Journal of Computer and Knowledge Engineering, Vol. 1, No. 1.
DOI: 10.22067/cke.v1i2.60404

Collusion-resistant Worker Selection in Social Crowdsensing Systems

Masood Niazi Torshiz[✦], Haleh Amintoosi

Abstract. The main idea behind social crowdsensing is to leverage social friends as crowdworkers to participate in crowdsensing tasks. A main challenge, however, is the identification and recruitment of well-suited workers. This becomes especially more challenging for large-scale online social networks with potential sparseness of the friendship network which may result in recruiting participants who are not in direct friendship relations with the requester. Such recruitment may increase the possibility of collusion among participants, thus threatening the application security and affecting data quality. In this paper, we propose a collusion-resistant worker selection method which aims to prevent the selection of colluders as suitable participants. For each participant who is considered to be selected as suitable, the proposed method is aimed to prevent any possible collusion. To do so, it determines whether the selection of a new participant may result in the formation of a colluding group among the selected participants. This has been achieved through leveraging the Frequent Itemset Mining technique and defining a set of collusion behavioral indicators. Simulation results demonstrate the efficacy of our proposed collusion prevention method in terms of selecting efficient collusion indicators and detecting the colluding groups.

Keywords: worker selection; collusion; data quality.

1.Introduction

1.1. Background

The widespread prevalence of mobile computing devices such as sensor-rich smartphones has propelled the emergence of a novel crowdsourcing [1] paradigm, known as mobile crowdsensing or participatory sensing [2]. In mobile crowdsensing, ordinary citizens volunteer to use their mobile phones for collecting sensor data from their nearby environment. The aim of such sensor data gathering methods include computing the aggregate statistics about a phenomenon, thus increasing the global awareness of the issues of interest. A plethora of applications have been recently proposed based on this revolutionary paradigm, ranging from personal health [3, 4] and prices of customer goods [5] to environmental monitoring such as road conditions [6] and noise pollution [7].

The involvement of people in the sensing process, however, brings about new challenges. Accepting to contribute to a task will inherently require that the worker devotes some time and effort towards it. Moreover, collecting and uploading the sensor data consumes the mobile phone battery and communication bandwidth. Most importantly, engaging in such crowdsourcing activities may lead to potential privacy threats such as the disclosure of home/work address or private conversations [8, 9, 10]. With all these in mind, a participant may be hesitant to contribute to a sensing task. This may result in a lack of adequate number of workers, which in turn may compromise the fidelity of the obtained information and ultimately render the application to be not very useful.

One potential solution to address this challenge is to leverage online social networks (constituting hundreds of millions of subscribers with various skills and expertise) as the underlying publish-subscribe infrastructure for crowdsensing applications [11, 12]. In a typical social crowdsensing system, social network members can act as service requesters and utilize social friends and friends-of-friends as crowdworkers to contribute to their tasks. A pertinent example of such a system is Jelly[1] which is built on top of existing social networks like Facebook[2] and Twitter[3]. When the users encounter something unusual, they can take a picture of the object, formulate a query and submit it to their social network. Another instantiation of the concept of social participatory sensing is found in [13] where Twitter is used as the underlying social network substrate. The authors proposed two mobile applications: (i)

Manuscript received November 21, 2016; accepted January 4, 2017.

Masood Niazi Torshiz, Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.
Haleh Amintoosi, Department of Computer Engineering, Engineering Faculty, Ferdowsi University of Mashhad, Mashhad, Iran.
[✦]The corresponding author's e-mail is: niazi@mshdiau.ac.ir

[1] http://blog.jelly.co/post/72563498393/introducing-jelly
[2] http://facebook.com
3 http://twitter.com

a weather radar application in which, Twitter members send tweets indicating the weather condition and (ii) a noise-mapping application where members gather sound samples via their mobile phones and contribute the noise level via Twitter.

1.2. Problem Statement

One of the important challenges in the success of social crowdsensing is the selection of suitable participants as crowdworkers. Leveraging well-suited participants is bound to increase the quality of obtained contributions, since suitable participants have better knowledge and expertise relevant to the task requirements.

In the context of mobile crowdsensing, tasks are normally location-based (i.e. contributions should be collected from a specific place) and are to be completed within a specific time period. As such, the suitability of the participant for a campaign is typically related to the participant's geographical and temporal availability as well as the participant's reputation [14]. The geographical and temporal availability is extracted from collecting and analysing the time-stamped location traces of the participants. The reputation of the participant is measured based on the quality of the contributions of the participant in the past.

In social crowdsensing, the existence of public profile information of participants (who are social network members as well), and the social links between them adds new dimensions to the evaluation of a participant's suitability. Through public profile information, access to the participant's interests, expertise and domain specific knowledge is possible. Moreover, the participant's social reputability can be derived from his social relations and interactions. These valuable pieces of information can be used to identify well-suited participants, and hence, overcome the challenge of suitability.

Another important issue that should be considered when evaluating the suitability of participants is the likelihood of their involvement in collusive groups. A group of malicious participants might form a colluding group in such a way that they are recruited in preference to other potentially high-quality workers. The colluding group would then have the power to sway the outcome of the task in accordance with their agenda. So, it is important to identify potentially collusive members and prevent them from being selected as suitable participants.

In order to prevent collusion in mobile crowdsensing, a series of works [15, 16] utilise a trusted platform module (TPM) [17]. TPM is a micro-controller provided with each sensor device to attest the integrity of sensor readings. This local integrity checking makes the system resistant to collusion. However, TPM chips are yet to be widely adopted in mobile devices. In other research that eschews TPM, such as [18, 19], the collusion detection is achieved by leveraging reputation management systems and outlier detection algorithms. The aim is to identify and revoke the colluders by investigating their behaviour and assigning a low reputation score to them.

In the context of social crowdsensing, the existence of social ties between members facilitates the formation of colluding groups. Colluders can easily communicate via the social network communication facilities. They are also able to establish social communities by creating groups in the social network and manage collusive attacks by collaboratively contributing to a series of tasks. They can also easily share their corrupted contributions with other group members and hence propagate the bias. So, selecting the participants in such a way that the probability of collusion among the selected members is very low is important for achieving high quality contributions.

