Xu *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:113
https://doi.org/10.1186/s13638-019-1439-8

EURASIP Journal on Wireless Communications and Networking

RESEARCH **Open Access**

A privacy-preserving exception handling approach for dynamic mobile crowdsourcing applications

Yanwei Xu[1,2], Hanwen Liu[2] and Chao Yan[1*]

Abstract

The ever-increasing popularity of mobile devices (e.g., mobile phones and smart watches) has created a variety of crowdsourcing applications by employing the massive and distributed mobile computing resources. Typically, a task requester sends his/her task request and constraint conditions to a crowdsourcing platform, and then the crowdsourcing platform is responsible for finding a set of appropriate workers (e.g., mobile users) from massive candidates to satisfy the task request. However, for a mobile crowdsourcing task being executed by a set of workers, a pre-selected worker may become unavailable due to various exceptions. In this situation, it is significant for the crowdsourcing platform to quickly find another similar worker to replace the unavailable worker so as to smooth the crowdsourcing process. However, the above exception handling process is often challenging as candidate workers are often not willing to release their sensitive information to the platform due to privacy concerns. In view of this challenge, in this paper, a novel privacy-preserving exception handling approach, named $ExH_{Simhash}$, is put forward based on Simhash technique. Finally, through a set of simulated experiments, we validate the feasibility of $ExH_{Simhash}$ in terms of substitution equivalence and computational time.

Keywords: Mobile crowdsourcing, Exception handling, Privacy, Efficiency, Simhash

1 Introduction

With the ever-increasing popularity of mobile computing techniques in daily life, people are apt to execute their business applications or complete their computing tasks through various mobile devices, such as mobile phones, smart watches, and laptops [1–8]. By utilizing the lightweight and smart mobile computing capabilities, a user can complete his/her complex computing tasks that cannot be executed traditionally by a single device or a single person, which gives birth to the mobile crowdsourcing technique. For example, if a user hopes to know about the real-time traffic condition of a routine that he/she is interested in, he/she can employ a set of mobile users to take some real-time pictures of the places near the routine. This way, on the one hand, the user can get his/her needed traffic data or information quickly; on the other hand, the mobile users that are employed for task execution can get a certain reward from the employer.

Generally, as Fig. 1 shows, the mobile crowdsourcing model consists of three roles: task requesters, candidate workers, and crowdsourcing platform. Typically, a task requester sends his/her outsourcing task to the crowdsourcing platform stating the task content (e.g., “take a set of pictures about the traffic condition of a route”) and constraint conditions (e.g., deadline, fees); candidate workers are the users who are willing to utilize their mobile devices to satisfy the outsourcing task from the task requester; the crowdsourcing platform receives the outsourcing tasks from task requesters and publish them to the candidate workers. If a worker meets the functional and non-functional requirements of the task simultaneously, then the crowdsourcing platform binds the requester and the worker; namely, an outsourcing contract is created between the two parties.

However, in the dynamic crowdsourcing environment, a previously selected worker may become unavailable for the crowdsourcing task due to various exceptions. For

* Correspondence: yanchao@qfnu.edu.cn
[1]Tianjin University, Tianjin 300354, China
Full list of author information is available at the end of the article

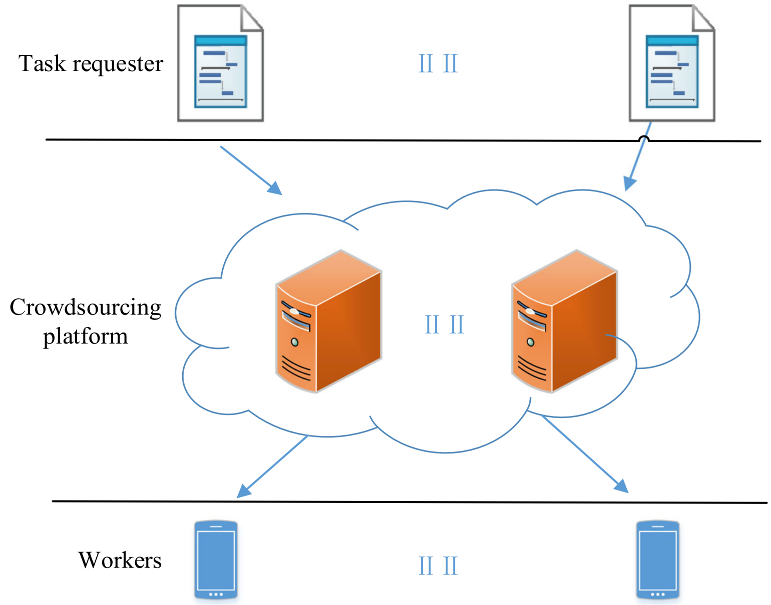


Fig. 1 Mobile crowdsourcing model

example, a worker who agreed to take some photos for the traffic conditions required by a task requester may become unavailable due to the low quantity of electric charge of his/her mobile phone. In this situation, for the task requester, it is necessary to quickly find another qualified worker who is similar with the unavailable worker to replace him/her, so as to smooth the normal execution of the mobile crowdsourcing task. However, the above similar worker search process is often a challenging problem, because the candidate workers are often not willing to release or publish their personal information to the crowdsourcing platform due to privacy concerns. For example, a worker is probably reluctant to tell the crowdsourcing platform that he/she has ever undertaken an outsourcing task "take some photos for a beautiful girl without her permission".

In view of the abovementioned challenge, a time-efficient and privacy-preserving data protection technique, i.e., Simhash [9], is introduced into the exception handling process of mobile crowdsourcing applications incurred by the unavailable workers. Furthermore, we put forward a novel exception handling approach based on Simhash, named $ExH_{Simhash}$ to enhance the robustness of traditional mobile crowdsourcing techniques.

