



**MACQUARIE**  
University

# **Trust Management in Crowdsourcing Environments**

by

**Bin Ye**

A thesis submitted in fulfilment of  
the requirements for the award of the degree

**Doctor of Philosophy**

from

Department of Computing  
Faculty of Science and Engineering  
MACQUARIE UNIVERSITY

Supervisors: A/Prof. Yan Wang

Prof. Mehmet A. Orgun

2019

© Copyright by

Bin Ye

2019

---

# Statement of Candidate

---

I certify that the work in this thesis entitled “**Trust Management in Crowdsourcing Environments**” has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree to any other university or institution other than Macquarie University.

I also certify that the thesis is an original piece of research and it has been written by me. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged.

In addition, I certify that all information sources and literature used are indicated in the thesis.



---

Bin Ye

10 September 2019

*To my parents and my wife,  
who make me understand the true meaning of love.*

---

# Abstract

---

As a cost-effective model for solving problems, crowdsourcing has been widely applied in various human intelligence tasks, such as data labeling, data translation, and prediction. However, without adequate trust management, a large number of untrustworthy workers submit low-quality or even junk answers in the tasks to benefit themselves or sabotage their competitors' crowdsourcing processes. The disturbance or attacks not only significantly increase the cost of solving a task, but also drastically reduce the effectiveness of crowdsourcing processes. Therefore, selecting trustworthy workers to participate in tasks has become a top-priority demand in crowdsourcing environments. To achieve an effective trustworthy worker selection, three challenging sub-problems including *context-aware trust evaluation*, *spam worker defense*, and *trustworthy worker recommendation* have to be tackled. As such, in this thesis, we systematically propose our solutions for the three sub-challenges. The main contributions are summarized as follows:

- In a crowdsourcing platform, a worker's trustworthiness varies in different contexts, complicating the trust evaluation of a crowdsourcing worker. Thus, we propose a new context-aware trust model that evaluates a worker's trust in two primary crowdsourcing contexts, i.e., *the context of task type* and *the context of reward amount*, respectively. In particular, we first propose a task type taxonomy and a task reward amount taxonomy. Based on them, we devise two novel context-aware trust metrics: Task Type-aware Trust (TaTrust) and Reward Amount-aware Trust (RaTrust). Finally, we devise a multi-objective combinatorial optimization algorithm to effectively select trustworthy workers.
- To defend against the threats from the spam workers who masquerade themselves as "trustworthy" workers with "good" reputations by colluding with their

accomplices, we propose a new spam worker defense model based on our proposed Worker Trust Vector (WTV). A WTV consisting of the trust opinions from different requesters can indicate a worker’s global trust level. Based on the workers’ WTVs, we then propose an algorithm to effectively defend against spam workers.

- Moreover, to effectively and proactively identify spam workers, we propose a novel spam worker identification model. In this model, we first devise a novel worker trust representation called Worker Trust Matrix (WTM). A worker’s WTM is essentially a global trust feature set where each element is a local trust indicator called trust trace. A trust trace measures the extent to which a requester trusts a worker in a trust subnetwork centering on the requester. Taking the WTM as input, we then devise a learning-based algorithm to predict each worker’s identity. With our proposed WTM-based model, spam workers are precisely identified and then prohibited from participating in the tasks.
- Furthermore, we propose a novel trust-aware model to recommend trustworthy workers to participate in tasks. In this model, we tackle the homogeneous worker, dishonest behaviours, data sparsity, and cold start problems in generating worker recommendations. In particular, we first propose two similarity metrics to measure two requesters’ similarities in transacting with the workers they commonly trust and the workers they commonly distrust, respectively. Targeting the data sparsity problem, we propose a new trust sub-network extraction algorithm (TSE) to effectively discover requesters who can provide trustworthy recommendation suggestions. Finally, we suggest two strategies for solving the cold start problem.

All the models proposed in this thesis have been validated and evaluated through extensive experiments on real datasets or real scenarios. The results have demonstrated that the proposed models significantly outperform the comparable models in the existing studies in terms of effectively selecting trustworthy workers.

---

# Acknowledgments

---

The thesis would not have been accomplished without the efforts of many kind people who unselfishly contribute to my research in one way or another.

First of all, I would like to express my sincere thanks to my supervisors, A/Prof. Yan Wang and Prof. Mehmet A. Orgun for their insightful and patient supervision during my PhD journey. Over the past few years, their professional suggestions kept me researching in the correct direction. Working with them let me truly understand the dedication spirit and the rigorous work attitude. It is my great honour to have them as my supervisors at Macquarie University.

I would also appreciate the help from Prof. Ling Liu in various stages of my research. My academic journey would not have been so rewarding without her kindness and wisdom.

I wish to express my thanks to my colleagues, Dr. Guanfeng Liu, Dr. Haibin Zhang, Dr. Jun Zou, Dr. Lie Qu, and Dr. Xiaoming Zhen, for their valuable suggestions in my research and their friendly support in my life.

I would like to thank Melina Chan, Sylvian Chow, Jackie Walsh, Donna Hua, and Fiona Yang who have provided the most professional administrative support for all PhD students in the Department of Computing.

Most importantly, I would like to thank my parents, Quanxing Ye and Dianyin Li. Their unconditional love, support, and encouragement make me brave enough to pursue my dream. Many thanks to my lovely wife, Xinyu Xu, for her understanding and support. All their love and inspiration have supported me to accomplish this work.

---

# Publications

---

This thesis is based on the research work I have performed with the help of my supervisors and other colleagues during my PhD program at the Department of Computing, Macquarie University between 2014 and 2017. Some parts of my research have been published/submitted in the following papers:

- [1] **Bin Ye**, Yan Wang, and Ling Liu: CrowdDefense: A Trust Vector-Based Threat Defense Model in Crowdsourcing Environments, 24th IEEE International Conference on Web Services (IEEE ICWS 2017), pages 245-252. (**research track, acceptance rate 20%, CORE2017<sup>1</sup> Rank A**).
- [2] **Bin Ye** and Yan Wang: CrowdRec: Trust-aware Worker Recommendation in Crowdsourcing Environments, 23th IEEE International Conference on Web Services (IEEE ICWS 2016), pages 1-8. (**research track, acceptance rate 14%, CORE2017 Rank A**).
- [3] **Bin Ye**, Yan Wang, and Ling Liu: CrowdTrust: A Context-aware Trust Model for Worker Selection in Crowdsourcing Environments, 22nd IEEE International Conference on Web Services (IEEE ICWS 2015), pages 121-128. (**research track, acceptance rate 17.4%, CORE2017 Rank A**).
- [4] **Bin Ye**, Yan Wang, and Mehmet A. Orgun: Trust Network-Based Spam Worker Identification in Crowdsourcing Environments, completed and to be submitted.
- [5] Jun Zou, **Bin Ye**, Lie Qu, Yan Wang, Mehmet Orgun and Lei Li: A Proof-of-Trust Consensus Protocol for Enhancing Accountability in Crowdsourcing

---

<sup>1</sup>CORE refers to the Computing Research and Education Association of Australasia (<http://www.core.edu.au>).



Services: IEEE Transactions on Services Computing (TSC), accepted, 2018,  
**(Impact Factor: 3.52).**

---

# Contents

---

<b>Abstract</b>	<b>iv</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>Publications</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	3
1.1.1 What is Crowdsourcing? . . . . .	3
1.1.2 Who are Trustworthy Crowdsourcing Workers? . . . . .	5
1.1.3 Motivations for Selecting Trustworthy Workers . . . . .	6
1.2 Challenges of Selecting Trustworthy Workers . . . . .	9
1.2.1 Context-aware Trust Evaluation . . . . .	10
1.2.2 Spam Worker Defense . . . . .	11
1.2.3 Trustworthy Worker Recommendation . . . . .	13
1.3 Contributions of the Thesis . . . . .	15
1.4 Roadmap of the Thesis . . . . .	19
<b>2 Literature Review</b>	<b>21</b>
2.1 Background Knowledge of Crowdsourcing . . . . .	22
2.1.1 A New Categorisation of Crowdsourcing . . . . .	22
2.1.2 Work Patterns of Crowdsourcing . . . . .	26
2.1.3 Characteristics of Crowdsourcing . . . . .	29
2.2 Trust and Trust Influence in Crowdsourcing . . . . .	30
2.2.1 Meanings of Trust . . . . .	31
2.2.2 Properties of Trust . . . . .	34