1.3 Contributions and Outline

In this paper, we propose a collusion-resistant worker selection method, which is aimed at preventing the selection of colluding members as suitable crowdworkers. In other words, we intend to identify whether the addition of each new participant to the previously selected group will result in the formation of a group of colluders within the selected participants.

Colluders are like-minded people who collaborate with each other on a specific agenda to obtain an objective by defrauding or gaining an unfair advantage. Their objective may be earning monetary or non-monetary profits. Colluders usually form a group which is large enough to make a considerable impact [20]. Moreover, group members usually target a considerable number of tasks and collaborate together in contributing to these tasks. Their contributions are typically similar to each other (in order to overwhelm the task with similar faulty contributions) and deviate from the other (genuine) participants (so as to change the task's outcome). Finally, the colluders may prefer to connect with each other in the form of social groups to facilitate their communications. Based on these collaborative behaviours, the collusion prevention method considers the following collusion indicators: (i) group size (i.e. number of colluders), (ii) group target size (i.e. number of tasks in which colluders have collaborated in the past), (iii) group deviation (i.e. an indicator to show the deviation of content produced by the colluders from those of other honest participants), (iv) group connectivity degree (an indicator to show to what extend the colluders are socially connected to each other), an (v) group content similarity (i.e., the degree of similarity of content produced by the group members). By considering all these indicators, the collusion prevention method determines a collusion probability for each participant and prevents the selection of the colluding participants.

In summary, the main contributions of this paper are as follows:

- We propose a method which is responsible for calculating a collusion possibility for each eligible participant to prevent any possible collusion on the task .
- We introduce five collusion indicators that resemble the behavior of colluding group members. We then provide equation to quantify each indicator and combine them to reach to a single value as the possibility of collusion .
- The accuracy and usability of the proposed techniques have been tested using real world datasets from the Advogato social network and Wikipedia Adminship Election and simulated experiments. The evaluation results show superiority of our method over the other common recruitment methods.

The rest of the paper is organised as follows. Related work is discussed in Section 2. We present the details of our collusion detection scheme in Section 3. Simulation results are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

Social participatory systems can be regarded as a subset of collective intelligence systems, which are defined broadly as groups of individuals doing things collectively that seem intelligent [37]. Due to the openness of such systems, the selection and recruitment of well-suited participants has always been a great concern [38]. In the following, we will have a short review on the related works on the issue of collusion prevention among selected participants and will discuss the state-of-the-art.

Collusion detection has been widely studied in P2P systems [39, 40]. A comprehensive survey on collusion detection in P2P systems can be found in [39]. In their work, the authors extensively study and classify the reputation systems and micro-payments systems (MPS) as two main approaches in P2P systems against collusion. Reputation management systems are also targeted by collusion. Colluders in reputation management systems try to manipulate reputation scores by collusion. Much effort is put into detecting collusion using majority rules, weight of the worker and temporal analysis of the behavior of the users [41], but none of these methods are strong enough to detect all sorts of collusion [41].

In [20], Mukherjee et al. have proposed a model for spotting fake review groups in online rating systems. The model analyzes textual feedback cast on products in Amazon's online market to find collusion groups. They use eight indicators to identify colluders and propose an algorithm for ranking collusion groups based on their degree of spamicity. However, their proposed method is still vulnerable to some attacks. For example, if the number of attackers is much higher than honest raters on a product the model cannot identify this as a potential case of collusion.

In the domain of participatory sensing, the authors in [18] aim at detecting the collusion by leveraging a reputation management system and outlier detection algorithms. In [15], a trusted platform module (TPM) is provided with each sensor device to attest the integrity of sensor readings. This local integrity checking makes the system resistant to collusion. To the best of our knowledge, the collusion prevention has not been discussed in social participatory sensing, and the methods proposed for participatory sensing are not applicable to this domain.

3. Collusion Detection

An online social network is best represented as an undirected graph with the set of nodes representing participants and the set of friendship relations between nodes. Each participant has a profile containing his attributes and related information. Some attributes represent the participant's personal information such as name and address. Others include the outcome of participant's social behaviour. Examples are the participant's reputation score, the history of his previous transactions, the pairwise trust scores, etc. A participatory task or simply a task is represented by θ_i, and Θ is the set of all the tasks to be solved (Θ = $\{\theta_i\}$. The owner of the task is also called the *requester*. Ψ is the set of *participants* who contribute to the task (Ψ = $\{\psi_i\}$). They provide the requester with a set of contributions represented by κ.

As mentioned above, selecting well-suited participants is important for acquiring high quality contributions since these participants have better knowledge and expertise relevant to the task requirements. In our previous works [21, 22, 23, 24], we addressed the challenge of well-suited participant selection. Specifically, we proposed schemes and procedures for crawling through the social network starting at the requester and identifying well-suited participants. We define the suitable participants as those who can satisfy the task requirements. In order to evaluate the participant's suitability, we defined and quantified a set of suitability parameters. These parameters include the participant's expertise to satisfy the task's skill requirements, his locality for location-based tasks, his reputation score, etc. The suitability parameters are evaluated for each eligible participant and combined to form a suitability score for him.

Once the participant ψ_i is considered to be suitable for being selected, a final check should be done to ensure that the selection of ψ_i will not result in potential collusion. In particular, we aim to identify whether the addition of ψ_i to the set of previously selected participants will result in the formation of a group of colluders.

Collaborative attacks which are also called collusion attacks are those in which, a number of individuals form a clique and collaborate on changing the results of a task [20]. For example, colluders may collaborate as they wish to produce poor quality contributions that severely impact the goal of the task.

We define a *group g* consisting of a set of participants Ψ^g and a set of tasks Θ^g. In other words, g = $\{\Psi^g, \Theta^g\}$. All the participants in Ψ^g have contributed to all the tasks in Θ^g.

Identifying the collusive groups requires two steps. In the first step, all existing collaborative groups (that fit within the above definition) are identified. In the second step, the potential collusive groups are detected among the identified groups. The detection of collusive groups is carried out based on a set of indicators. In the following section, we discuss these steps in detail.

