2. Revealing the Impacts of the Key Parameters. There are three key parameters: the empirical parameter K, the number of behavior properties in target multiple behavior profile r, and the threshold δ. We will vary them for exploring the impacts of these parameters, respectively.

5.2.1. Impact of K. Based on our correlation computing model, K represents the importance of behavior factor in ultimate score metric. When using BM25 model in searching, K usually gets the value of 1.2 based on past experience. However, this setting might not be applicable in our multiple behavior dissemination scenarios. For the purpose of revealing the impact of K on ODMBP, we measure the delivery ratio and delay of ODMBP with different value of K when setting $r = 5$. As shown in Figure 5, ODMBP gets the best delivery ratio and delay when $K = 1.75$. Based on the observation, we fix $K = 1.75$ in the following simulations. However, the setting of K may be closely related to the real dataset adopted.

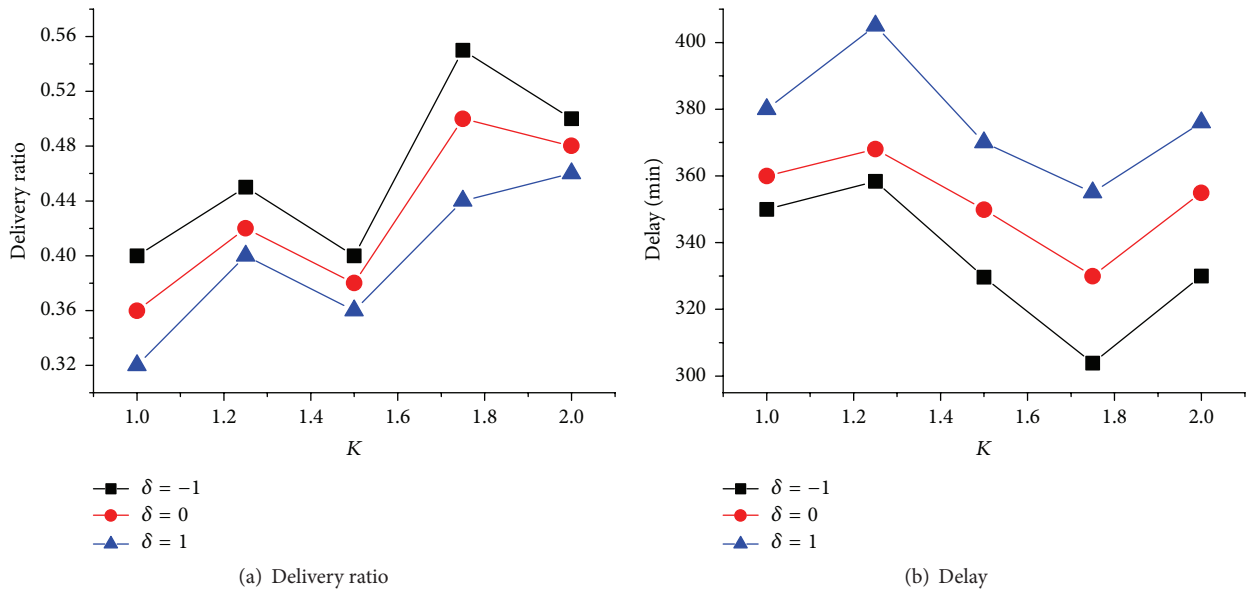
5.2.2. Impact of δ. Threshold δ is a criterion to judge whether the user is a receiver. It is also the trigger of ODMBP to enter the group spread stage. We measure the performance of ODMBP with different δ. Figure 6 shows three groups of results corresponding to $r = 2$, $r = 6$, and $r = 10$, respectively. When the value of δ goes on, the delivery ratio decreases drastically for all settings of r. This is because the number of receivers reduces when the threshold increases. Accordingly, it takes more time to find the receivers, and the delay increases.