---

2.2.3	The Influence of Trust in Crowdsourcing Activities . . . . .	38
2.3	Trustworthy Worker Selection in Crowdsourcing . . . . .	41
2.3.1	Trust Evaluation in Crowdsourcing . . . . .	42
2.3.2	Spam Worker Defense in Crowdsourcing . . . . .	45
2.3.3	Worker Recommendation in Crowdsourcing . . . . .	51
2.4	Conclusion . . . . .	54
<b>3</b>	<b>Trustworthy Worker Selection based on Context-aware Trust Evaluation</b>	<b>55</b>
3.1	Two-Dimensional Context-Aware Trust Evaluation . . . . .	56
3.1.1	Task Type-based Trust Evaluation . . . . .	56
3.1.2	Reward Amount-based Trust Evaluation . . . . .	60
3.2	A Worker Selection Algorithm based on Context-aware Trust . . . . .	63
3.2.1	Modelling Multi-Objective Worker Selection Problem . . . . .	64
3.2.2	A Multi-Objective Worker Selection Algorithm . . . . .	66
3.3	Experiments and Analysis . . . . .	68
3.3.1	Experiment Setting . . . . .	68
3.3.2	Performance Comparison in Trustworthy Worker Selection . . . . .	70
3.4	Conclusions . . . . .	72
<b>4</b>	<b>Spam Worker Defense based on Trust Vector</b>	<b>73</b>
4.1	Crowdsourcing Trust Network-based Threat Analysis . . . . .	76
4.1.1	Crowdsourcing Trust Network (CTN) . . . . .	76
4.1.2	Three Threat Patterns in a CTN . . . . .	78
4.2	Trust Vector-based Threat Defense . . . . .	82
4.2.1	Trust Pheromone-based Trust Inference . . . . .	82
4.2.2	The SOT Estimation Algorithm . . . . .	84
4.2.3	Worker Selection based on Worker Trust Vector (WTV) . . . . .	87
4.3	Experiments and Analysis . . . . .	90
4.3.1	Data Preparation . . . . .	90
4.3.2	Experiment Results . . . . .	92

---

4.4	Conclusion . . . . .	95
<b>5</b>	<b>Spam Worker Identification based on Trust Matrix</b>	<b>96</b>
5.1	Worker Trust Matrix (WTM) for Spam Worker Identification . . . . .	97
5.1.1	Spam Worker Identification Problem Formulation . . . . .	97
5.1.2	Worker Trust Matrix (WTM) . . . . .	98
5.2	WTM-based Spam Worker Identification Model . . . . .	103
5.2.1	WTM Estimation Algorithm . . . . .	104
5.2.2	Learning Algorithm CLnet-6 . . . . .	105
5.3	Experiments and Analysis . . . . .	107
5.3.1	Data Preparation . . . . .	107
5.3.2	Comparison Models . . . . .	109
5.3.3	Parameter and Measure Settings . . . . .	110
5.3.4	Experimental Results . . . . .	110
5.4	Conclusion . . . . .	114
<b>6</b>	<b>Trust-aware Worker Recommendation</b>	<b>115</b>
6.1	Trust-based Similarity Metrics . . . . .	117
6.1.1	Explicit Similarity Metrics . . . . .	117
6.1.2	Implicit Similarity Metrics . . . . .	120
6.2	A Trust-aware Recommendation Method in Crowdsourcing . . . . .	121
6.2.1	A Trust-based Similarity Network Extraction Algorithm . . . . .	122
6.2.2	Strategies for the Cold Start Problem . . . . .	124
6.2.3	Trust-aware Worker Recommendation . . . . .	125
6.3	Experiments and Analysis . . . . .	126
6.3.1	Experiment Settings . . . . .	127
6.3.2	Experiment Results . . . . .	129
6.4	Conclusion . . . . .	134
<b>7</b>	<b>Conclusions</b>	<b>135</b>

<b>A</b>	<b>The Notations in the Thesis</b>	<b>139</b>
<b>B</b>	<b>The Acronyms in the Thesis</b>	<b>145</b>

---

# List of Figures

---

1.1	A Typical Crowdsourcing Process . . . . .	4
1.2	A Typical Example of a Crowdsourcing Task . . . . .	7
2.1	A Discrete Crowdsourcing Process . . . . .	24
2.2	A Discrete Crowdsourcing Process . . . . .	24
3.1	An Intelligence Space for Human Intelligence Tasks Classification . . . . .	57
3.2	An Example of Differentiating Reward Amount-based Distance . . . . .	62
3.3	An Example of the Initial Worker Combinations . . . . .	65
3.4	The Comparison in TaTrust and RaTrust . . . . .	71
3.5	The Comparison in Trustworthy Worker Selection . . . . .	71
4.1	An Example of a Crowdsourcing Trust Network (CTN) . . . . .	78
4.2	An Example of Threat Pattern A . . . . .	79
4.3	An Example of Threat Pattern B . . . . .	80
4.4	An Example of Threat Pattern C . . . . .	81
4.5	The Comparison of Different Methods in Threat Pattern A . . . . .	93
4.6	The Comparison of Different Methods in Threat Pattern B . . . . .	93
4.7	The Comparison of Different Methods in Threat Pattern C . . . . .	94
4.8	The Comparison of Different Methods with Different Numbers of S- pam Workers . . . . .	94
5.1	A Worker Trust Matrix (WTM) encapsulates the global trust network- based features of the worker and can be exploited by learning algo- rithms for further prediction. . . . .	97
5.2	An Example of One-hop Sub-CTN . . . . .	100

---

5.3	An Example of a WTM for a Worker . . . . .	101
5.4	The Comparison of Test Errors of <i>WTM</i> -based CLnet-6 and <i>WTM'</i> -based CLnet-6 . . . . .	113
6.1	The Effectiveness Comparison of Different Recommendation Models on Homogeneous Workers . . . . .	129
6.2	The Effectiveness Comparison of Different Recommendation Models on Homogeneous Worker and Dishonest Behaviour Problems . . . . .	130
6.3	The Effectiveness Comparison of Different Models on Different Number of Dishonest Workers . . . . .	131
6.4	The Effectiveness Comparison of Different Models on Homogeneous Worker, Dishonest Behaviour and Data Sparsity Problems . . . . .	132
6.5	The Comparison of Different Methods on Homogeneous Worker, Dishonest Behaviour, Data Sparsity, and Cold Start Problems . . . . .	133

---

# List of Tables

---

3.1	Constraints for Generating 1000 Workers . . . . .	68
3.2	Parameters for the Multiple-Objective Worker Selection Algorithm . .	69
4.1	The Dishonest Participants in Experiment1 . . . . .	92
4.2	The Dishonest Participants in Experiment2 . . . . .	92
5.1	The Comparison of Different Models in Identifying Workers with D- ifferent Identities . . . . .	111
6.1	The Number of Different Participants in a Simulated Crowdsourcing Environment . . . . .	127
6.2	Effectiveness of Different Recommendation Models Under Homoge- neous Worker Problem . . . . .	129
6.3	Effectiveness of Different Recommendation Models Under Homoge- neous Worker and Dishonest Behaviour Problems . . . . .	130
6.4	Effectiveness of Different Recommendation Models Under Homoge- neous Worker, Dishonest Behaviour, and Data Sparsity Problems . . .	132
6.5	Effectiveness of Different Recommendation Models under Homoge- neous Worker, Dishonest Behaviour, Data Sparsity and Cold Start Prob- lems . . . . .	133
A.1	The Notations in Chapter 3 . . . . .	139
A.2	The Notations in Chapter 3 (continued) . . . . .	140
A.3	The Notations in Chapter 4 . . . . .	140
A.4	The Notations in Chapter 4(continued) . . . . .	141
A.5	The Notations in Chapter 5 . . . . .	142



---

A.6	The Notations in Chapter 5 (continued) . . . . .	143
A.7	The Notations in Chapter 6 . . . . .	143
A.8	The Notations in Chapter 6 (continued) . . . . .	144
B.1	The Acronyms in All the Sections . . . . .	145

# Chapter 1

---

## Introduction

---

Crowdsourcing is a novel problem-solving model that utilizes human intelligence to cost-effectively complete complex tasks in the form of an open call [125]. In recent years, the considerable success of crowdsourcing platforms, e.g., CrowdFlower<sup>1</sup>, FreeLancer<sup>2</sup>, and Amazon Mechanical Turk<sup>3</sup>, has demonstrated the superior effectiveness and the immense potential of crowdsourcing in solving various tasks. In particular, CrowdFlower mainly focuses on tasks from the domain of data processing, such as collecting data, cleaning data, and labelling data. In 2011, CrowdFlower had managed its one-millionth worker and had successfully organized workers to accomplish more than 100 million tasks<sup>4</sup>. FreeLancer is a popular crowdsourcing platform in Australia, which focuses on more comprehensive tasks, such as software development, data collection, and article writing. By 2017, the number of employers who had ever participated in the tasks at FreeLancer has exceeded 25,483,234<sup>5</sup>. Amazon Mechanical Turk is one of the most well-known crowdsourcing platforms around the world. It engages a diverse, on-demand and scalable workforce to tackle more than ten thousands of tasks every day. By 2015, there are over 500,000 workers who had signed up at Amazon Mechanical Turk [145]. Moreover, according to the analysis in [78], the crowdsourcing market is projected to be gross in the range of \$15 billion to \$25 billion by 2020.