3.1 Identifying Potentially Colluding Groups

In order to identify all collaborative groups among the selected participants, the collusion prevention method employs the Frequent Itemset Mining (FIM) technique [25]. FIM is a method for market basket analysis. It aims at finding regularities in the shopping behaviour of customers of supermarkets, mail-order companies, on-line shops etc. More specifically, FIM intends to find sets of products that are frequently bought together. There are multiple applications for the identified frequent item sets such as improving arrangement of products in shelves, on catalogue pages etc., supporting cross-selling (suggestion of other products), product bundling and fraud detection [26, 27, 28]. Identified patterns are typically expressed as association rules, e.g., if a customer buys bread and butter, then this customer will probably buy cheese, too. The performance and accuracy of the FIM technique is discussed in [29]. FIM is one of the major group detection algorithms which have been extensively used for collusion detection in online rating systems [20]. Hence, in our collusion prevention method, we make use of the FIM algorithm to find potential collusive groups.

The description of the FIM is as follows [29]: Let I = $\{i_1, i_2, \ldots, i_n\}$ be a set of items and D be a multiset of transactions, where each transaction T is a set of items such that $T \subseteq I$. For any $X \subseteq I$, we say that a transaction T contains X if $X \subseteq T$. The set X is called an itemset. The count of an itemset X is the number of transactions in D that contain X. The support of an itemset X is the proportion of transactions in D that contain X. An item set X is called *frequent* if its support is greater than or equal to some given percentage s, where s is called the minimum support. In our context, the set of items *(I)* is the set of all selected participants for the current task. The set of transactions *(D)* is the set of all tasks that a participant has been involved in the past. By mining frequent itemsets, we find groups of participants who have contributed to multiple tasks together.

3.2 Collusion Indicators

Most existing collusion detection techniques rely on 'behavioural' indicators to identify colluding groups [20, 30, 31]. These indicators reflect suspicious behaviour from a group of participants which indicates the possibility of collusion. Colluders usually form a group which is typically large enough to gain the majority and make a considerable impact [20]. Moreover, group members usually target a considerable number of tasks and collaborate together in contributing to these tasks. We also claim that the colluders prefer to connect with each other in the form of groups (such as social groups in OSNs) to facilitate their communications.

Group connivance is also represented by some 'content-related' indicators. Colluders normally report contributions with typically similar (duplicate or near duplicate) contents in order to ensure that the task outcome is different from the true consensus. Moreover, their contributions deviate from the other (genuine) participants in order to change the task outcome. In order to have a better view of content-based collusion indicators, we provide an illustrative example. Recall the PetrolWatch application [5] in which, participants are recruited to take photos of fuel price billboards. The photos are then aggregated in the server and the fuel prices are extracted. The cheapest fuel price for each area is then identified (for example by leveraging majority consensus). People are then able to query the server to access the cheapest fuel price in their area of interest. Consider a situation in which, a service station operator is aware that there is a contest between the nearby stations to have more costumers. The operator is aware that PetrolWatch uses majority consensus and comes up with a plan to game the system with the aim of attracting more customers to his business. The service station operator asks several of his social friends to collusively report false data for the competing service stations by uploading old pictures of higher fuel prices. If the false prices reported by his friends are more than the correct prices reported by other people, the collusion attack will be successful.

Table 1. Fuel prices of three different service stations uploaded by eight participants The abbreviation "inc." is used to denote incorrect prices (e.g., due to not being able to successfully recognise the price in the image).

Participants	Station1	Station2	Station3
1	123.0	119.0	Inc.
2	123.0	119.0	Inc.
3	123.0	119.0	121.3
4	123.0	119.0	Inc.
5	123.0	119.0	121.3
6(m)	123.0	125.0	124.5
7(m)	123.0	125.0	124.5
8(m)	123.0	125.0	124.5
correct ¢	123.0	119.0	121.3
majority consensus ¢	123.0	119.0	124.5

Table 1 (taken from [18]) is an example of this scenario that represents the fuel prices reported by 8 participants. We assume that the malicious operator owns service station 1 and that participants ψ_6 , ψ_7, and ψ_8 (denoted by 'm' in the table) are his workers who have formed a collusive group and report higher prices for service stations 2 and 3. As shown in Table 1, all the colluding members report the same data (¢125.0 for station 2 and ¢124.5 for station 3) to set a higher price for these stations. Also, the price reported by the group members deviates from the prices forwarded by other genuine participants 1-5 in order to change the outcome of majority consensus. The result obtained from the majority consensus (the last row of Table 1) shows that the colluding group is successful in falsifying the genuine price of service station 3 since they constitute the majority and hence, their reported price is selected as the true price. This example illustrates the need to examine certain features that suggest the likelihood of the existence of a colluding group.

Similar to the concepts discussed above, in our collusion prevention method, we consider a set of indicators. These indicators suggest that a colluding group is likely to exist among the selected participants. Note that these indicators reflect the likelihood of collusion only when they all occur together. In the following, we explain each indicator in detail and discuss how they identify possible collusive activities.

- **Group Size (GS).** The first indicator is the group size which is proportional to the *number of colluders* who have collaborated as a group in similar tasks. Group size (normalised) for a group g *(GS^g)* is calculated as follows,

  $$GS^g = \frac{|\Psi^g|}{\max(|\Psi^g|)} \tag{1}$$

  Where $\max(|\Psi^g|)$ is the largest group size of all found groups. *GS* is a parameter in the range of (0, 1], i.e. $0 < GS \leq 1$, showing how large the group is in comparison with other groups.

- **Group Target Size (GTS).** While the group size measures the **number** of group members, group target size measures the *number of tasks* in which the group members have targeted to collaborate in the past. Groups with a high value of target size are more likely to be colluding as the probability of a group of random people to have attended the same tasks together is rather small. For a group g, *GTS^g* is calculated as follows.

  $$GTS^g = \frac{|\Theta^g|}{\max(|\Theta^g|)} \tag{2}$$

  Where max $(|\Theta^g|)$ is the largest target size of all found groups. GTS is a number in range (0, 1], i.e. $0 < GTS \leq 1$.