5.2.3. Impact of r. The number of behavior properties in target multiple behavior profile r, which is provided in advance, indicates the comprehensiveness of commodities/services or people's versatility. We cannot adjust the value of r to improve the performance of ODMBP; however, we can evaluate the scalability of designed protocol through the observation of the impact of r on ODMBP. We can see from Figure 7 that the curves of delivery ratio are not monotonous. Based on formula (4), $\text{Score}_i(\text{TMBP}, \text{UMBP}) = \sum_{j=1}^{m} w_j \times \text{BF}_{ij}$; thus, the score is a summation value for all behavior locations in TMBP. Note that $w_j = \log((n - q_j + 0.5)/(q_j + 0.5))$, and the value of w_j will be negative if $q_j > n/2$. Thus, the value of score might decrease with great value of r. As a result, the number of receivers would reduce. As can be seen from Figure 6, ODMBP achieves the best performance in the aspect of delivery ratio and delay when $r = 6$ among all measured r in our simulations.

5.3. Compare with Other Protocols. We compare ODMBP with other classical routing protocols, Epidemic routing and Spray and Wait in opportunistic network. In Epidemic routing protocol, the message is delivered to each encountered node that does not have the same message. The Spray and Wait routing protocol consists of two phases: Spray and Wait. The message copies are forwarded to l different nodes in the spray phase, and then the direct transmission is performed in the wait phase. We set $l = 6$ and apply binary mode in the spray phase of Spray and Wait routing protocol. We set $r = 6$, $K = 1.75$, and $\delta = 0$ for ODMBP. We also compare the performance of ODMBP and ODMBP-Cos. We set $r = 6$, $\delta = 0.7$ in order to obtain the best performance of ODMBP-Cos. Such settings are based on the similar measures in Section 5.2.

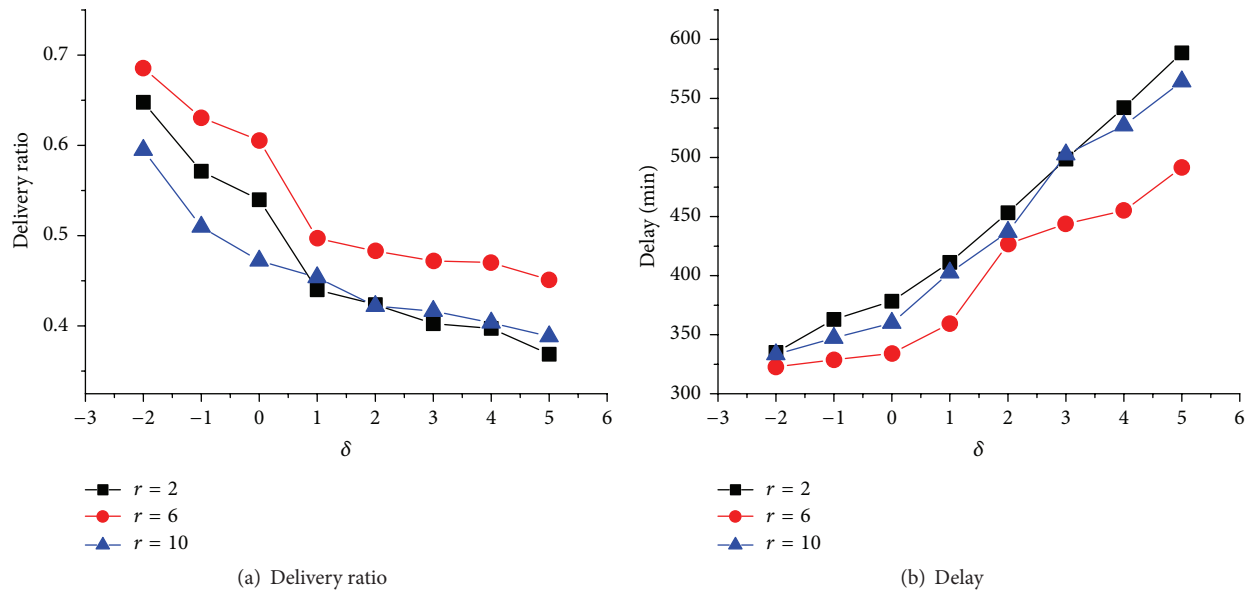
As shown in Figure 8, ODMBP has higher delivery ratio compared with ODMBP-Cos. This is because there are some locations where few people stay in general. Thus, the correlation function based on Cosine similarity, which only considers the time spent at the location, might lose sight of these sparsely populated locations. However, ODMBP can balance it well. On the other hand, the delay performance of two protocols is close.