Concretely, the major contributions of this paper are summarized as follows:

1. We recognize the probable mobile crowdsourcing exceptions caused by the unavailability of previously selected workers. The exceptions decrease the robustness of the crowdsourcing approaches.
2. We introduce the time-efficient and privacy-preserving data protection technique, i.e., Simhash into the mobile crowdsourcing process and further put forward a novel approach named $ExH_{Simhash}$, so as to handle the exceptions incurred by the unavailable workers.
3. We conduct a set of simulated experiments based on the well-known *WS-DREAM* dataset. Experiment results show that our suggested $ExH_{Simhash}$ approach performs better than other competitive approaches, especially in terms of substitution equivalence and computational time.

The reminder of this paper is structured as below. Related work is briefly introduced in Section 2. In Section 3, we formulate the privacy-preserving exception handling problems in mobile crowdsourcing applications. The suggested exception handling approach, i.e., $ExH_{Simhash}$, is specified in Section 4. In Section 5, we designed, deployed, and tested a set of experiments to prove the effectiveness and efficiency of $ExH_{Simhash}$. At last, in Section 6, we summarize the whole paper and discuss the potential improvement directions in the future research.

2 Related work

In recent years, mobile crowdsourcing technique has been introduced to satisfy the complex requirements that cannot be solved by the traditional techniques. During the mobile crowdsourcing process, selecting proper workers for the crowdsourcing task is crucial for providing better services. Many researchers have investigated this hot topic and provided their resolutions. In this section, we introduce the related work from the following two aspects.

2.1 Ability-based worker selection in crowdsourcing

The ability values of candidate workers play an important role in worker selection for mobile crowdsourcing applications. In the outsourcing platform Amazon Mechanical Turk, the candidate workers are evaluated, compared, and selected for an outsourcing task based on their ability values, e.g., worker reputation and worker skills. In [10], a self-organized outsourcing toolkit, i.e., Crowdlet is proposed to employ a set of workers to execute the crowdsourcing task from the requester. In Crowdlet, a candidate worker is selected or not depends on the worker's service quality levels which are influenced by the skills, arrival time, and rewards of the worker. While the reputation values and mobile device performance are often neglected. In [11], the authors suggest an online crowdsourcing framework which is response for allocating appropriate workers to crowdsourcing tasks based on the important information from both workers and requesters. The final goal is to maximize the utility function defined in the crowdsourcing framework. While the utility function only depends on the state of network. In [12], a behavior-aware worker selection strategy is suggested to improve crowdsourcing performances, which is often self-adaptive. Typically, the strategy calculates the reputation value of a candidate worker based on the worker's historical service quality data. However, the reputation model only considers the skills of workers without considering other elements such as device capability. In [13], a budget-aware crowdsourcing model evaluates and generates the final crowdsourcing results based on the workers' information, such as his/her willingness, his/her reliability in executing the task. Therefore, high-quality crowdsourcing results can be expected within a certain budget. However, the decision-making basis of crowdsourcing is still not comprehensive enough.

2.2 Dynamic mobile crowdsourcing selection

An online auction mechanism is put forward in [14] to satisfy the dynamic crowdsourcing scenarios where each candidate worker can arrive at or leave the crowdsourcing platform at will. The auction mechanism can be utilized to accommodate the dynamic arrivals of both candidate workers and crowdsourcing tasks. However, it is often hard to get the optimal crowdsourcing performance through auction. In [15], an incentive mechanism is brought forth to evaluate and rank the candidate workers for a certain crowdsourcing task based on the workers' context information, e.g., location and time. However, in this work, the proposed mechanism does not consider the ability values of workers. In [16], the authors put forward an online double auction strategy between candidate workers and task requesters. The proposal can guarantee to achieve better crowdsourcing results than the simple one-side interaction between workers and requesters. An online learning method is brought forth in [17] to produce the optimal pricing strategy and incentive mechanism, whose final goal is regret minimization. In [18], two online auction mechanisms are suggested to accommodate the dynamic arrival of candidate workers; the final optimization goal is to recruit enough workers to maximize the defined utility value before deadline constraint. In [19], the candidate workers are encouraged to publish their configuration files to the crowdsourcing platform; this way, the platform can reduce the budget while selecting enough workers to execute a crowdsourcing task. However, the abovementioned auction strategies in [17–19] are often not efficient enough to satisfy the quick response requirements from the task requesters in mobile crowdsourcing applications. Moreover, additional auction burdens are possible for both the candidate workers and the task requesters.

Through the above analyses, a conclusion can be drawn that existing mobile crowdsourcing approaches seldom consider the probable exceptions (as well as the resulted privacy leakage in similar worker search and replacement) caused by worker unavailability in the dynamic crowdsourcing environment. Considering this drawback, a privacy-preserving exception handling approach for dynamic mobile crowdsourcing is proposed in this paper, to smooth the normal execution of crowdsourcing process and improve the crowdsourcing robustness. The details of our exception handling approach will be introduced in Section 4.

3 Formulation

In this section, we formulate the mobile crowdsourcing problems and introduce the symbols to be used in the following discussions of this paper.

1. *Requester* is a user who outsources his/her tasks to a crowdsourcing platform.
2. $WORKER = \{worker_1, \ldots, worker_m\}$ denotes the candidate worker set.
3. *CP* is a crowdsourcing platform that receives tasks from *Requester* and publish or advertise them to the candidate workers in set *WORKER*.
4. $TASK = \{task_1, \ldots, task_n\}$ denotes the task set in historical crowdsourcing applications.
5. For $worker_i$ ($i = 1, \ldots, m$), his/her ever-executed crowdsourcing task set can be specified with an n-dimensional vector $V_i = (v_{i,1}, \ldots, v_{i,n})$, where $v_{i,j}$ denotes whether $worker_i$ has ever executed task t_j: if the answer is yes, then $v_{i,j} = 1$; otherwise, $v_{i,j} = 0$.