- **Group Deviation (GD).** The third indicator is group deviation which is an indicator to show the difference between the contents contributed by the colluders and those reported by other (honest) participants. In order to calculate the group deviation, we first calculate the deviation of the contents produced by group members from those of other participants for a single task $t\epsilon\Theta^g$. For each task $t\epsilon\Theta^g$, the deviation of the group (GD_t^g) is calculated as follows:

  $$GD_t^g = \left| \underset{\iota\epsilon\Psi^g}{\overline{\kappa_{\iota,t}}} - \underset{\jmath\notin\Psi^g}{\overline{\kappa_{\jmath,t}}} \right|, for\ all\ i, j\epsilon\Psi^g \tag{3}$$

  Where $\overline{\kappa_{\iota,t}}$ and $\overline{\kappa_{\jmath,t}}$ are the average of contents for task *t* given by members of group *g* and by other participants not in *g*, respectively.

  Now, for a group *g*, the group deviation, denoted by *GD^g*, is the maximum of all group deviations for all tasks in Θ^g. In other words, *GD^g* is computed as:

$$GD^g = \max_{t\epsilon\Theta^g}(GD_t^g) \quad (4)$$

GD is a number in range (0, 1], i.e. $0 < GD \leq 1$.

- **Group Connectivity (GC).** The fourth indicator which is specifically suited for social communities is the group connectivity degree which is an indicator to show to what extent the colluders are connected to each other. For a group g, we first calculate the number of links between group members and denote it by link count (LC^g). LC^g is calculated as:

$$LC^g = \sum_{i\epsilon\Psi^g}\sum_{j\epsilon\Psi^g} T_{i,j}\ for\ all\ i,j\epsilon\Psi^g \quad (5)$$

Where,

$$T_{i,j} = \begin{cases} 1 & if\ i \rightarrow j\ (there\ is\ a\ link\ from\ i\ to\ j) \\ 0 & otherwise \end{cases}$$

GC^g is then computed as follows.

$$GC^g = \frac{LC^g}{\max(LC^g)} \quad (6)$$

GC is a number in range (0, 1], i.e. $0 < GC \leq 1$.

- **Group Content Similarity (GCS).** The fifth indicator is group content similarity which indicates the degree of similarity of contents produced by group members. In order to evaluate this similarity, we first calculate the pairwise content similarity between every pair of members in the group. Pairwise content similarity between ψ_i and ψ_j, denoted by $GCS_{i,j}^g$, shows to what extent ψ_i and ψ_j have reported similar contents. In order to calculate the pairwise similarities between group members, we use the cosine similarity model, a well-known model for similarity detection [32]. Specifically, $GCS_{i,j}^g$ will be the cosine of the angle between two vectors containing the contribution contents of ψ_i and ψ_j and is a value in the range (0, 1). The value 1 for $GCS_{i,j}^g$ means completely the same while 0 means completely different. $GCS_{i,j}^g$ is calculated as follows.

$$GCS_{i,j}^g = \frac{\sum_{t\epsilon\Theta^g}\kappa_{i,t}\times\kappa_{j,t}}{\sqrt{\sum_{t\epsilon\Theta^g}(\kappa_{i,t})^2}\times\sqrt{\sum_{t\epsilon\Theta^g}(\kappa_{j,t})^2}} \quad (7)$$

We then calculate an overall degree of similarity for the group to show how all members are similar in terms of contents they have contributed. Group content similarity for every group g, denoted by GCS^g is the minimum amount of pairwise similarities between group members. In other words,

$$GCS^g = \min_{i,j\epsilon\Psi^g}(GCS_{i,j}^g) \quad (8)$$

GCS is a number in range (0, 1], i.e. $0<GCS \leq 1$.

3.3 Possibility of Collusion

It is often difficult to determine with certainty whether a group is collusive [20]. Therefore, we define a metric called *Possibility of Collusion (PoC)* to show to what extent a group is potentially collusive. *PoC* is an aggregation of five collusion indicators. Since the importance of these indicators may be different in various applications, the collusion prevention method enables the applications to assign weight to each indicator based on its importance.

Suppose that W_{GS}, W_{GTS}, W_{GD}, W_{GC} and W_{GCS} are corresponding weights for indicators GS^g, GTS^g, GD^g, GC^g and GCS^g. The weights are initialised in a way that: $W_{GS} + W_{GTS} + W_{GD} + W_{GC} + W_{GCS} = 1$ and are set based on the application settings and nature, as described above. *PoC* is then calculated as:

$$PoC(g) = GS^g \times W_{GS} + GTS^g \times W_{GTS} + GD^g \times W_{GD} + GC^g \times W_{GC} + GCS^g \times W_{GCS} \quad (9)$$

PoC is a number in range (0, 1]. For each eligible participant ψ_i to be selected, we calculate the possibility of collusion (*PoC(g))*. If greater than a certain threshold, it implies that the selection of ψ_i may lead to potential collusion, and hence, the participant will not be selected.

4. Experimentation and Evaluation

In this section, we conduct a simulation-based evaluation to analyse the behaviour of our proposed collusion prevention method. First, we explain the experimentation set up and the datasets we used in experiments in Section 3.1. Then in Section 3.2, we investigate the efficiency of our proposed collusion prevention method.

4.1. Simulation Set-up

Our simulations have been conducted on a PC running Windows 7:0 Professional and having 4GB of RAM. We used Matlab R2012 for developing the simulator.