The delivery ratio increases with increasing message TTL for all four protocols. This is because there is more time to



(a) Delivery ratio (b) Delay

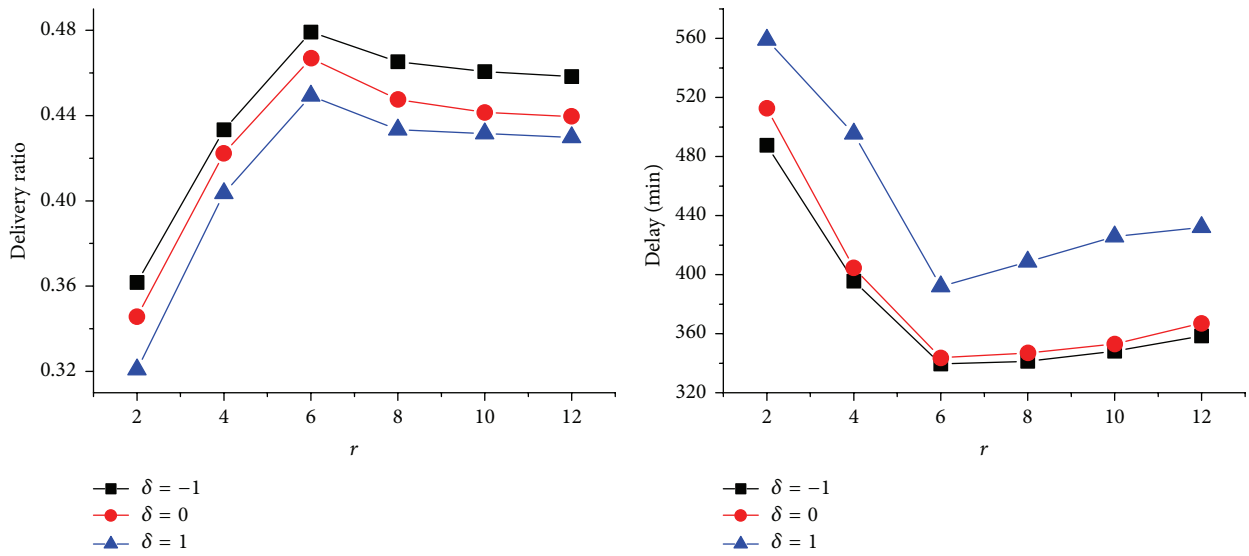
Figure 5: Impact of K.



(a) Delivery ratio (b) Delay

Figure 6: Impact of δ.

deliver the message to the receivers before dropping it in the forwarding queue. However, the delay increases when the message TTL increases. Epidemic routing achieves the best performance among four protocols; however, it will suffer high overhead and is not efficient in our mobile social network applications. This is because Epidemic routing does not provide filtering scheme in the dissemination. The ONE simulator defines the parameter overhead ratio (number of relayed messages − number of delivered messages)/number of delivered messages, while ODMBP has the threshold to filtrate the user with different correlations. Thus, ODMBP can reduce the amount of relayed messages. As shown in Figure 8, ODMBP has lower overhead ratio than Epidemic routing. In most cases, the performance of ODMBP is better than Spray and Wait, and ODMBP improves 11.6% and 12.5% in the aspect of delivery ratio and delay, respectively, on average.



(a) Delivery ratio (b) Delay

Figure 7: Impact of r.

This is because ODMBP can forward the message to the users who are more similar to the target, while Spray and Wait does not consider the correlation metric.