With the above definitions, we can further specify the dynamic mobile crowdsourcing problem formally: the crowdsourcing platform *CP* selects a set of qualified workers from candidate set *WORKER* to satisfy the requirements of a task (from *Requester*) in set *TASK*. Furthermore, the privacy-preserving exception handling problem can be specified as below: if a previously selected worker $worker_{excep}$ becomes unavailable due to some exceptions, the crowdsourcing platform *CP* should quickly find another candidate worker $worker_x$ who is similar with $worker_{excep}$, and meanwhile guarantee that the sensitive information contained in vectors V_i ($1 \le i \le m$) is still secure.

4 Exception handling for mobile crowdsourcing based on Simhash

In this section, we introduce the details of our suggested exception handling approach for mobile crowdsourcing, i.e., $ExH_{Simhash}$. The main idea of $ExH_{Simhash}$ is: when the previously selected $worker_{excep}$ turns unavailable, the crowdsourcing platform *CP* utilizes Simhash technique to search for the workers (denoted by set $WORKER_X$) similar with $worker_{excep}$ in an efficient and privacy-preserving manner, and utilizes the optimal $worker_{optimal}$ ($\in WORKER_X$) to replace the unavailable $worker_{excep}$.

Concretely, $ExH_{Simhash}$ consists of the following three steps.

Step 1: Building worker indices offline based on Simhash. According to the historical crowdsourcing tasks ever-executed by workers, build worker indices offline based on Simhash.
Step 2: Search for the similar workers of unavailable worker. For the unavailable worker $worker_{excep}$, search for his/her similar workers $worker_x$ based on the indices built in step 1.
Step 3: Optimal worker selection and replacement. From all the similar $worker_x$ obtained in step 2, we select an optimal one based on the service quality and utilize him/her to replace the unavailable $worker_{excep}$.

4.1 Step 1: building worker indices offline based on Simhash

As Section 3 introduces, if $worker_i$ ($\in WORKER$) has ever executed the crowdsourcing task $task_j$ ($\in TASK$), then the value (indicating whether a worker has ever executed a task) for $worker_i$ and $task_j$ (denoted by $v_{i,j}$ in this paper) is equal to 1; else, $v_{i,j} = 0$. Afterwards, an m^*n matrix is obtained as in Eq. (1), where each entry is a Boolean value. Furthermore, the ith row of the matrix, i.e., $V_i = (v_{i,1}, \ldots, v_{i,n})$ represents the historical task execution records of $worker_i$. As we introduce in Section 1, V_i should be protected by the crowdsourcing platform *CP* as V_i often contains certain sensitive information of $worker_i$.

$$\begin{bmatrix} v_{1,1} & \cdots & v_{1,n} \\ \vdots & \ddots & \vdots \\ v_{m,1} & \cdots & v_{m,n} \end{bmatrix} \tag{1}$$

Next, we introduce how to transform the sensitive V_i ($1 \le i \le m$) data into the less sensitive hash value $H(V_i)$ through the Simhash technique. Here, we set parameter $r = \lceil log_2^n \rceil$. Thus each crowdsourcing task can be depicted by an r-dimensional vector. We demonstrate the calculation process with the following example. Assume there are 100 tasks and $worker_i$ has ever executed ten tasks (i.e., $task_1$, …, $task_{10}$) in the past, then the ten tasks can be denoted by the following seven-dimensional vectors, respectively.

$$\begin{aligned} &task_1 = (0,0,0,0,0,0,1) \\ &task_2 = (0,0,0,0,0,1,0) \\ &\ldots \\ &task_{10} = (0,0,0,1,0,1,0) \end{aligned}$$

Afterwards, for the above ten seven-dimensional vectors, we replace the element “0” by “− 1” (for example, $task_{10}$ is transformed from (0, 0, 0, 1, 0, 1, 0) to (− 1, − 1, − 1, 1, − 1, 1, − 1)) and then count the sum of each column of the matrix constituted by the above ten seven-dimensional vectors. Afterwards, we derive a seven-dimensional sum vector W as presented in (2). In (2), the negative values are substituted by “0”, the positive values are substituted by “1”. Afterwards, we obtain the new vector W' in (3). In our suggested $ExH_{Simhash}$ approach, W' is regarded as the index value of $worker_i$, i.e., $H(V_i) = W'$ holds. Thus through the above process, we transform the original 100-dimensional vector for $worker_i$, i.e., V_i into a seven-dimensional worker index $H(V_i)$. Therefore, the data scale is reduced significantly. Besides, the index values $H(V_i)$ employed for similar worker search are less sensitive; therefore, the workers’ private information is secure.

$$W = (-10, -10, -10, -4, -2, 0, 0) \tag{2}$$

$$W\prime = (0,0,0,0,0,0,0) \tag{3}$$

As the historical worker-task execution records are already recorded in the crowdsourcing platform *CP*, the worker indices building job can be finished offline. Therefore, the efficiency of subsequent exception handling process is improved significantly.

4.2 Step 2: search for the similar workers of unavailable worker

Next, we utilize the worker indices derived in step 1 to search for the similar workers of the unavailable worker

worker$_{excep}$. Concretely, as formulated in step 1, $H(V_{excep}) = (p_1, \ldots, p_r)$ and $H(V_i) = (q_1, \ldots, q_r)$. Thus their Hamming distance $d(H(V_{excep}), H(V_i))$ can be calculated by (4), where "x" means XOR. According to the nature of Simhash [9], if condition $d(H(V_{excep}), H(V_i)) < 3$, we can conclude that the tasks executed by *worker*$_i$ are approximately equal to the tasks executed by *worker*$_{excep}$; namely, *worker*$_i$ and *worker*$_{excep}$ are similar with high probability. Here, we utilize $WORKER_X$ to denote the similar worker set of the unavailable *worker*$_{excep}$; namely, the equation in (5) holds.

$$d\left(H\left(V_{excep}\right), H\left(V_i\right)\right) = \left(p_1 \mathrm{x} q_1, p_2 \mathrm{x} q_2, \ldots, p_r \mathrm{x} q_r\right) \quad (4)$$

$$WORKER_X = \left\{worker_i \mid worker_i \text{ is similar with } worker_{excep}\right\} \quad (5)$$

4.3 Step 3: optimal worker selection and replacement

In step 2, we have obtained the similar workers of the unavailable *worker*$_{excep}$, i.e., $WORKER_X$. The set $WORKER_X$ may contain multiple elements and each element is a qualified worker who can replace the unavailable *worker*$_{excep}$. Next, we evaluate and select an optimal candidate worker from set $WORKER_X$.