4.1.1 Datasets

In order to evaluate the performance of our proposed collusion prevention method, we set two experiments. In the first experiment, we aimed at utilising a real dataset for which, the possibility of collusion exists due to gaining benefits. Hence, we utilised the Wikipedia voting dataset. In Wikipedia, the voting process is used to elect administrators[4]. Every registered user can nominate himself or another user as an administrator in Wikipedia and initiate an election. The other users participate in the election and cast their votes on the eligibility of nominee. If the majority of users recognise a user as eligible, this user then will become a Wikipedia administrator. In order to incorporate this dataset in the context of our method, we employ the following mapping. The requester is the nominee, the worker is the voter, the task is evaluating the

[4] http://en.wikipedia.org/wiki/Wikipedia:Requests_for_adminship

eligibility of the nominee as an administrator in Wikipedia and the contribution is the worker's vote. We use the log of Wikipedia Adminship Election[5] which was collected by Leskovec et al. for behaviour prediction in online social networks [33], referred to as WIKILog. WIKILog contains about 2800 elections (tasks) with around 100000 total votes and about 7000 users participating in the elections either as a voter or a nominee. We use the WIKILog to demonstrate the efficacy of our proposed method to detect collusion.

The dataset that we use for the second experiment is the real web of trust of Advogato.org [34]. Advogato.org is a web-based community of open source software developers in which, site members rate each other in terms of their trustworthiness. Trust values are one of the three choices master, journeyer and apprentice, with master being the highest level in that order. The result of these ratings is a rich web of trust, which comprises of 14019 users and 47347 trust ratings. In order to conform it to our framework, we map the textual ratings to the range of [0, 1] as master = 0.8, journeyer = 0.6, and apprentice = 0.4. Advogato web of trust can be regarded as a social participatory sensing system with users as the potential participants and trust ratings as the friendship relations. In order to better investigate the performance of our method, we artificially created collusive groups among Advogato members. We then investigated whether the proposed collusion prevention method is able to identify these groups.

4.2 Collusion Prevention Analysis

As mentioned in Section 3.1, to evaluate the performance of the collusion prevention method, we set two experiments. The experiments differ in their employed datasets. In the following, we explain the results of each experiment in detail.

4.2.1 Wikipedia Adminship Election Dataset

In the first experiment, we use the Wikipedia adminship election dataset to investigate the performance of our proposed collusion prevention method. The dataset contains the information related to 2794 tasks. The average number of participants in these tasks is 40. In order to obtain reliable results, we consider the tasks with number of participants greater than the average as the sample data, and randomly select 100 tasks from these. We then test our proposed method to identify any potential colluding group among the participants. As mentioned in Section 2.2, we consider five indicators for detecting potential collusion. Among these indicators, the two indicators Group Size (GS) and Group Target Size (GTS) are the most important indicators as they are the basic conditions for the f*ormatio*n of a group. Basically, a group *g* is created when at least th_1 members of *g* have collaborated in at least th_2 tasks. So, we first run a short experiment to define the optimal values for th_1 and th_2.

In order to find the optimum value for th_2, we set an experiment in which, the target size (i.e. number of the tasks for which the group members have collaborated in the past) is changed. For each target size, we measure the number of groups identified, together with their size. As can be seen in Fig. 1, the maximum size of identified groups is decreased by increasing the target size. This is rational since the probability of finding groups whose members have collaborated in a greater number of tasks is smaller. We believe that the best setting is the one which results in the identification of the largest groups to make a considerable impact. As derived from the figure, this situation is related to the case where the target size is 6. So, we set th_2 to be equal to 6. For the sake of simplicity, we assume that th_1 equals to th_2.

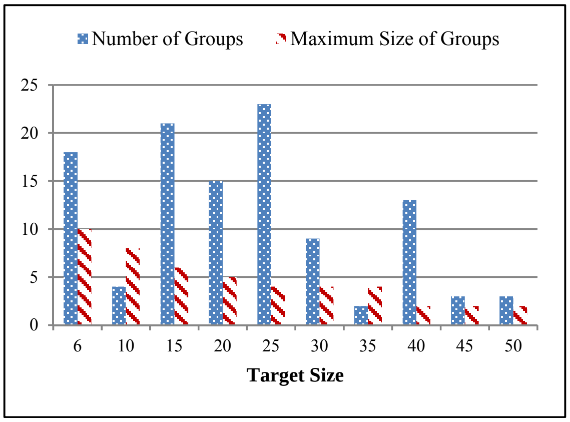


Fig. 1: Evolution of number of groups and their maximum size according to the target size.

In order to investigate the performance of our proposed collusion prevention method, we first utilise the FIM technique to find the candidate groups among the participants. The outcome is the discovery of 18 candidate groups with at least 10 members. We then employ our collusion prevention method and identify 9 of these 18 groups as collusive. To evaluate the efficiency and accuracy of our method, we examine a number of statistical metrics. At first, we measure the ratio of the tasks targeted with the colluding groups. The result shows that 14% of the tasks were affected by these 9 colluding groups. This means that

[5] http://snap.stanford.edu/data/wiki-Elec.html

our collusion prevention method is able to prevent these tasks from being targeted by the colluders. We then calculate the success ratio of the tasks targeted by the colluding groups as well as all 100 tasks. In the Wikipedia adminship election dataset, a task (an election) is successful if it results in the selection of the user as an administrator (note that the results are available in the dataset). By success ratio, we mean the ratio of the tasks that have resulted in a desired decision (i.e. resulted in the selection of a user as an admin), to the total number of tasks. We observe that overall success ratio of the tasks in our dataset is 71%. This ratio is 83% for the groups identified by our collusion detection method. This means that there is a high probability that the groups identified by our method are colluding groups, since their collaboration has resulted in a considerably high success ratio. This is a significant indication that the identified groups are much more likely to be collusive.

4.2.2 Advogato Dataset

In the second experiment, we use Advogato dataset. We first create a set of candidate groups among the Advogato members, and then, we define some of these candidate groups as collusive. In order to create candidate groups, we first select 90 Advogato members with at least 30 trust relations (i.e. 30 friends). Each of these members along with 20 out of his 30 friends forms a candidate group. When a task is released, a set of Advogato members are considered as eligible to contribute (by using the aforementioned suitability assessment and eligibility assessment techniques). Each candidate group with at least 10 eligible members is considered as collusive. The collusive group members contribute polluted data while other eligible members contribute genuine data. Specifically, we assume that the genuine data (d) is a random number in [0,1], while the polluted data is a random number in (d - μ , d + μ). Greater values for μ result in polluted values with great deviation from the genuine values, which makes the collusion detection easier. In our experiments, we set μ to be 0.2. Note that for each task, all the collusive members report the contaminated data, while others report the genuine data. We run the experiment for 10 rounds. In each round, 20 tasks are released. At the end of each round, we utilise the FIM technique to find the groups. The outcome is the set of all groups among the eligible participants (who have collaborated in at least 5 tasks). Then, for each group identified by FIM, the possibility of collusion (*PoC*) is computed by utilising Equation 9. While we believe that the threshold for *PoC* should be application-specific, in our experiments, we assume that groups with *PoC* > 0.5 are identified as collusive (In Equation 9, for simplicity, we assume that all the indicator weights are equal to 0.2).