---

<sup>1</sup><https://www.crowdfLOWER.com/>

<sup>2</sup><https://www.freelancer.com/>

<sup>3</sup><https://www.mturk.com/>

<sup>4</sup><http://venturebeat.com/2011/05/17/crowdfLOWER-100-million-tasks/>

<sup>5</sup><https://www.freelancer.com/about>

With the rapid development of the crowdsourcing market, not only the trustworthy workers but also the untrustworthy workers such as low-quality workers and even attackers, are being attracted to participate in crowdsourcing. These untrustworthy workers submit valueless and even malicious answers to the crowdsourcing tasks published by honest requesters to maximize their benefits or sabotage others' crowdsourcing processes. First, a low-quality worker may participate in as many tasks as possible for increasing its opportunities for earning the reward from a task. In general, a low-quality worker submits a valueless answer in a task, which undermines the effectiveness of crowdsourcing. Second, an attacker may either submit junk answers to a task for cheating the reward with a small probability or collude with its accomplices to uniformly submit the same answer in a task to manipulate the crowdsourcing outcome of the task. The existence of these untrustworthy workers significantly increases the cost of solving a crowdsourcing task and drastically reduces the effectiveness of crowdsourcing processes, which will finally lead crowdsourcing to a failure. In order to eliminate the negative impacts of the untrustworthy workers, selecting trustworthy workers to participate in tasks has become a top-priority demand in crowdsourcing environments. However, in the literature, few studies focusing on the trustworthy worker selection problem have been proposed in crowdsourcing environments.

In this thesis, we leverage trust management techniques to select trustworthy workers to participate in crowdsourcing tasks. In order to thoroughly understand the nature and the necessity of selecting trustworthy worker in crowdsourcing environments, we first present what crowdsourcing exactly is, who are trustworthy workers in crowdsourcing environments, and the motivations for studying on the trustworthy worker selection issue in crowdsourcing environments. Based on them, we then present three particular challenging problems that are needed to be tackled for achieving an effective trust management. The three challenges are (1) context-aware trust evaluation, (2) spam worker defense, and (3) trustworthy worker recommendation, respectively. They will be correspondingly solved in this thesis.

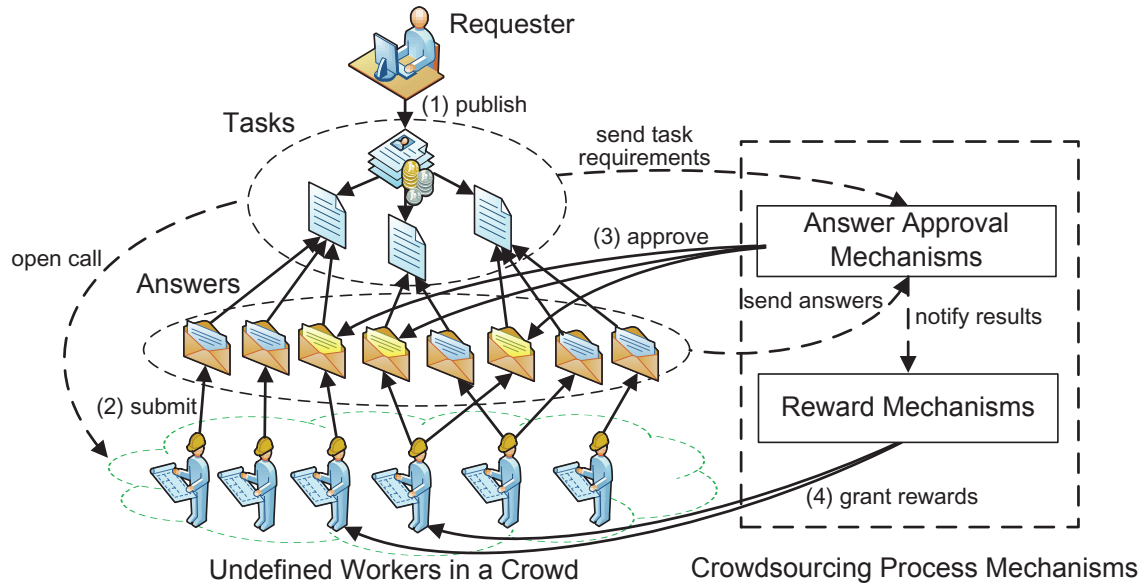
---

## 1.1 Problem Statement

### 1.1.1 What is Crowdsourcing?

The term of Crowdsourcing is first coined by Jeff Howe in a Wired Magazine article in June 2006 [58], which is defined as “*an act of outsourcing a function once performed by traditional task performers (such as an employee or a contractor) to an undefined (and generally large) network of people in the form of an open call*” [57]. Considering the specific applications of crowdsourcing, Daren C. Brabham describes crowdsourcing as a process: a company posts problems online, then a vast number of individuals offer solutions to the problem and finally the winning ideas are awarded in some form of a bounty [17]. Essentially, the comprehension of crowdsourcing is that everyone has the potential to provide valuable information for solving a problem [50]. In addition, among the early definitions of crowdsourcing, competition is commonly regarded as the core of crowdsourcing [38]. However, with the more complex crowdsourcing tasks appearing, the collaboration becomes a new keyword in defining crowdsourcing [33]. For example, regarding the crowdsourcing tasks like reconstructing shredded documents and jigsaw puzzles, the collaboration-based working pattern is a unique identifier that differentiates the crowdsourcing model for solving these crowdsourcing tasks from the others. Moreover, in the study of Estells-Arolas and Gonzalez-LadroneGuevara [40], 40 definitions of crowdsourcing are listed from various perspectives.

Though the existing definitions of crowdsourcing have many differences due to different concerns, we can conclude six commonly essential elements of crowdsourcing. They are (1) requester (service demander), (2) human intelligence task, (3) worker (service provider), (4) the form of an open call, (5) answer approval mechanism, and (6) reward mechanism, respectively. Accordingly, in this thesis, we define crowdsourcing as *a problem-solving model where a requester (service demander) publishes a group of tasks to be available for all the undefined workers (service providers) in the form of an open call, then a worker whose answer is approved in a task by the answer approval mechanism (such as a voting-based mechanism or a verification-based*



**Figure 1.1:** A Typical Crowdsourcing Process

*mechanism) will be awarded based on the reward mechanism.*

Based on our proposed definition of crowdsourcing, we can further decompose a typical crowdsourcing process into four procedures: (1) publishing task, (2) submitting answer, (3) approving answer, and (4) granting reward. As shown in Fig. 1.1, at the beginning of a crowdsourcing process, a requester first publishes tasks to all the workers in the form of an open call. Then, any worker who is interested in a published task can submit its answer to the task. Afterwards, a preset answer approval mechanism is executed to judge which answer or answers should be approved. If a worker's answer is approved in a task, the worker is granted the reward of the task according to the deployed reward mechanism. In general, when facing different types of tasks, different answer approval mechanisms and reward mechanisms will be applied for adapting the specific tasks. For example, the voting-based consensus mechanisms and the verification-based consensus mechanisms are two types of the typical answer approval mechanisms in crowdsourcing environments. In addition, the one winner-oriented reward mechanisms and the multiple winners-oriented reward mechanisms are two types of the commonly applied reward mechanisms.

### 1.1.2 Who are Trustworthy Crowdsourcing Workers?

In different environments, the trust may be very differently defined to precisely explain the specific changes of both the relation between a trustor and a trustee and the corresponding behaviours of them. Thus, when we discuss the trustworthiness of workers in crowdsourcing environments, one of the most important things is to exactly define *who is a trustworthy worker*. Referring to the most classic trust definition in the Oxford Dictionary, trust is a “firm belief in the reliability, truth, or ability of someone or something”<sup>6</sup>. Based on this definition, we suggest that a *trustworthy worker* in crowdsourcing environments should be a worker who is *reliable*, *truthful*, and *capable*. Accordingly, we below present the basic criteria for judging if a worker is reliable, truthful or capable.

- A reliable worker should be the one who responsibly works in any task and thus will not lightly submit an uncertain answer or a low-quality answer in a task.
- A truthful worker should be the one who honestly participates in any crowdsourcing activity and never takes any fraud actions to make itself be more competitive than other workers.
- A capable worker should be the one who possesses the skills for solving a task and thus will submit a reliable and nearly correct answer in the task with a very high probability.