Concretely, for each *worker*$_x$ in set $WORKER_X$, we calculate his/her Hamming distance with *worker*$_{excep}$, i.e., $d(H(V_{excep}), H(V_x))$. According to the Simhash theory, a smaller Hamming distance $d(H(V_{excep}), H(V_x))$ often means a larger similarity between *worker*$_x$ and *worker*$_{excep}$. Therefore, we select the optimal (i.e., the most similar) worker, denoted by *worker*$_{optimal}$, from set $WORKER_X$ according to the Hamming distance; in other words, the equation in (6) holds. Specifically, if multiple workers have the minimal Hamming distance, then the crowdsourcing platform *CP* selects a worker randomly and utilizes the worker to replace the unavailable *worker*$_{excep}$. This way, the exception caused by the unavailable worker is handled successfully.

$$\begin{aligned} worker_{optimal} = \{ & worker_x \mid worker_x \in WORKER_X \\ & \text{and } d\left(H\left(V_{excep}\right), H\left(V_x\right)\right) \text{ is the minimal}\} \end{aligned} \quad (6)$$

5 Experiments

5.1 Experiment dataset and environment

We conduct a set of experiments based on the *WS-DREAM* [20] dataset to test the performances of *ExH*$_{Simhash}$. We compare the suggested *ExH*$_{Simhash}$ approach with three approaches, i.e., *UCF* (user-based CF), *ICF* (item-based CF), and *P-UIPCC* [21]. To evaluate the performances of different exception handling approaches, time cost and substitution equivalence are compared, respectively. If the ever-executed task sets of workers u and v are similar, then we can conclude that v is an ideal alternative for u when u becomes unavailable. Motivated by this hypothesis, a new criterion "substitution equivalence" ($\in$[0, 1], the larger the better) is defined as in (7), where A and B denote the set of tasks executed by *worker*$_{optimal}$ and *worker*$_{excep}$, respectively (Fig. 2).

$$\text{substitution equivalence} = \frac{|A \cap B|}{|A \cup B|} * 100\% \quad (7)$$

Our experiments are deployed on a PC with 2.40 GHz processor and 4.0 GB RAM. The operation system is Windows 7 and the programming language is JAVA. Experiments are executed ten times repeated and their average values are registered.

5.2 Experiment results

Next, three experiment profiles are designed, deployed, and tested, respectively. Please note that m and n represent the size of worker set and the size of crowdsourcing task set, respectively.

5.2.1 Profile 1: efficiency of different approaches

Here, we test the exception handling efficiency of different mobile crowdsourcing approaches. The following are the parameter settings: $m = \{50, 100, 150, 200, 250, 300\}$, $n = \{1000, 2000, 3000, 4000, 5000\}$. Concrete experiment results are presented in Fig. 3. As Fig. 3a ($n = 5000$ holds) and Fig. 3b ($m = 300$ holds) shows, the exception handling efficiencies of *UCF*, *ICF*, and *P-UIPCC* all decrease when m or n grows; this is because all the candidate workers and crowdsourcing tasks are recruited in the three exception handling approaches. While in *ExH*$_{Simhash}$, most jobs (e.g., worker indices building) can be finished offline before the crowdsourcing platform begins to handle the exception. Therefore, the exception handling efficiency of our approach performs better than the other three approaches, which means that *ExH*$_{Simhash}$ is suitable for the mobile crowdsourcing situations when a quick exception handling response is required by the task requester.

5.2.2 Profile 2: substitution equivalence of different approaches

Substitution equivalence is an important criterion to evaluate the performance of an exception handling approach. In this profile, we measure and compare the substitution equivalence values of different exception handling approaches. The following are the parameter settings: $m = \{50, 100, 150, 200, 250, 300\}$, $n = \{1000, 2000, 3000, 4000, 5000\}$. Experiment results are demonstrated in Fig. 4.