6. Related Work

At present, there are many studies on exploring the behavior attributes of users in mobile social networks. In [7], Hsu et al. established a user behavior oriented communication model through the analysis of the participators' mobile data in university, demonstrating that the user has high stability in his mobile attribute. On this basis, they presented a protocol for the profile-cast service with high transmissibility and low delay, named CSI. However, only single behavior attribute is considered in their designed protocol. InterestCast [20], a novel communication protocol, also considers users' interests, solving the problem for a wide range of social scenarios and applying to an opportunistic network where nodes are the personal devices of moving individuals, possibly interacting with fixed road-side devices. In [21], Zhao et al. study a new coverage problem, opportunistic coverage, to characterize the sensing quality of such people-centric sensing systems. Compared with the traditional static coverage and dynamic coverage in sensor networks, opportunistic coverage has some unique characteristics caused by the requirements of urban sensing applications and human mobility features. The interest-aware implicit multicast (iCast) [22] is a new casting paradigm and works based on the inferred interest profiles. In this paradigm, messages are sent to a behavioral interest profile (not to an IP or device address). It combines user's interest and behavior for multicast communication. In [23], Elsherief et al. explore the notion of mobile users' similarity as a key enabler of innovative applications hinging on opportunistic mobile encounters. SANE [24] combines the advantages of both social-aware and stateless approaches. It is based on the observation that individuals with similar interests tend to meet more often. In [25], Matsuo et al. propose an efficient boundary detection method in dense mobile wireless sensor networks. Each node preliminarily recognizes locations of itself and all its neighboring nodes. The authors determine the node forwarding direction by comparing the similarity score with the encounter nodes. Cheng et al. present iZone [26], a mobile social networking system based on the analysis of general requirements of MSN and location based services (LBS). The ultimate goal is developing and establishing an integrated framework for providing social network based healthcare information services targeting patient safety, empowerment, and guidance. Besides, [27] explains the interaction relationship of social network users and mutual influence and social network privacy behavior characteristics and motivation, including the prediction of user behavior as well.

7. Conclusion

We have extracted the multiple behavior profiles from the users' daily trace through mapping the multiple properties in the interest space to the behavior space. The BM25 based correlation computing model was proposed to calculate the correlation metric of multiple behavior profiles. Moreover, we have proposed an Opportunistic Dissemination Protocol based on Multiple Behavior Profile termed ODMBP, in mobile social networks. It consists of three stages: user initialization, gradient ascent, and group spread. Through extensive simulations, we have demonstrated that the proposed multiple behavior profiles and correlation computing model are