In order to evaluate the efficiency and the accuracy of our proposed method in identifying the colluding groups, we utilise two criteria. For the evaluation of accuracy, we use the well-known measures of precision and recall [35]. Precision measures the quality of the identification results, and is defined by the ratio of the correct identification of colluding groups, to the total number of groups identified by our method. Recall measures coverage of the identification results, and is defined by the ratio of the collusive groups identified correctly to the total number of all correct colluding groups that should be found. These two definitions are summarised in the following equations:

$$Precision = \frac{number\ of\ collusive\ groups\ identifised\ correctly}{total\ number\ of\ identified\ groups}$$

$$\boldsymbol{Recall = \frac{number\ of\ collusiveroups\ identifised\ correctly}{total\ number\ of\ existing\ collusive\ groups}}$$

These two measures are usually expressed as percentages. For an approach to be effective, it should achieve a high precision and high recall. However, in reality these two metrics tend to be inversely related [36]. This means that the improvements in precision come at a cost of reduction in recall, and vice versa.

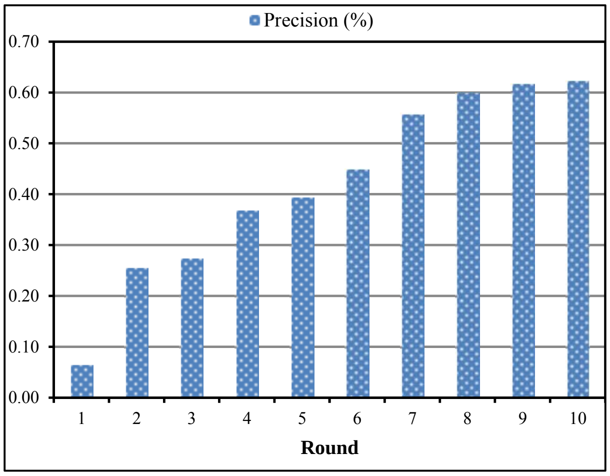


Fig. 2. Evolution of precision (%) in different rounds.

Fig. 2 shows the evolution of precision in different rounds. As displayed in this figure, the collusion detection method achieves a precision of 63%. This means that our collusion prevention method is able to prevent 63% of the tasks from being targeted by the colluders. This is due to the suitability of the indicators, which correctly model the collusive behaviour of group members. Note that it may be possible to achieve greater precision but would result in a drop in recall. As can be observed in this figure, the precision values evolve in a constantly in-creasing manner. A lower value of precision in the first rounds is due to the lack of adequate history related to the colluders' behaviours. In other words, due to the small number of released tasks in the first rounds of the experiment, the collusion prevention method does not have the required information (e.g., content similarity, target size, etc.) at hand. As time goes by, collusive members collaborate in more tasks which results in the availability of more behavioural information such as number of the tasks they have collaborated on, the contributions they have re-ported to these tasks, etc. This helps the collusion prevention method to better detect the collusive behavioural pattern.

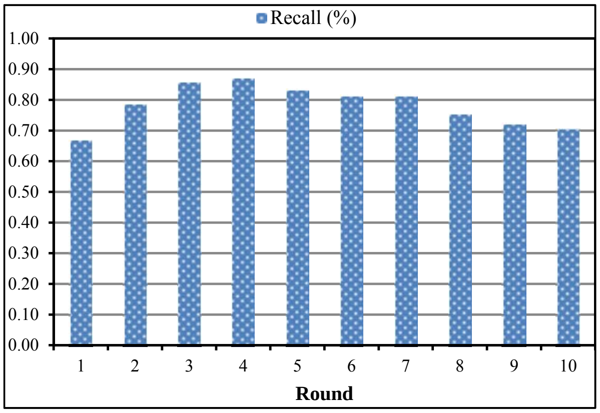


Fig. 3: Evolution of recall (%) in different rounds.