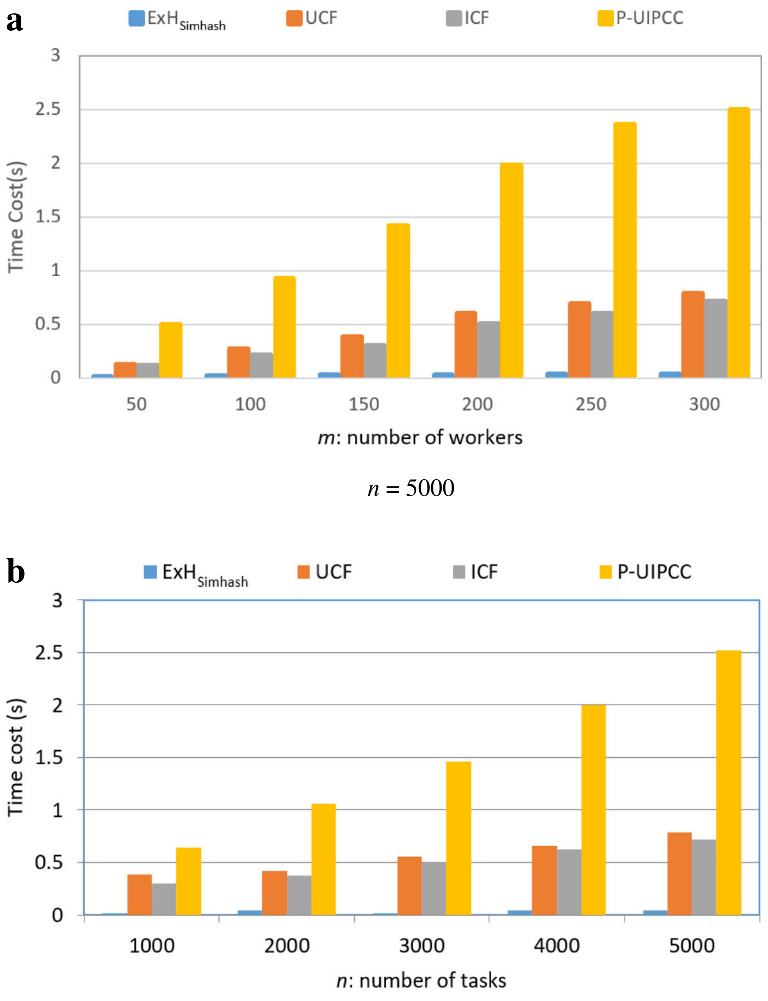
Step-1: Building worker indices offline based on Simhash. According to the historical crowdsourcing tasks ever-executed by workers, build worker indices offline based on Simhash.

Step-2: Search for the similar workers of unavailable worker. For the unavailable worker *worker*$_{excep}$, search for his/her similar workers *worker*$_x$ based on the indices built in Step-1.

Step-3: Optimal worker selection and replacement. From all the similar *worker*$_x$ obtained in Step-2, we select an optimal one based on the service quality and utilize him/her to replace the unavailable *worker*$_{excep}$.

Fig. 2 Three steps of our exception handling approach $ExH_{Simhash}$

As can be seen from Fig. 4a ($n = 5000$ holds) and Fig. 4b ($m = 300$ holds), the substitution equivalence of *P-UIPCC* is not as high as expected, because a set of approximate strategies are adopted to achieve the privacy-preservation goal. Due to the inherent tradeoff between data privacy and availability, an increment of privacy-preservation capability often leads to a drop of substitution equivalence. While in our $ExH_{Simhash}$ approach, the nature of Simhash technique can guarantee to find the workers who are similar with the unavailable worker and recruit them to replace the unavailable worker. Therefore, the substitution equivalence of the



$n = 5000$

$m = 300$

Fig. 3 Exception handling efficiency of different approaches w.r.t. m and n. **a** $n = 5000$. **b** $m = 300$

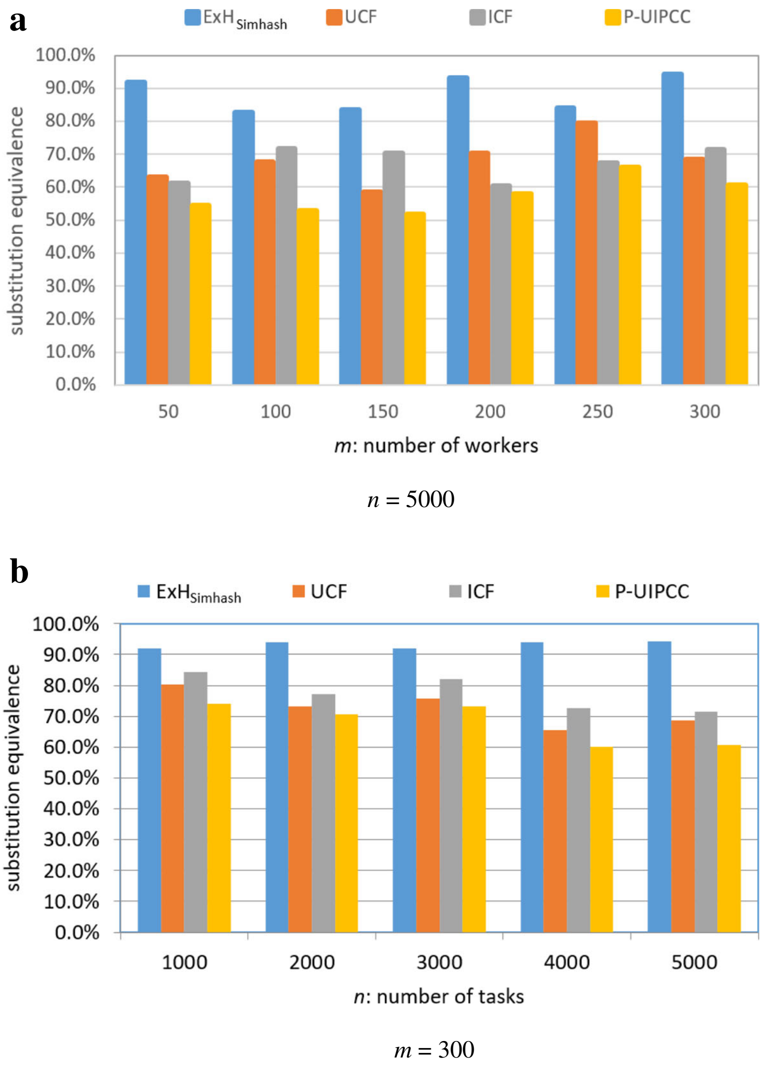


Fig. 4 Substitution equivalence of different approaches w.r.t. m and n. **a** $n = 5000$. **b** $m = 300$

$ExH_{Simhash}$ approach is improved accordingly and outperforms that of the two benchmark approaches, i.e., *UCF* and *ICF*.