In this thesis, we mainly focus on discussing the solutions for effectively selecting the workers who can satisfy the above three criteria. In the following subsection, by introducing the specific examples of different workers, we will explain the reasons why selecting the trustworthy workers (i.e., the reliable, truthful, and capable workers) is the key to guarantee the cost-effectiveness of crowdsourcing.

---

<sup>6</sup><http://en.oxforddictionaries.com/definition/trust>

### 1.1.3 Motivations for Selecting Trustworthy Workers

In most of the crowdsourcing tasks, e.g., data transcribing tasks and image tagging tasks, we cannot directly verify if a submitted answer is correct due to the lack of the ground truth. As such, crowdsourcing is applied to estimate the ground truth with a low cost. In fact, a crowdsourcing process takes an answer or answers approved by the majority of participants as the approximate ground truth. The essence of a crowdsourcing process determines that a worker is preferred to participate in a task if he/she is most likely to submit an answer closing to the ground truth. In general, the probability that a worker provides such an accurate answer positively relates to the worker's trustworthiness. In other words, the trustworthiness of all the workers who participate in a task jointly determine if the task can be successfully solved. Selecting trustworthy workers to participate in tasks can significantly improve the cost-effectiveness of crowdsourcing in solving various tasks. Therefore, we need to find an effective way to make an accurate judgement on a worker's trustworthiness.

As we have discussed in the last subsection, a trustworthy worker is not a simple concept in crowdsourcing environments. In particular, we suggest that a trustworthy worker should be *reliable*, *truthful*, and *capable*. Below, by taking a data transcribing task as an example, we introduce three typical scenarios to clarify the meanings of a trustworthy worker, and reasons why selecting trustworthy workers is necessary. As depicted in Fig. 1.2, the data transcribing task requires each participating worker to transcribe the text from the given picture to the text box. In this task, there are five workers Aaron, Bob, Clark, Daniel, and Eden who have submitted their answers. To clarify the three scenarios **S1-S3** that motivate our work, here, we first state the ground truth of the textual content in the picture, i.e., *V. Hhicast Broken Arrow 74012 Tuls*.

**S1:** As shown in Fig. 1.2, Aaron is a worker who does not lie but submits an ambiguous answer to a task that cannot be solved by him easily. In particular, in the data transcribing task, Aaron gives out the answer: *V. Hhicast Abcde Arow T4012 Turk*. Naturally, Aaron correctly transcribes some parts of the picture.

**Transcribe data**

**Requester:** p9r **Reward:** \$0.01 per task

**Qualification Required:** Answer approval rate (%) is greater than 90

- Copy the text from the picture into the box below it according to the Special Instruction at the top of the page.
- If there are no Special Instructions, copy all text **exactly as it appears** the picture.
- Copy only what appears inside the **pink box**.

City      Zip      County

V. Hhica st Broken Arrow 74012 Tuls

**Aaron**  
V. Hhicast Abcde Arow T4012 Turk

**Bob**  
V.Hhicast Brotcn Arrow T4012 Tuls

**Eden**  
V.Hhicast Broken Arrow 74012 Tuls

**Clark**  
V. Hhicast Broken Arrow 14012 Tuls

**Daniel**  
V. Hhicast Broken Arrow 14012 Tuls

**Figure 1.2:** A Typical Example of a Crowdsourcing Task

But, he randomly answers “*Abcde Arow*” in the blurred part that is difficult to be recognized. Aaron is not dishonest, however, he is unreliable because his answer is ambiguous. By contrast, Bob is more reliable than Aaron because he conscientiously works in the task though he also does not correctly solve the task. In addition, given some pictures that are difficult for the vast majority of workers to transcribe, workers like Aaron will quickly and irresponsibly submit ambiguous answers based on their knowledge. In such a case, it is more likely for these workers to form a consensus on an ambiguous answer than to find out the correct answer. As a result, workers may find that they can maximize their profits by submitting ambiguous answers in as many tasks as possible [153, 137]. On the other side, requesters will believe that crowdsourcing is cheap but can only return the low-quality solutions, which will severely dampen their participating enthusiasm. Without an effective trustworthy worker selection, a large number of ambiguous answers will be submitted to a crowdosurcing task, which finally undermines the interest of both requesters and workers. Therefore, when facing workers who are not dishonest, how to judge who are reliable is the first task of



selecting trustworthy workers in crowdsourcing environments.

**S2:** As shown in Fig. 1.2, Clark and Daniel are two workers who aim to mislead the requester to adopt an incorrect answer, i.e., *V. Hhicast Broken Arrow 14012 Tuls*, as the correct answer. In particular, they uniformly give a wrong zip code: “14012”. Based on our given ground truth, we know that the “74012” is a correct zip code and also is the most critical information of the picture. However, the “14012” is very confusing for the workers who do not well know the zip codes in America. In common sense, a zip code starts with a small integer, and thus the “14012” may mislead other workers to vote for it, e.g., worker Bob in the task. In addition, Clark and Daniel will vote for their answers to get two votes of the total five votes in this example. As a result, it is most likely for Clark and Daniel to manipulate the crowdsourcing result of such a task under the voting based consensus mechanisms. In recent years, though crowdsourcing platforms tend to organize the irrelevant workers or the specified verifiers to vote for answers, the availability of high-degree anonymity still enables an attacker to create many worker accounts and let them behave like Clark and Daniel to manipulate the voting results. A well-known attack mounted by anonymous spam workers was reported in the DARPA Shredder Challenge 2013 where the team that completed five jigsaw puzzles first would win the prize of US \$50,000. In this attack, anonymous spam workers sabotaged the crowdsourcing processes of a team from University of California San Diego who had reached the second place [126]. In particular, no other jigsaw puzzle was successfully assembled after their crowdsourcing processes underwent the attack from numerous spam workers. Thus, it is vital to judge if a worker is truthful. In other words, we need to defend against the threats of the spam workers who are created by someone else.

**S3:** As shown in Fig. 1.2, Bob submits an answer: “*V. Hhicast Brotcn Arrow T4012 Tuls*”, while Eden submits an answer “*V. Hhicast Broken Arrow 74012 Tuls*”.

Based on our given ground truth “*V. Hhicast Broken Arrow 74012 Tuls*”, it is evident that the quality of the answer submitted by Eden is much better than that of Bob . However, in crowdsourcing environments, there lacks an effective method to differentiate the workers like Bob from the workers like Eden because all workers participate in a task based on the “first-come-first-serve” [53]. Though Bob and Eden are both reliable and truthful, recommending more workers like Eden to the task can significantly improve the probability of finally obtaining a correct answer. As such, a demand arises here is that we need to select those most capable workers to participate in a particular task. Moreover, in complicated crowdsourcing tasks like video annotating tasks, most crowdsourcing workers have poor performance [135]. Without a mandating control on the capability of workers, these crowdsourcing tasks cannot be effectively solved.

Based on the scenarios **S1-S3**, we can conclude that an adequate trust management needs to ensure trustworthy workers can be selected to participate in tasks for guaranteeing the correctness of the outcome in a crowdsourcing process. Regarding how to judge if a worker is trustworthy, three aspects of trust has to be considered. They are the *reliability*, *truthfulness*, and *capacity*, respectively. However, in the literature, there are very few studies that systematically investigate the impacts of the reliability, the truthfulness, and the capacity on selecting trustworthy workers in crowdsourcing environments. Therefore, in this thesis, we bridge the gap between the demand of selecting trustworthy workers and the existing works in crowdsourcing environments.

## 1.2 Challenges of Selecting Trustworthy Workers

Considering the significance of the *reliability*, *truthfulness*, and *capacity* in selecting trustworthy workers, three particular challenging problems are discussed in this section. They are *Context-aware Trust Evaluation*, *Spam Worker Defense*, and *Trustworthy Worker Recommendation*, respectively.

### 1.2.1 Context-aware Trust Evaluation

As we have discussed in the scenario **S1**, a worker can irresponsibly and quickly submit ambiguous answers in as many tasks as possible to maximize its profit if there is no effective mechanism to prevent it from participating in tasks [109, 137]. The success of these untrustworthy workers drastically reduces the participating enthusiasm of both the trustworthy workers and requesters, leading the reliability of a crowdsourcing platform to be severely damaged [55]. As such, we need to build a model to effectively evaluate a worker's trust and accordingly permit trustworthy workers to participate in tasks. However, it is very challenging because the trustworthiness of a worker varies in different contexts in crowdsourcing environments. For example, in a programming task, a software engineer is commonly more trustworthy than a constructing engineer. In addition, a worker may more considerably work in a task with a solid reward than in a task with a low-amount reward. Thus, the basis of an effective trust evaluation is to model the impacts of the new crowdsourcing contexts on the reliability and capability of a worker.