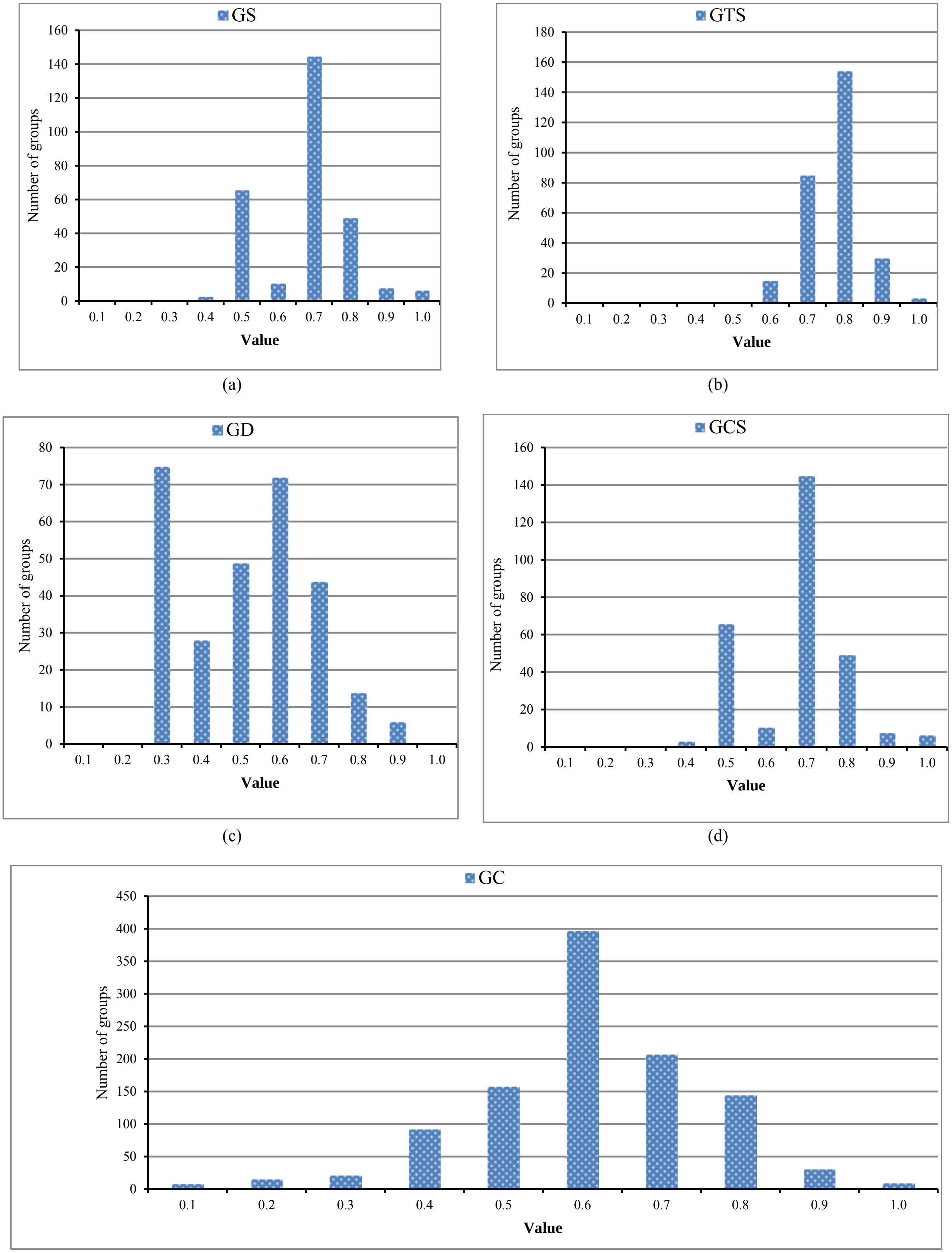
Fig. 3 depicts the evolution of recall in various rounds. As can be seen in this figure, our method also achieves a high percentage of recall (86%), which denotes that our collusion prevention method is successful in detecting 86% of the existing collusive groups. It can be observed that there is a slightly descending growth in recall after the fourth round which, as mentioned above, is natural in real systems, since precision and recall typically evolve inversely [36].

As mentioned above, a group is identified as collusive if the possibility of collusion (*PoC*) for this group is above 0.5. The possibility of collusion is obtained by averaging the indicator values. However, in order to ensure that the indicators are selected correctly, we calculate the distribution of values of each indicator in all collusive groups identified by our method. Figures 4.a. to 4.e depict the distribution of values calculated for collusion indicators. The values calculated for indicators are almost always higher than 0.5. This illustrates that the identified indicators are suitable and effective for detecting collusion in social participatory sensing.

In a nutshell, the results show that our proposed collusion prevention method is successful in preventing the formation of colluding groups among the selected participants with high accuracy.

5. Conclusion

In this paper, we proposed a collusion-resistant participant selection method with the goal to prevent the formation of colluding groups within the selected suitable participants. The method investigates the possibility of collusion upon each eligible participant. This decision is made based on a set of indicators that are related to the common approaches utilised by colluders to arrange a collusive attack. Colluders normally form a large group and collectively collaborate on a large number of tasks. They normally contribute similar content which deviate from the genuine contributions provided by honest participants. They may also benefit from the social groups to better manage their communications. We then calculated the possibility of collusion based on these indicators. In order to measure the performance of the collusion prevention method, we set up two experiments in which, the datasets Wikipedia adminship election and Advogato were employed. The result of these experiments showed that our proposed method is able to detect the collusive groups with high precision. The results also demonstrated the correctness and effectiveness of proposed indicators.



(a) (b)

(c) (d)

(e)

Fig. 4. Distribution of the values of indicators in collusion attacks

References

[1] A. Doan, R. Ramakrishnan, and A. Y. Halevy, “Crowdsourcing systems on the world-wide web,” Communications of the ACM, vol. 54, no. 4, pp. 86–96, 2011.

[2] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and B. Srivastava, “Participatory sensing,” in Proceedings of the 1st Workshop on World-Sensor-Web (WSW), 2006, pp. 1–5.

[3] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, “Image browsing, processing, and clustering for participatory sensing: Lessons from a dietsense prototype,” in Proceedings of the 4th ACM Workshop on Embedded Networked Sensors, 2007, pp. 13–17.

[4] E. P. Stuntebeck, J. S. Davis II, G. D. Abowd, and M. Blount, “Healthsense: classification of health-related sensor data through user-assisted machine learn-ing,” in Proceedings of the 9th workshop on Mobile computing systems and applications. ACM, 2008, pp. 1–5.

[5] Y. F. Dong, S. S. Kanhere, C. T. Chou, and R. P. Liu, “Automatic image capturing and processing for petrolwatch,” in Proceedings of the 17th IEEE International Conference on Networks (ICON), 2011, pp. 236–240.

[6] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, Balakrishnan, and S. Madden, “Cartel: a distributed mobile sensor computing system,” in Proceedings of the 4th international conference on Embedded networked sensor systems. ACM, 2006, pp. 125–138.

[7] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, “Ear-phone: an end-to-end participatory urban noise mapping system,” in Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2010, pp. 105–116.

[8] P. Johnson, A. Kapadia, D. Kotz, N. Triandopoulos, and N. Hanover, “People-centric urban sensing: Security challenges for the new paradigm,” TR2007-586, Dartmouth College, Computer Science, Hanover, NH, Tech. Rep., 2007.

[9] K. Shilton, “Four billion little brothers?: Privacy, mobile phones, and ubiqui-tous data collection,” Communications of the ACM, vol. 52, no. 11, pp. 48–53, 2009.