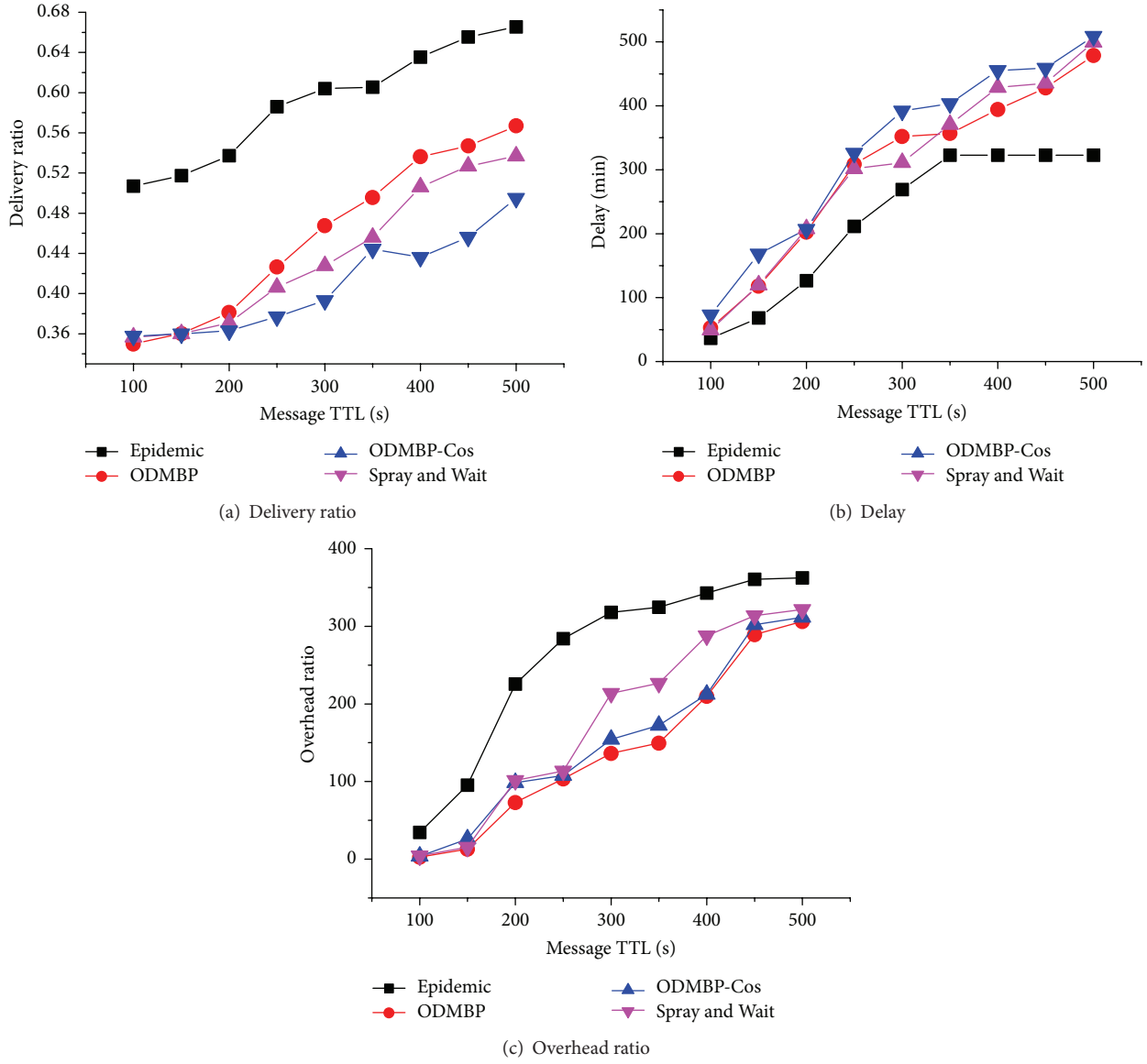


Figure 8: Compare ODMBP with other protocols.

efficient. Compared to the other classical routing protocols, ODMBP can significantly improve the performance in the aspect of delivery ratio and delay.

In the future work, we will consider more complex scenarios. For example, the behavior locations in the target multiple behavior profile can be associated with specific weights, which indicate the importance of the behavior locations.

Notations

U, n: Set of users and the number of users
L, m: Set of locations and the number of locations
TMBP, r: Target multiple behavior profile and the number of behavior properties in target multiple behavior profile
UMBP, UMBP$_i$: User multiple behavior profiles and user i's user multiple behavior profile
x_{ij}: Behavior indicator of user i to location j
tx_j: Target behavior indicator of location j
w_j: Weight of location j
BF$_{ij}$: Behavior factor of user i to location j

t_{ij}: Total time that user i spent at location j
q_j: Total number of users at location j
K: A parameter used for computing the behavior factor
δ: A threshold used for control of entering the group spread stage
$\text{Score}_i(\text{TMBP}, \text{UMBP})$: The correlation metric between the target multiple behavior profile and user i's multiple behavior profile.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is sponsored in part by NSFC Grants (nos. 61472193, 61472192, and 61373139), The Natural Science Foundation of Jiangsu Province (nos. BK20141429, BK20130852), Scientific and Technological Support Project (Society) of Jiangsu Province (no. BE2013666), CCF-Tencent Open Research Fund (no. CCF-Tencent RAGR20150107), China Postdoctoral Science Foundation (no. 2014M562662, 2013T60553), Jiangsu Postdoctoral Science Foundation (no. 1402223C), Independent Research Project of Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks (no. WSNLBZY201524), NUPTSF (Grant no. NY215098), and the "1311" Talent Project of NJUPT.

References

[1] The worldwide smartphone market compared to the previous quarter's unprecedented results, 2015, http://www.idc.com/prodserv/smartphone-market-share.jsp.

[2] X. Hu, T. H. S. Chu, V. C. M. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. B. Chan, "A survey on mobile social networks: applications, platforms, system architectures, and future research directions," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1557–1581, 2014.

[3] S. Moghaddam and A. Helmy, "Interest-aware implicit multicast (iCast): opportunistic mobile data dissemination without per-group management," in *Proceedings of the 9th ACM MobiCom Workshop on Mobility in the Evolving Internet Architecture (MobiArch '14)*, pp. 55–60, ACM, Maui, Hawaii, USA, September 2014.

[4] R. Grob, M. Kuhn, R. Wattenhofer, and M. Wirz, "Cluestr: mobile social networking for enhanced group communication," in *Proceedings of ACM International Conference on Supporting Group Work (GROUP '09)*, pp. 81–90, Sanibel Island, Fla, USA, May 2009.