To date, many of crowdsourcing platforms, e.g., Amazon Mechanical Turk and CrowdFlower, apply a worker's *overall answer approval rate* in the historical tasks or an overall answer approval rate-based trust level to indicate a worker's trust degree in future tasks [90]. In particular, a worker's overall answer approval rate is the percentage of the accepted answers in all the answers submitted by the worker. Using this rate to measure a worker's trust is intuitive and thus can easily be accepted by participants. However, it neglects the influences of the contextual factors in a worker's reliability and capacity, leading the trust evaluation to become ineffective. For example, given two workers who have the same overall answer approval rate, while one of them has significantly better performance in translating German literature to English version than the other one. If these two workers simultaneously participate in a German article translation task, their trustworthiness cannot be differentiated by merely using the overall answer approval rate. In fact, discussing their performance in the

---

task type: German Translation is more meaningful as it enables to indicate which of the two workers is more reliable and capable in a specific task. In addition, given a worker who possesses a high overall answer approval rate in many easy tasks, without taking the contexts into account, the worker is permitted to participate in tasks and thus can pursue the maximal profit in some high-value tasks by submitting as many answers as possible in the high-value tasks. Regarding such a worker, differentiating its trustworthiness in the tasks with different reward amounts is more effective than using the overall answer approval rate.

In the literature, from different perspectives, several trust evaluation models have been proposed in crowdsourcing environments. For example, in [74, 39, 38], some qualitative trust evaluation models are introduced. However, one common drawback of these studies is that no quantitative standards can be well designed to differentiate workers' trustworthiness. Thus, these models are hardly to be deployed in practice. In addition, some quantitative trust models focusing on particular crowdsourcing applications are proposed. However, in the existing studies, e.g., [153, 55], the impacts of contextual factors on a worker's trustworthiness still do not receive sufficient attention. In fact, context-aware trust evaluation has been proved to be more effective than general trust evaluation in many traditional online systems, e.g., e-commerce, social networks, and cloud-based systems [140, 86, 160, 114]. Unfortunately, in crowdsourcing environments, few studies are proposed to investigate the contexts and the impacts of the contexts on a worker's trust. Thus, we first need to define the contexts in crowdsourcing environments. Then, we need to rationally model context-aware trust metrics with the reasonable physical meanings that can reflect the impacts of contexts on a worker's trustworthiness.

### **1.2.2 Spam Worker Defense**

In a crowdsourcing task, an answer is approved based on the consensus achieved by the voters in the task. However, the correctness of a voting-based consensus is ex-

tremely vulnerable to the spam workers who intentionally submit random answers and erroneous answers to the task. As we have discussed in the scenario **S2**, a sufficient number of spam workers can manipulate an incorrect answer, which has been uniformly submitted by them in a task, to be approved under voting-based consensus mechanisms [115]. Moreover, spam workers can also cheat for rewards by submitting random answers to as many tasks as possible because a random answer still has a probability to be approved and then be rewarded, especially in the multi-choice tasks [136]. Different from an irresponsible worker, a spam worker dishonestly behaves in all tasks and colludes with its accomplices in some tasks for counterfeiting “good” reputations to whitewash itself. The success of these spam workers drastically reduces the enthusiasm of honest participants and the reliability of crowdsourcing platforms [55].

Defending against the threats from these spam workers is a top-priority issue in crowdsourcing environments, however, it is incredibly challenging because spam workers could leverage the low transaction fee [17, 21, 98, 22] and the availability of high-degree anonymity [155] in crowdsourcing environments to masquerade themselves as “trustworthy” workers via low-cost collusions [5]. In general, a spam worker can obtain two types of guises **G1** and **G2** via collusions:

- **G1:** a spam worker can obtain “successful” transaction records, and thus possess a “good” reputation, by colluding with some requesters to manipulate the crowdsourcing outcomes in *shadow tasks*. A shadow task is one whose answer is preset and revealed to the colluding spam workers beforehand to ensure that they can succeed in the task.
- **G2:** a spam worker can collude with some requesters and workers who are directly trust by other honest participants, and thus indirectly links itself to the honest region consisting of honest participants via the edges between their colluding partners and the honest participants. Such an edge is called an *attack edge* because the spam worker can leverage it to deceive honest requesters and

then mount attacks in the tasks published by such requesters.

In the literature, in order to defend against the spammers in general online applications, e.g., e-commerce or P2P file-sharing systems, two categories of trust-aware defense models have been widely discussed. They are *reputation-based defense models* and *trust network-based defense models*, respectively. In recent years, particularly targeting spam workers in crowdsourcing environments, a few reputation-based defense models have been proposed, e.g., a sequential performance-based defense model [153] and a consistency-based defense model [64]. The common intuition of these models is that a worker’s historical transaction records are truthful and thus can genuinely indicate the worker’s trustworthiness in future tasks. However, these models are vulnerable to the spam workers who possess many “successful” transaction records (i.e., guise **G1**). Essentially, the reason is that these reputation-based defenses only pay attention to the *reliability* and *capacity* of a worker, but neglect to investigate the *truthfulness* of a worker’s reputation. With respect to the trust network-based defense models, to the best of our knowledge, no study has been proposed to target spam workers with both the guise **G1** and the guise **G2** in crowdsourcing environments. Nevertheless, in P2P networks, trust network-based defense models, such as SybilLimit [151] and SybilInfer [27], have shown the high effectiveness in limiting spammers with guise **G1** when a spammer only possesses few attack edges. However, in crowdsourcing environments, a spam worker may possess many attack edges (i.e., guise **G2**), which leads these trust network-based defense models to be ineffective and thus make effective spammer defense more challenging. Therefore, a new model has to be devised for effectively identifying the spam workers with both the guises **G1** and **G2**.

### 1.2.3 Trustworthy Worker Recommendation

As we have discussed in the scenario **S3**, the capacity of two reliable workers may be different and thus causes the difference of their performance in a task. In general, a crowdsourcing task is published in the form of an open call, thus all the reliable work-

ers fairly compete for participating in a task following the first-come-first-serve basis. In other words, all these workers have an equal opportunity to obtain the opportunity to participate in a task even if some of them are more likely to submit an answer closing to the ground truth in the task. A trustworthy worker should be a capable worker, we suggest that a worker should be recommended to a task if its capacity can support it to submit a high-quality answer to the task.

Aiming to discover trustworthy workers and then recommend them to appropriate tasks, in the literature, several *worker-oriented recommendation models* have been proposed. In general, a *worker-oriented recommendation model* recommends tasks to a worker based on the worker's interests in different tasks [7, 37, 156]. One common drawback of these recommendation models is that a recommended worker may not be the most suitable one who can correctly complete this task. Instead, a recommended worker may be the one who is only interested in the task but cannot well solve the task. Moreover, even the reliable and truthful workers may tend to overstate their skills to gain more opportunities in participating tasks. As a result, a *worker-oriented recommendation model* may actually decrease the quality of the answers in a task because those genuinely appropriate workers may miss the task in the vast task pool. Considering the drawbacks of the *worker-oriented recommendation models*, we suggest that recommending trustworthy workers to a task published by a requester is more meaningful in improving the quality of answers rather than recommending appropriate tasks to a worker. In this thesis, we call this type of recommendation models as *requester-oriented recommendation model*. A requester-oriented recommendation model judges if a worker is suitable to participate in a task published by a requester by taking its capacity as one of the most critical indicators. In the design of a requester-oriented recommendation model, we need to take four problems that profoundly influence the quality of recommendations into account.

- **Homogeneous Worker:** There are a large number of homogeneous workers have an identical opportunity to be recommended as they all possess good rep-

---

utations. Even with an effective context-aware trust evaluation, there are still many similarly trustworthy workers who are needed to be more precisely differentiated in a task published by a particular requester.

- **Dishonest Behaviours:** Workers may boost good reputations in easy tasks and also overstate their skills in their registered files to obtain more opportunities to be recommended.
- **Data Sparsity:** In crowdsourcing environments, as a requester may transact with a tiny fraction of all the workers, the requester-worker matrix where each element records a transaction record between a requester and a worker is very sparse.
- **Cold Start:** As there is nearly no information available about freshly registered requesters and freshly registered workers, it is very difficult to generate recommendations for them.

To sum up, in a task published by a requester, given a worker who is reliable and truthful but has not transacted with the requester, the target of a requester-oriented worker recommendation model is to predict the capacity/performance of the worker in the task by using the limited and sparse available data.

## 1.3 Contributions of the Thesis

With the sprawl of crowdsourcing, the demand for selecting trustworthy workers emerges and becomes prominent in crowdsourcing environments. This thesis aims to bridge the enormous gap between the demand for effectively selecting trustworthy workers and the existing studies. In particular, this thesis systematically discusses the solutions for selecting trustworthy workers by taking the three aspects of a crowdsourcing worker's trust into account, i.e., the reliability, the truthfulness, and the capacity. The main contributions of this thesis contain four significant aspects.