[10] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, “A survey on privacy in mobile participatory sensing applications,” Journal of Systems and Software, vol. 84, no. 11, pp. 1928–1946, 2011.

[11] I. Krontiris and F. C. Freiling, “Integrating people-centric sensing with social networks: A privacy research agenda,” in Proceedings of the 8th IEEE Inter-national Conference on Pervasive Computing and Communications Workshops (PERCOM), 2010, pp. 620–623.

[12] I. Krontiris and F. Freiling, “Urban sensing through social networks: The tension between participation and privacy,” in Proceedings of the International Tyrrhenian Workshop on Digital Communications (ITWDC), 2010.

[13] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosman-oglu, “Crowd-sourced sensing and collaboration using twitter,” in Proceedings of the IEEE International Symposium on World of Wireless Mobile and Multi-media Networks (WoWMoM), 2010, pp. 1–9.

[14] S. Reddy, D. Estrin, and M. Srivastava, “Recruitment framework for partici-patory sensing data collections,” in Pervasive Computing, ser. Lecture Notes in Computer Science, 2010, vol. 6030, pp. 138–155.

[15] A. Dua, N. Bulusu, W.-C. Feng, and W. Hu, “Towards trustworthy participa-tory sensing,” in Proceedings of the Usenix Workshop on Hot Topics in Security (HotSec), 2009, pp. 8–8.

[16] S. Saroiu and A. Wolman, “I am a sensor, and i approve this message,” in Proceedings of the 11th ACM Workshop on Mobile Computing Systems and Applications (HotMobile), 2010, pp. 37–42.

[17] “Trusted computing group,” https://www.trustedcomputinggroup.org/home.

[18] K. L. Huang, S. S. Kanhere, and W. Hu, “On the need for a reputation system in mobile phone based sensing,” Ad Hoc Networks, vol. 12, no. 0, pp. 130–149, 2014.

[19] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” in ACM Transactions on Sensor Networks (TOSN), 2008, vol. 4, no. 3, p. 15.

[20] A. Mukherjee, B. Liu, and N. Glance, “Spotting fake reviewer groups in consumer reviews,” in Proceedings of the 21st ACM International Conference on World Wide Web, 2012, pp. 191–200.

[21] H. Amintoosi and S. Kanhere, “A reputation framework for social participa-tory sensing systems,” in Mobile Networks and Applications (MONET), 2014, vol. 19, no. 1, pp. 88–100.

[22] H. Amintoosi and S. S. Kanhere, “A trust-based recruitment framework for multi-hop social participatory sensing,” in Proceedings of the 9th IEEE Inter-national Conference on Distributed Computing in Sensor Systems (DCOSS), 2013, pp. 266–273.

[23] H. Amintoosi and S. Kanhere, “Privacy-aware trust-based recruitment in so-cial participatory sensing,” in 10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MobiQuitous), 2013, pp. 262–275.

[24] H. Amintoosi, S. S. Kanhere, and M. Allahbakhsh, “Trust-based privacy-aware participant selection in social participatory sensing,” Journal of Infor-mation Security and Applications, vol. 20, pp. 11–25, 2015.

[25] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), vol. 1215, 1994, pp. 487–499.

[26] T. Brijs, B. Goethals, G. Swinnen, K. Vanhoof, and G. Wets, “A data mining framework for optimal product selection in retail supermarket data: the generalized profset model,” in Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 300–304.

[27] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets, “Using association rules for product assortment decisions: A case study,” in Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 254–260.

[28] W. Lee, S. J. Stolfo, and K. W. Mok, “Adaptive intrusion detection: A data mining approach,” Artificial Intelligence Review, vol. 14, no. 6, pp. 533–567, 2000.

[29] G. Grahne and J. Zhu, “Fast algorithms for frequent itemset mining using fp-trees,” IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 10, pp. 1347–1362, 2005.

[30] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S.-M.-R. Beheshti, E. Bertino, and N. Foo, “Collusion detection in online rating systems,” in Proceedings of the 15th Asia Pacific Web Conference (APWeb), 2013, pp. 196–207.

[31] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, “Detecting product review spammers using rating behaviors,” in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), 2010, pp. 939–948.

[32] G. Salton, C. Buckley, and E. A. Fox, “Automatic query formulations in in-formation retrieval,” Journal of the American Society for Information Science, vol. 34, no. 4, pp. 262–280, 1983.

[33] J. Leskovec, L. Adamic, and B. Huberman, “The dynamics of viral market-ing,” ACM Transactions on the Web (TWEB), vol. 1, no. 1, pp. 1–39, 2007.

[34] R. Levien and A. Aiken, “Attack-resistant trust metrics for public key cer-tification,” in Proceedings of the 7th USENIX Security Symposium, 1998, pp. 229–242.

[35] H. R. Motahari Nezhad, G. Y. Xu, and B. Benatallah, “Protocol-aware matching of web service interfaces for adapter development,” in Proceedings of the 19th ACM International Conference on World Wide Web, 2010, pp. 731– 740.

[36] G. Chowdhury, Introduction to Modern Information Retrieval. Facet pub-lishing, 2010.

[37] T. Malone, R. Laubacher, and C. Dellarocas, “Harnessing crowds: Mapping the genome of collective intelligence,” in MIT Sloan Research Paper, 2009.

[38] S. S. Kanhere, “Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces,” in Distributed Computing and Internet Tech-nology. Springer, 2013, pp. 19–26.

[39] “Collusion in peer-to-peer systems,” in Computer Networks, 2011, vol. 55, no. 15, pp. 3517 – 3532.

[40] Q. Lian et. al., “An empirical study of collusion behavior in the maze p2p file-sharing system,” in 27th International Conference on Distributed Comput-ing Systems. IEEE Computer Society, 2007, pp. 56–.

[41] Y. Sun and Y. Liu, “Security of online reputation systems: The evolution of attacks and defenses,” in Signal Processing Magazine, IEEE, 2012, vol. 29, no. 2, pp. 87 –97.