[5] R. Blanco and P. Boldi, "Extending BM25 with multiple query operators," in *Proceedings of the 35th Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*, pp. 921–930, August 2012.

[6] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 1082–1090, San Diego, Calif, USA, August 2011.

[7] W.-J. Hsu, D. Dutta, and A. Helmy, "CSI: a paradigm for behavior-oriented profile-cast services in mobile networks," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1586–1602, 2012.

[8] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[9] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *Journal of the American Society for Information Science*, vol. 27, no. 3, pp. 129–146, 1976.

[10] M. Murata, H. Nagano, R. Mukai, K. Kashino, and S. Satoh, "BM25 with exponential IDF for instance search," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1690–1699, 2014.

[11] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau, "Okapi at TREC4," in *Proceedings of the 4th Text REtrieval Conference (TREC '96)*, pp. 21–30, Vienna, Va, USA, May 1996.

[12] S. Robertson, "On the history of evaluation in IR," *Journal of Information Science*, vol. 34, no. 4, pp. 439–456, 2008.

[13] H. Yildirim and M. S. Krishnamoorthy, "A random walk method for alleviating the sparsity problem in collaborative filtering," in *Proceedings of the ACM Conference on Recommender Systems (RecSys '08)*, pp. 131–138, ACM, Lausanne, Switzerland, October 2008.

[14] C. Cattuto, D. Benz, A. Hotho, and G. Stumme, "Semantic analysis of tag similarity measures in collaborative tagging systems," in *Proceedings of the 3rd Workshop on Ontology Learning and Population (OLP3 '08)*, pp. 39–43, Patras, Greece, July 2002.

[15] A. Banerjee, R. Agarwal, V. Gauthier, C. K. Yeo, H. Afifi, and F. B.-S. Lee, "A self-organization framework for wireless ad hoc networks as small worlds," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2659–2673, 2012.

[16] R. Wang, F. L. Chen, Z. Y. Chen et al., "StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp '14)*, pp. 3–14, Seattle, Wash, USA, September 2014.

[17] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *Proceedings of the 1st ACM SIGMOBILE Workshop on Mobility Models*, pp. 33–40, ACM, Hong Kong, May 2008.

[18] A. Vahdat and D. Becker, *Epidemic routing for partially-connected ad hoc networks [M.S. thesis]*, 2000.

[19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05)*, pp. 252–259, ACM, August 2005.

[20] E. Pagani, L. Valerio, and G. P. Rossi, "Weak social ties improve content delivery in behavior-aware opportunistic networks," *Ad Hoc Networks*, vol. 25, pp. 314–329, 2015.

[21] D. Zhao, H. D. Ma, and L. Liu, "Energy-efficient opportunistic coverage for people-centric urban sensing," *Wireless Networks*, vol. 20, no. 6, pp. 1461–1476, 2014.

[22] S. Moghaddam and A. Helmy, "Interest-aware implicit multicast (iCast): opportunistic mobile data dissemination without per-group management," in *Proceedings of the 9th ACM Workshop*

on Mobility in the Evolving Internet Architecture (MobiArch '14), pp. 55–60, ACM, Maui, Hawaii, USA, September 2014.

[23] M. Elsherief, T. Elbatt, A. Zahran, and A. Helmy, "The quest for user similarity in mobile societies," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS '14)*, pp. 569–574, Budapest, Hungary, March 2014.

[24] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless routingin pocket switched networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 252–261, 2014.

[25] K. Matsuo, K. Goto, A. Kanzaki, T. Hara, and S. Nishio, "Overhearing-based efficient boundary detection in dense mobile wireless sensor networks," in *Proceedings of the 15th IEEE International Conference on Mobile Data Management (IEEE MDM '14)*, pp. 225–234, IEEE, Queensland, Australia, July 2014.

[26] R. Cheng, Z. Yang, and F. Xia, "IZone: a location-based mobile social networking system," in *Proceedings of the 3rd International Symposium on Parallel Architectures, Algorithms and Programming (PAAP '10)*, pp. 33–38, Dalian, China, December 2010.

[27] A. Acquisti, L. Brandimarte, and G. Loewenstein, "Privacy and human behavior in the age of information," *Science*, vol. 347, no. 6221, pp. 509–514, 2015.