---

1. The first contribution of the thesis is a new context-aware trust-based crowdsourcing worker selection model called CrowdTrust.

(a) In CrowdTrust, we first propose two context-aware trust metrics: Task Type-aware Trust (TaTrust) and Reward Amount-aware Trust (RaTrust) to evaluate a worker's trustworthiness in two contextual dimensions. The TaTrust of a worker is modelled by aggregating the worker's past performance in all the different types of tasks. The RaTrust of a worker is modelled by aggregating the worker's past performance in the tasks with varying amounts of reward. Based on the two proposed context-aware trust metrics, the degree of a worker's trustworthiness in the tasks that belong to different types and possesses different reward amounts can be specifically evaluated. The context-aware trust evaluation provides the basis for making an effective context-aware trustworthy worker selection.

(b) Conventional trust evaluation models, e.g., [157, 139] commonly apply an aggregated value of a person's scores in multiple trust metrics to indicate a user's trust level among all users. Inevitably, the weights for aggregating the scores may generate subjective bias. To eliminate the bias, we model the trustworthy worker selection problem as a multi-objective combinatorial optimization problem. Then, we propose a multi-objective worker selection algorithm based on NSGA-II [31] to find the worker combination consisting of the most trustworthy workers.

The results of the experiments on the real crowdsourcing scenarios-based synthetic data show that our proposed CrowdTrust can effectively select workers who are more reliable and capable than the workers selected by the overall answer approval rate-based model. Moreover, CrowdTrust can effectively differentiate trustworthy workers from untrustworthy workers when both of them have high overall answer approval rates.

2. The second contribution of the thesis is a trust vector-based spam worker defense

---

model called CrowdDefense that infers the trust relationships between a worker and different types of requesters to judge if the worker is a spam worker.

- (a) We propose a new Crowdsourcing Trust Network (CTN) that consists of workers, requesters, and edges with transaction-based direct trust values. Based on the CTN, we analyze three typical threat patterns with which spam workers collude with their accomplices to boost reputations and then mount attacks.
- (b) Based on the analysis of the three threat patterns, we devise a new method to infer the trust relation between a worker and a requester who are indirectly connected in a CTN. Then, we propose a novel Worker Trust Vector (WTV) to represent a worker’s global trust level. A worker’s WTV includes the trust scores of the worker from the views of three types of requesters, respectively.
- (c) Based on the calculated WTVs, we further propose a novel worker selection method. With this method, the workers are prevented from participating in tasks if the trust scores in their WTVs are lower than the average scores. This method effectively defends against the threats from spam workers.

The experiments on the semi-synthetic datasets generated from a real dataset *soc-sign-epinions*<sup>1</sup> demonstrate that our proposed CrowdDefense significantly outperforms three state-of-the-art approaches in preventing spam workers from participating in the tasks published by honest requesters. Moreover, CrowdDefense can filter out the spam workers who may not be recognized by CrowdTrust.

3. The third contribution of the thesis is a trust network-based spam identification model that learns the worker samples with known identities to predict an unknown worker’s identity.

---

<sup>1</sup><https://snap.stanford.edu/data/soc-sign-epinions.html>

- 
- (a) We propose a requester taxonomy and a worker taxonomy according to the transaction behaviours of the requesters and the workers in a Crowdsourcing Trust Network (CTN).
  - (b) We propose a new trust metric called trust trace. A trust trace measures the extent to which a worker is trusted by a requester in a fixed-hop sub-CTN starting from the requester. We then devise a novel worker trust representation called Worker Trust Matrix (WTM). A worker's WTM contains the trust traces between the worker and all the requesters and thus is a global trust network-based feature set of the worker.
  - (c) We prove that a WTM cannot be manipulated by any worker and contains the usable information for identifying a worker's identity. Both the unmanipulable property and the usable property of a WTM are critical for effective spam worker identification.
  - (d) We propose a novel learning algorithm to predict a worker's identity.  
The results of extensive experiments over one real dataset and three semi-synthetic datasets demonstrate the superior effectiveness of our proposed model in identifying spam workers.
4. The fourth contribution of the thesis is a trust-aware worker recommendation model called CrowdRec that recommends capable workers to a requester according to the requester's transaction preferences.
- (a) We propose two types of metrics to evaluate the similarity between two requesters who have commonly transacted workers. The values of the two metrics are calculated based on the two requesters' transactions with their commonly trusted workers and their commonly distrusted workers, respectively. Modelling the similarity between two requesters from both the trust view and the distrust view can help improve the accuracy in aggregating different requesters' opinions in predicting a worker's performance.

- 
- (b) We propose a novel trust metric called Strength of Trust (SOT) to measure the degree of trust between two requesters who have not transacted with any common workers. Targeting the sparse data, we propose a Trust Sub-Network Extraction algorithm (TSE) to discover more trustworthy requesters for gathering opinions for generating recommendations.
  - (c) We propose a strategy for recommending workers for a task published by a newly registered requester and a strategy for letting newly registered workers obtain the opportunities to be recommended. By incorporating the similarity metrics, the new trust metric, the new trust sub-network extraction algorithm and the new strategies, we propose a novel trust-aware worker recommendation model called CrowdRec. To the best of our knowledge, in the literature, this is the first requester-oriented recommendation model in crowdsourcing environments.

The experiments conducted on a simulated crowdsourcing platform demonstrate that our proposed CrowdRec significantly outperforms the classic collaborative filtering recommendation model and three state-of-art trust-based recommendation models in terms of both accuracy and coverage.

## 1.4 Roadmap of the Thesis

The thesis is structured as follows:

Chapter 2 starts with a comprehensive literature review on crowdsourcing and trust and then presents the current studies targeting our proposed three challenging problems of selecting trustworthy workers in crowdsourcing environments.

Chapter 3 presents a trustworthy worker selection model called CrowdTrust based on our proposed context-aware trust metrics and a genetic algorithm-based worker selection method. This chapter includes our paper published at IEEE ICWS 2015 [149] (refer to the publication list on Pages ix and x).

Chapter 4 presents a trust vector-based spam worker defense model called Crowd-

Defense that evaluates a worker's trustworthiness from different requesters' views to help distinguish spam workers from honest workers. This chapter includes our paper published at IEEE ICWS 2017 [150].

Chapter 5 proposes a novel spam worker identification model by incorporating trust network-based trust metrics and machine learning techniques to effectively identify spam workers. This chapter includes our completed paper to be submitted.

Chapter 6 introduces CrowdRec, a trust-aware worker recommendation model that discovers trustworthy workers by taking the homogeneous workers, dishonest behaviours, data sparsity, and cold-start phenomenon problems into account. This chapter includes our paper published at IEEE ICWS 2016 [148].

Finally, Chapter 7 concludes the work in this thesis and presents opening trust challenges in crowdsourcing environments, some of which have been discussed in our paper accepted by IEEE Transactions on Services Computing (TSC) in 2018.

# Chapter 2

---

## Literature Review

---

In recent years, the success of crowdsourcing has attracted increasing attention from the business world, non-profit organizations, and academia [162]. Different from traditional problem-solving models, crowdsourcing possesses four desirable characteristics, i.e., the high scalability, the low or even free transaction fee, the high-degree anonymity, and the voting-based consensus mechanisms, respectively. These characteristics have empowered crowdsourcing to be much more economical and effective than the traditional problem-solving models in solving various human intelligence tasks. Unfortunately, the untrustworthy workers can also leverage the four characteristics to cheat or sabotage the effectiveness of crowdsourcing. However, the research on selecting trustworthy workers is still at the embryonic stage without enough attention paid to investigate the *reliable*, *truthful*, and *capable* properties of trustworthy workers.

In order to present the trust management problem in crowdsourcing and the efforts have been made for solving the problem, in this chapter, we first provide a systematic review of the background knowledge of crowdsourcing followed by a discussion of the trust in different domains. After that, a comprehensive literature review of studies focusing on three challenges of the trust management problem is presented. In particular, this chapter is organized as follows:

- Section 2.1 introduces a new categorisation of crowdsourcing models followed by the typical characteristics of various crowdsourcing models.
- Section 2.2 introduces the definition of trust from different perspectives, followed by a new trust definition of a crowdsourcing worker.

- Section 2.3 systematically reviews the existing studies proposed for trust management in crowdsourcing from three aspects: *context-aware trust evaluation*, *spam worker defense*, and *trustworthy worker recommendation*, respectively.
- Section 2.4 presents a summarize on the existing studies proposed for enabling trustworthy crowdsourcing environments.