5.2.3 Profile 3: successful rate of ExH$_{Simhash}$ with respect to m and n

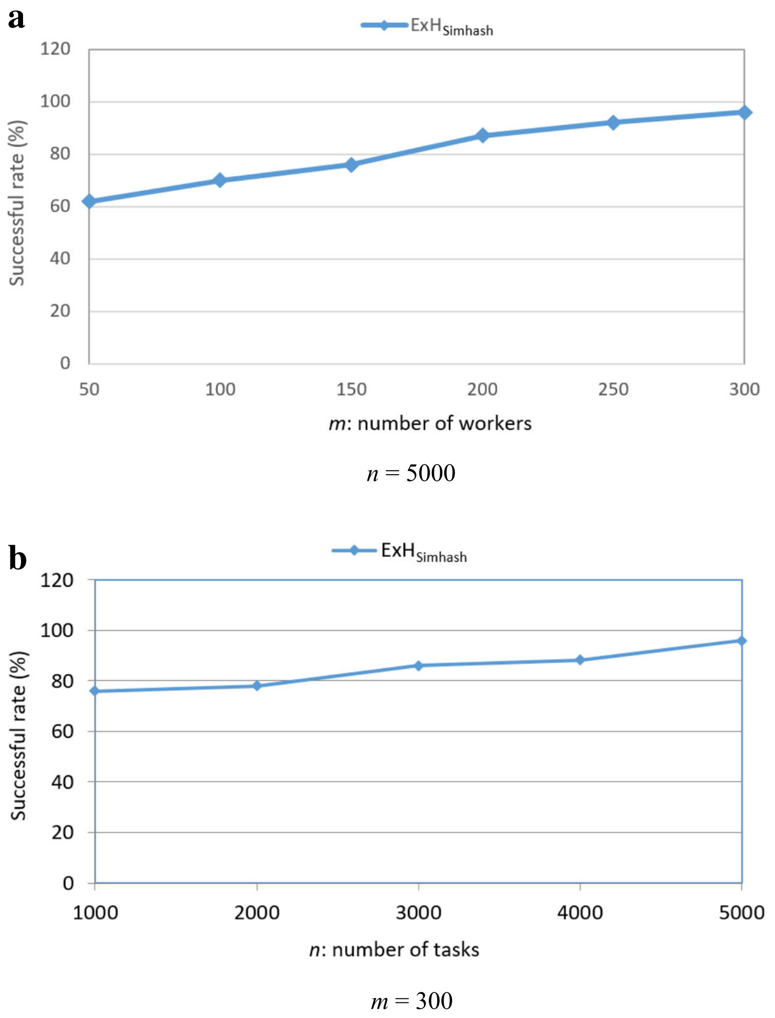
Simhash is essentially a probability-based search technique. So our suggested exception handling approach $ExH_{Simhash}$ cannot guarantee a 100% successful rate. Considering this, it is necessary to measure the successful rate of $ExH_{Simhash}$. Here, the successful rate is equal to the ratio between the successful replacement times and total replacement times. The following are the concrete parameter settings: $m = \{50, 100, 150, 200, 250, 300\}$, $n = \{1000, 2000, 3000, 4000, 5000\}$. Experiment results are demonstrated in Fig. 5.

As Fig. 5 indicates, there is an approximately linear relationship between the successful rate and the parameters m and n. This is due to the fact that when m or n increases, more workers who are similar with the unavailable worker can be returned to the crowdsourcing platform as the replacement candidates; accordingly, the successful rate of exception handling approach $ExH_{Simhash}$ is enhanced. Therefore, through tuning the parameters m and n, we can guarantee to maintain a relatively high successful rate in handling the exception.

5.3 Discussions

However, there are still several potential shortcomings in the experiments.

1. First, in the mobile crowdsourcing process, we only consider one evaluation criterion or dimension in



n = 5000

m = 300

Fig. 5 Successful rate of $ExH_{Simhash}$ w.r.t. *m* and *n*. **a** *n* = 5000. **b** *m* = 300

similar worker search, i.e., *whether a worker has executed a task*. While multi-dimensional application scenarios are more common in practical applications [22–34] where each dimension is assigned a weight value to indicate the significance [35–41].

2. Second, only the dimension with Boolean values is tested without considering the diversity of data types, such as the discrete data [42–52], fuzzy data [53–55], and continuous data [56–63], as well as their integration issues.
3. For simplicity, crowdsourcing task execution context is assumed to be fixed in this paper; in the future, we will improve our exception handling method to be more comprehensive and reasonable by considering more context factors, e.g., time factor that is crucial in many domains [64, 65].
4. Finally, due to the inherent shortcoming of Simhash technique, it is really hard to measure the privacy-preservation capability of the suggested exception handling method $ExH_{Simhash}$ in this paper; we will investigate this complex scientific problem in the future work.

6 Conclusions and future work

The ever-increasing popularity of mobile devices has created a variety of crowdsourcing applications by employing the massive and distributed mobile computing resources. However, in the mobile crowdsourcing process, a previously selected worker may become unavailable due to various exceptions. In this situation, it is significant for the crowdsourcing platform to quickly find another similar worker to replace the unavailable worker, in an efficient and privacy-preserving way. In view of this challenge, a novel privacy-preserving exception handling approach, named $ExH_{Simhash}$, is put forward in this paper. Finally, through a set of simulated experiments, we validate the feasibility of $ExH_{Simhash}$ in terms of substitution equivalence and computational time.

In the future, we will refine our exception handling approach by considering more context factors and multiple data types. Additionally, how to enhance the successful rate of the proposed exception handling method is still another research challenge that calls for intensive study.

Abbreviations

ICF: Item-based CF; UCF: User-based CF

Acknowledgements

Not applicable.

Funding

Partial work of this paper is supported by the Natural Science Foundation of China (No. 61872219).

Availability of data and materials

The recruited experiment dataset *WS-DREAM* is available at wsdream.github.io/.

Authors' contributions

YX finished the algorithm and English writing of the paper. HL finished the experiments. CY put forward the idea of this paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]Tianjin University, Tianjin 300354, China. [2]Qufu Normal University, Rizhao 276826, China.