## 2.1 Background Knowledge of Crowdsourcing

The core idea of crowdsourcing is to scalably utilize the human intelligence of the part-time workers to cost-effectively solve tasks rather than to hire full-time professionals to address the same tasks costly [38]. Based on it, a wide variety of crowdsourcing models have been proposed to solve various practical problems, such as jigsaw puzzle challenges [126], making plans [128, 91], retrieving information [80], detecting objects [127], and labelling data [98]. By investigating the differences among the workflows applied in different crowdsourcing models, we below present a new crowdsourcing model categorisation to exhibit the existing crowdsourcing models. Then, we conclude the typical characteristics of the existing crowdsourcing models.

### 2.1.1 A New Categorisation of Crowdsourcing

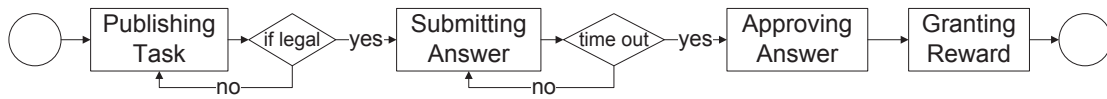
As we defined in Chapter 1, a crowdsourcing process consists of four necessary procedures: (1) publishing task, (2) submitting answer, (3) approving answer, and (4) granting reward. Based on the four necessary steps, various crowdsourcing models have been designed and been applied to build crowdsourcing systems. Typically, the existing crowdsourcing models can be classified into *discrete crowdsourcing* and *continuous crowdsourcing*, respectively, according to the life circles of the crowdsourcing processes in different crowdsourcing models.

### 2.1.1.1 Discrete Crowdsourcing

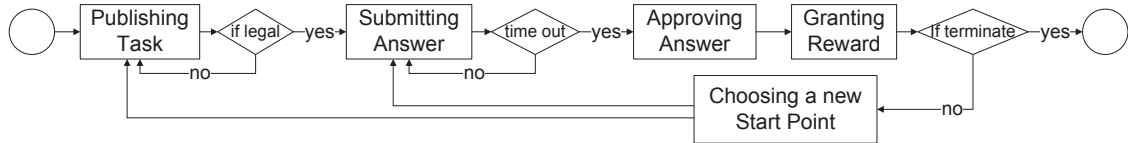
In a discrete crowdsourcing model, a crowdsourcing process is completed after the four necessary crowdsourcing procedures are all executed in sequence, as depicted in Fig. 2.1. In particular, a requester packs the tasks belonging to the same task type into a task group beforehand and then publishes the packed tasks online for workers in the form of an open call. In such a task group, all the tasks commonly require a worker to take the similar operations, e.g., tagging images. The only difference of the tasks in a task group is that their targeting instances are different. For example, in an image tagging task group, a worker is asked to tag different images assigned in different tasks. Moreover, in a discrete crowdsourcing model, all the tasks in a task group are independent to each other, which means that the crowdsourcing outcome of a task in a task group does not make any influence on the crowdsourcing outcomes of other tasks in the task group.

A typical application of a discrete crowdsourcing model is the ESP game where pairs of workers are asked to guess the labels of an image in each task [134]. In the ESP game, if two workers type the same label for an image, the label will be adopted and the image is marked as labelled, indicating a crowdsourcing process is completed. Another example is a group of missing boat seeking tasks published at Amazon's Mechanical Turk [38]. In this example, each task associated with a satellite picture requires a fixed number of workers to report if there is a boat in the picture. A task starts from the point that a requester publishes the task and ends at the point that the crowdsourcing process determines the final approval answer and accordingly grants rewards to the winning workers. Besides the two typical examples, many other crowdsourcing systems apply the discrete crowdsourcing model to solve various problems. For instance, Threadless.com applies a discrete crowdsourcing model to organize workers to design a T-shirt. At Threadless.com, workers with valid email addresses can directly submit their T-shirt designs online, and then the design with the highest scoring (i.e., the answer approval mechanism) is selected and finally rewarded. Such a T-shirt de-





**Figure 2.1:** A Discrete Crowdsourcing Process



**Figure 2.2:** A Discrete Crowdsourcing Process

sign task is finished after executing the four necessary steps in sequence. Moreover, in many other well-known crowdsourcing platforms, e.g., CrowdFlower, FreeLancer, Witmart, and InnoCentive, the discrete crowdsourcing model is also commonly applied.

**2.1.1.2 Continuous Crowdsourcing**

Different from a discrete crowdsourcing model, in a continuous crowdsourcing model, the four necessary crowdsourcing procedures of a crowdsourcing process may be repeated several times to continuously solve a series of correlative tasks, as depicted in Fig. 2.2. In general, a continuous crowdsourcing model is applied when the answer of a task can be continuously improved, or a task is too complex to be solved by a discrete crowdsourcing process and thus needs to be decomposed into relatively easy subtasks. Similar to the start point of a crowdsourcing process in a discrete crowdsourcing model, in a continuous crowdsourcing model, a requester first packs tasks belonging to the same task type into a task group, and then publishes them online for workers in the form of an open call. Differently, in a continuous crowdsourcing model, the outputs of a task, i.e., the answers, are used as the input of the next task which is automatically published the crowdsourcing system or manually published by the requester. For example, in a survey writing task, a continuous crowdsourcing model is commonly applied. In particular, the first subtask is to organize workers to write a sur-

---

vey based on the references collected from the last subtask. Note that, in a continuous crowdsourcing model, the tasks are correlative, which means that the quality of the crowdsourcing outcomes of previous subtasks will profoundly influence the quality of the crowdsourcing results of the following tasks.

A well-known application of a continuous crowdsourcing model is the editing on an entry in Wikipedia. In a wiki entry edition task, creating an entry is the first type of subtasks while updating the content under an entry is another type of subtasks that are explicitly correlative to the entry creation. In particular, once an entry is created, the entry allows different workers from the people crowd to continuously updated the information for refining the entry. Each time of the refinement implicitly republishes a crowdsourcing task, in which a new discrete crowdsourcing process is launched. Likewise, in the development of many famous open-source softwares, such as Linux and Hadoop, the continuous crowdsourcing model is applied, which enables programmers around the world to continuously improve the systems.

In academia, continuous crowdsourcing models have also been widely discussed. A bounding box annotations system is designed and demonstrated to be effective in marking target object in images [127]. In a bounding box annotations task, three types of subtasks including drawing bound, quality verification, and coverage verification are solved one-by-one until all the target objects in a given image have been exactly bounded. In addition, Turkomatic authorizes the workers to help requesters to break down a task into a series of subtasks (i.e., implicitly correlative tasks). In a task consisting of several subtasks, the approved answer of a subtask is the basis of solving the next subtask [79]. Likewise, a novel crowd-based route recommendation system crowdPlanner decomposes a route plan task into two correlative type of tasks: computers produce initial solutions and workers submit solutions, respectively [128].

## 2.1.2 Work Patterns of Crowdsourcing

Targeting different types of crowdsourcing tasks, different work patterns have been applied to organize the participating workers to work effectively. In general, the work patterns can be classified into four types: *competition based worker pattern*, *contribution based worker pattern*, *collaboration based worker pattern*, and *hybrid worker pattern*.

### 2.1.2.1 Competition-based Work Pattern

Competition is an intuitive approach to promoting the participating workers to make more efforts in a published task so that a better solution for the task can be finally obtained. In particular, in a crowdsourcing task, the competition based work pattern first requires all the participating workers to submit their own answers in the task by the deadline. Once the deadline approaches, all the submitted answers are available to voters for determining the final rankings of the answers. In a competition based work pattern, only the workers whose answers rank in the top- $k$  places will be rewarded. In fact, traditional contests can be regarded as the original form of crowdsourcing process that utilizes competition based work pattern to organize workers. Differently, in crowdsourcing environments, competitions are more frequently launched for solving a variety of tasks.

For example, Threadless is a typical practical case that uses competition based work pattern to organize workers for completing tasks. In Threadless, global workers join to T-shirt design tasks, in which the highest scoring designs are selected and rewarded. However, one difference between a task in Threadless and a traditional contest is that both the amateurs and the professionals are all welcomed to participate in the task in Threadless. Another example is InnoCentive that utilizes competition based work pattern to organize scientists to solve R&D challenges post by companies with financial rewards. Moreover, Witmart provides a foundational platform for tackling various tasks, including design, programming, text editing and other simple repetitive

tasks, most of the tasks in Witmart adopt the competition based work pattern.

### **2.1.2.2 Contribution-based Work Pattern**

Contribution based work pattern is another vital work pattern, which can be described as: workers submit their works to a specified functional site where their submissions are automatically converted into valuable goods. In this work pattern, the intelligence works submitted by workers are labelled and then displayed to all the internet users as exchangeable goods. When a user applies for obtaining a submission from the site, a monetary or non-monetary reward needs to be paid by the user for both the original owner of the submission and the site operator. For example, in iStockphoto, Flickr, and CPAN, the contribution-based work pattern is applied to organize workers for collecting their intelligence works.