Received: 5 January 2019 Accepted: 18 April 2019
Published online: 07 May 2019

References

1. J. Li, J. Li, X. Chen, C. Jia, W. Lou, Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans. Comput. **64**(2), 425–437 (2015)
2. L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, J. Chen, A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross- platform edge environment. Futur. Gener. Comput. Syst. **88**, 636–643 (2018)
3. S. Ding, S. Qu, Y. Xi, S. Wan, A long video caption generation algorithm for big video data retrieval. Futur. Gener. Comput. Syst. **93**, 583–595 (2019)
4. W. Gong, L. Qi, Y. Xu, Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wirel. Commun. Mob. Comput. **2018**, 8 (2018) ID 3075849
5. L. Qi, W. Dou, W. Wang, G. Li, H. Yu, S. Wan, Dynamic mobile crowdsourcing selection for electricity load forecasting. IEEE Access **6**, 46926–46937 (2018)
6. Y. Cheng, X. Zhou, S. Wan, K.K.R. Choo, Deep belief network for meteorological time series prediction in the internet of things. IEEE Internet Things J. (2018). https://doi.org/10.1109/JIOT.2018.2878477
7. Y. Xia, S. Qu, S. Wan, Scene guided colorization using neural networks. Neural Comput. & Applic., 1–14 (2018)
8. S. Wan, Y. Zhang, J. Chen, On the construction of data aggregation tree with maximizing lifetime in large-scale wireless sensor networks. IEEE Sensors J. **16**(20), 7433–7440 (2016)
9. S. Caitlin and G. Levin. Simhash: Hash-based similarity detection. Available: https://www.webrankinfo.com/dossiers/wp-content/uploads/simhash.pdf, 2007
10. L. Pu, X. Chen, J. Xu, *Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing*, IEEE International Conference on Computer Communications, San Francisco, CA, USA (2016), pp. 1–9
11. M. Karaliopoulos, O. Telelis, I. Koutsopoulos, *User recruitment for mobile crowdsensing over opportunistic networks*, IEEE Conference Computer Communications (2015), pp. 2254–2262
12. Y. Zeng, D. Li, A self-adaptive behavior-aware recruitment scheme for participatory sensing. Sensors **15**(9), 23361–23375 (2015)
13. W. Wang, H. Gao, C.H. Liu, Credible and energy-aware participant selection with limited task budget for mobile crowd sensing. Ad Hoc Netw. **43**, 56–70 (2016)
14. Z. Feng, Y. Zhu, Q. Zhang, *Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones*, IEEE 34th International Conference on Distributed Computing Systems, Madrid, Spain (2014), pp. 11–20
15. Y. Fan, H. Sun, X. Liu, *Truthful incentive mechanisms for dynamic and heterogeneous tasks in mobile crowdsourcing*, IEEE 27th International Conference on Tools with Artificial Intelligence, Vietri Sul Mare, Italy (2015), pp. 881–888
16. Y. Wei, Y. Zhu, H. Zhu, *Truthful online double auctions for dynamic mobile crowdsourcing*, IEEE Conference on Computer Communications, Kowloon, Hong Kong (2015), pp. 2074–2082
17. A. Singla, A. Krause, *Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. 22nd International Conference on World Wide Web, Toronto, ON* (ACM, Canada, 2013), pp. 1167–1178
18. D. Zhao, X.Y. Li, H. Ma, *How to crowdsource tasks truthfully without sacrificing utility: online incentive mechanisms with budget constraint*, IEEE Conference on Computer Communications (2014), pp. 1213–1221
19. D. Zhao, H. Ma, L. Liu, *Frugal online incentive mechanisms for crowdsourcing tasks truthfully. arXiv preprint arXiv:1404.2399* (2014)
20. wsdream.github.io/ .Accessed on 2018-04-11
21. J. Zhu, P. He, Z. Zheng, M.R. Lyu, *A privacy-preserving QoS prediction framework for web service recommendation*, IEEE International Conference on Web Services (2015), pp. 241–248
22. S. Wan, Y. Liang, Y. Zhang, Deep convolutional neural networks for diabetic retinopathy detection by image classification. Comput. Electr. Eng. **72**, 274–282 (2018)
23. L. Qi, J. Yu, Z. Zhou, An invocation cost optimization method for web services in cloud environment. Sci Program **2017**, 9 (2017) Article ID 4358536
24. S. Wan, Y. Zhao, T. Wang, Z. Gu, Q.H. Abbasi, K.K.R. Choo, Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things. Futur. Gener. Comput. Syst. **91**, 382–391 (2018)
25. M. Wang, G. Tian, Robust group non-convex estimations for high-dimensional partially linear models. J. Nonparametr. Statist. **28**(1), 49–67 (2016)
26. X. Wang, M. Wang, Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure. J. Appl. Stat. **43**(5), 796–809 (2016)
27. P. Wang, L. Zhao, Some geometrical properties of convex level sets of minimal graph on 2-dimensional riemannian manifolds. Nonlinear Anal. **130**, 1–17 (2016)
28. X. Wang, M. Wang, Adaptive group bridge estimation for high-dimensional partially linear models. J. Inequal.Appl. **2017**(158), 1–18 (2017)
29. X. Wang, S. Zhao, M. Wang, Restricted profile estimation for partially linear models with large-dimensional covariates. Statist. Probab. Lett. **128**, 71–76 (2017)
30. H. Tian, M. Han, Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems. J. Differ. Equ. **263**, 7448–7474 (2017)
31. P. Wang, X. Wang, The geometric properties of harmonic function on 2-dimensional Riemannian manifolds. Nonlinear Anal. **103**, 2–8 (2014)
32. M. Wang, X. Wang, Adaptive lasso estimators for ultrahigh dimensional generalized linear models. Statist. Probab. Lett. **89**, 41–50 (2014)
33. M. Wang, X. Wang, X. Wang, A note on the one-step estimator for ultrahigh dimensionality. J. Comput. Appl. Math. **260**, 91–98 (2014)
34. G. Tian, M. Wang, L. Song, Variable selection in the high-dimensional continuous generalized linear model with current status data. J. Appl. Stat. **41**, 467–483 (2014)
35. S. Yang, Z. Yao, C. Zhao, The weight distributions of two classes of p-ary cyclic codes with few weights. Finite Fields Appl. **44**, 76–91 (2017)
36. Y. Wang, C. Yin, X. Zhang, Uniform estimate for the tail probabilities of randomly weighted sums. Acta Math.Appl.Sin. Engl.Ser. **30**(4), 1063–1072 (2014)

37. S. Yang, Z. Yao, C. Zhao, A class of three-weight linear codes and their complete weight enumerators. Cryptogr. Commun. **9**, 133–149 (2017)
38. J. Cai, An implicit sigma (3) type condition for heavy cycles in weighted graphs. Ars Combin. **115**, 211–218 (2014)
39. S. Yang, Z. Yao, Complete weight enumerators of a family of three-weight linear codes. Des. Codes Crypt. **82**, 663–674 (2017)
40. S. Yang, Z. Yao, Complete weight enumerators of a class of linear codes. Discret. Math. **340**, 729–739 (2017)
41. S. Yang, X. Kong, C. Tang, A construction of linear codes and their complete weight enumerators. Finite Fields Appl. **48**, 196–226 (2017)
42. P. Li, Two classes of linear equations of discrete convolution type with harmonic singular operators. Complex Variables Elliptic Equ. **61**(1), 67–75 (2016)
43. Z. Zheng, Invariance of deficiency indices under perturbation for discrete Hamiltonian systems. J.Differ. Equ. Appl. **19**(8), 1243–1250 (2013)
44. M. Han, X. Hou, L. Sheng, C. Wang, Theory of rotated equations and applications to a population model. Discrete Continuous Dyn. Syst. A **38**(4), 2171–2185 (2018)
45. J. Cai, H. Li, A new sufficient condition for pancyclability of graphs. Discret. Appl. Math. **162**, 142–148 (2014)
46. L.L. Liu, B. Zhu, Strong q-log-convexity of the Eulerian polynomials of Coxeter groups. Discret. Math. **338**(12), 2332–2340 (2015)
47. H. Liu, F. Meng, Some new generalized Volterra-Fredholm type discrete fractional sum inequalities and their applications. Journal of Inequal. Appl. **2016**, 213 (2016) https://doi.org/10.1186/s13660-016-1152-7
48. P. Li, G. Ren, Some classes of equations of discrete type with harmonic singular operator and convolution. Appl. Math. Comput. **284**, 185–194 (2016)
49. P. Li, Singular integral equations of convolution type with hilbert kernel and a discrete jump problem. Adv.Difference Equ. **360**, 1–13 (2017)
50. Y. Bai, L. Liu, New oscillation criteria for second order delay differential equations with mixed nonlinearities. Discret. Dyn. Nat. Soc. **2010**, 1–9 (2010) ID 796256)
51. Y. Wang, C. Yin, Approximation for the ruin probabilities in a discrete time risk model with dependent risks. Stat. Probab.Lett. **80**, 1335–1342 (2010)
52. Q. Feng, F. Meng, Y. Zhan, Generalized gronwall bellman type discrete inequalities and their application. J. Inequal.Appl. **47**, 2011 (2011)
53. L. Wang, Intuitionistic fuzzy stability of a quadratic functional equation. Fixed Point Theory Appl. **2010**, 107182 (2010)
54. X. Du, Z. Zhao, On fixed point theorems of mixed monotone operators. Fixed Point Theory Appl. **2011**, 563136 (2011)
55. B. Zhu, L. Liu, Y. Wu, Local and global existence of mild solutions for a class of nonlinear fractional reaction-diffusion equations with delay. Appl. Math. Lett. **61**, 73–79 (2016)
56. C. Yan, X. Cui, L. Qi, X. Xu, X. Zhang, Privacy-aware data publishing and integration for collaborative service recommendation. IEEE ACCESS **6**, 43021–43028 (2018)
57. L. Qi, X. Zhang, W. Dou, Q. Ni, A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. IEEE J. Sel. Areas Commun. **35**(11), 2616–2624 (2017)
58. Y. Xu, L. Qi, W. Dou, J. Yu, Privacy-preserving and scalable service recommendation based on simhash in a distributed cloud environment. Complexity **2017**, 9 (2017) Article ID 3437854
59. C. Hou, Q. Meng, Continuity of (α,β)-derivations of operator algebras. J. Korean Math.Soc. **48**(4), 823–835 (2011)
60. C.Q. Ma, J.F. Zhang, On formability of linear continuous multi-agent systems. J. Syst. Sci. Complex. **25**(1), 13–29 (2012)
61. W. Helin, R. Rong, H. Feng, Continuous dependence property of BSDE with constraints. Appl. Math. Lett. **45**, 41–46 (2015)
62. L.L. Liu, Continued fractions and the derangement polynomials of types A and B. Ars Combin. **125**, 321–330 (2016)
63. H. Feng, The modulus of continuity theorem for G-Brownian motion. Commun.Stat.Theory Methods **46**(7), 3586–3598 (2017)
64. L. Qi, R. Wang, S. Li, Q. He, X. Xu, C. Hu, Time-aware distributed service recommendation with privacy-preservation. Inf. Sci. (2018). https://doi.org/10.1016/j.ins.2018.11.030
65. L. Qi, P. Dai, J. Yu, Z. Zhou, Y. Xu, "Time-location-frequency"-aware internet of things service selection based on historical records. Int.J.Distrib.Sens. Netw. **13**(1), 1–9 (2017)