Note that, in some crowdsourcing systems, workers unconsciously contribute their works to the crowdsourcing systems. For example, in the ESP game, workers believe that they are playing games while while the Google Image Labeler automatically collects the tags for the published images for further leverage. In addition, the social network sites, e.g., LinkedIn, MySpace, and Facebook, and the e-commerce sites, e.g., eBay and Taobao, commonly apply the contribution based work pattern to organize their own users to contribute their data implicitly. From the view of an individual user, it only regards the sites as the service providers, while from the perspective of the site owners, the users are organized to contribute to building various datasets. In this kind of crowdsourcing system, the data of users' operations, habits, or even interaction based network can be recomposed as the basis for solving other crowdsourcing tasks, e.g., advertisement, investigation and science research.

### **2.1.2.3 Collaboration-based Work Pattern**

Collaboration based work pattern is devised and increasingly applied in crowdsourcing tasks to solve the cumbersome tasks that cannot be easily solved by either a computer

---

or a single person. In a collaboration based work pattern, workers can study based on the results of those former works have been solved to reduce the cost of solving a complex task. In particular, in a collaboration based work pattern, a crowdsourcing worker is required to complete the subtasks of a complex task in its field. By combining all workers' efforts in all the subtasks, an effective solution for the original complex task is finally obtained. Essentially, the key to collaboration based work pattern is to decompose a complex task into small and easy subtasks that can be solved by ordinary workers. Workers submit their answers to subtasks according to their specialized knowledge. Finally, an aggregated solution for the original task is formed by merging every valid worker's answers.

For example, in most of the continuous crowdsourcing models, like Wikipedia, Linux, and Hadoop, the collaboration based work pattern is commonly applied to continuously improve the answer of a task. Another representative example is Google Fusion Tables, which is an online platform for users to share, edit or recompose data into visualizations.

#### **2.1.2.4 Hybrid Work Pattern**

Recently, the hybrid work pattern that combines any number of the three above work patterns is proposed for well organizing the computer capacity and the human intelligence to solve more complex tasks, such as decision-making tasks. The computers are commonly good at the quantitative evaluation while the humans perform better in dealing with the qualitative problems. For example, in the CDAS [90], a quality sensitive model is applied in the computer system to automatically evaluate the quality of workers' answers. Another example is the CrowdPlanner [128] that is a novel crowd-based route recommendation system. In CrowdPlanner, the computers are in charge of two essential procedures: *producing initial solutions* and *evaluating answers*, while the humans provide their suggestion for completing the final recommended plan.

### 2.1.3 Characteristics of Crowdsourcing

- **The huge number of undefined workers:** In crowdsourcing environments, a requester who publishes a task needs to face more than ten thousands of undefined workers [32]. As a crowdsourcing task is published in the form of an open call, all the undefined workers have the equal chance to participate in the task following the basis of the first-come-first-serve. Differently, in traditional problem-solving models, a requester directly assigns tasks to his/her known full-time employees rather than publishing tasks online for collecting candidate solutions from the undefined crowdsourcing workers.
- **The low or even free transaction fee:** In crowdsourcing environments, the transaction fee is extremely low or even free [17, 21, 98, 22]. For example, at Amazon Mechanical Turk, a task commonly provides a micro-reward of around \$0.1, and only 10% of the reward is charged by the platform. Thus the cost of a transaction is only about \$0.01 with a minimum of \$0.005 per task [98, 136]. The commissions charged by other crowdsourcing platforms for a successful task are commonly low, e.g., In addition, regarding the tasks of editing the entries in Wikipedia and the tasks of improving the open-source projects released in Github, the workers provide their contributions to the tasks without charging for any monetary reward.
- **The high-degree anonymity:** In crowdsourcing environments, the high-degree anonymity of workers are guaranteed by most of the crowdsourcing platforms for attracting more diverse workers to contribute their human intelligence to solve the published tasks [155, 146]. In crowdsourcing platforms, e.g., Amazon Mechanical Turk, workers are required to provide their personal information for registration. However, from the view of a requester, a registered worker in a crowdsourcing platform is still anonymous because the rarely available information about the worker cannot obviously reflect if the worker is reliable, truthful, and capable.

- **The voting-based consensus:** In fact, the essence of a crowdsourcing process is to seek for an answer that is close to the unknown ground-truth for a published task. Thus, voting-based consensus mechanisms, such as the half-voting mechanism and the majority-voting mechanism, are widely applied to determine if an answer should be approved as the ground truth [38]. In particular, in a task, the half-voting mechanism approves an answer if no less than half of the voting workers agree with it. In addition, the majority-voting mechanism approves an answer if it receives the most votes from the workers. The most important benefit of the voting-based consensus mechanisms is that the cost of hiring a professional to verify an answer is saved.

## 2.2 Trust and Trust Influence in Crowdsourcing

In the past decades, trust has been widely discussed across disciplines, such as psychology, sociology, economics and computer science. In general, trust is explained as one type of relationships between a trustor and a trustee, which indicates that the trustor believes the trustee is dependable in a specified period within a specified context [8]. It is valuable to model trust as it can help the interaction between two entities move toward to a better outcome than the expected one. However, modelling trust is a very complicated issue because various factors may jointly influence the trust between two entities and these factors may vary with the applying environments changing. In particular, in different disciplines, trust can be referred to honesty, truthfulness, competence or reliability of a trustee. Moreover, in the interactions among humans, trust can be influenced by personal preference, the suggestions of friends, the psychological factors, and the expected benefits from building the trust [76, 113].

In order to well explain the trust in crowdsourcing environments, in this section, the main components of trust including the meanings of trust and the properties of trust are introduced first. Then, a discussion of the impacts of trust on crowdsourcing activities is presented.

## 2.2.1 Meanings of Trust

As the meanings of trust vary in different disciplines, it is surprisingly tricky to give out a restricted definition of trust. Thus, based on the studies on trust in the literature, we list out the most commonly applied trust definitions in the primary disciplines, such as psychology, sociology, economics and computer science, respectively.

### 2.2.1.1 Trust in the Primary Disciplines

- **Trust in Psychology.** In Psychology, a widely used trust definition is proposed by Deutsch [35]. He defines trust as a psychological state of a trustor (an individual). Moreover, Deutsch explains that trusting behaviour occurs when a trustor encounters a situation where it needs to balance the possible cost and benefits for making a judgement on the risk of taking actions of a given commitment. Once the trustor takes actions based on his/her judgement, the outcomes of the interaction between the trustor and a trustee will only be determined by the actions taken by the trustee. In such an interaction, compared to the gain of a good result, an unfortunate result commonly brings more loss. Likewise, Kramer [77] points out that an individual places itself at risk when it chooses to trust another individual. Furthermore, by investigating the nature of interpersonal trust in psychology, Rotter [118] states that trust is an expectancy held by an entity (a person or a group of people) that the word, promise, or verbal or written statement of another entity can be relied on. From another perspective, trust in psychology can be interpreted from three aspects: cognitive, emotive, and behavioral [11]. Regarding the cognitive aspect of trust, a trustor commonly makes a rational evaluation on a trustee by taking impartial observations into account. By contrast, the emotive aspect of trust is reflected on the influences of a trustor's nonrational or emotional preferences on his/her trust on a trustee. Moreover, the behavioural aspect of trust refers to the final actions taken by the trustor after he/she well knows the potential risk, which can also be conceptual-



---

ized as the confidence degree held by the trustor on the reliability and integrity of the trustee [9].

- **Trust in Sociology.** In the field of sociology, Sztompka [129] gives out a brief and concise trust definition, i.e., trust is “a bet about the future contingent actions of others”. In fact, trust in sociology is widely investigated from the perspectives of personal trust (individual level) and the perspective of institutional trust (societal level) [60], respectively. At the individual level, the definition of trust in sociology is similar to those proposed in psychology [35, 118] because they commonly indicate a trustor’s judgement on the future outcomes that will be led by the actions of a commitment. For example, Golbeck et al. claim that “trust in a person is a commitment to an action based on a belief that the future action will lead to a good outcome”. By contrast, at the societal level, trust is generally agreed as one of the properties of a specific social community which is measured by a collective psychological state of the members in the community [124]. Moreover, from the perspective of explaining the function of trust in society, Luhmann [92] claims that trust is “a means for reducing the complexity of society” since social members follow the rules to perform predictable behaviours. Furthermore, Seligman [123] points out that “if people play their roles according to role expectations, we can safely conduct our own transaction accordingly”. Essentially, the societal-level trust is formed based on the general expectation of a trustor on the members from a type of social community and can be further extended to imply the trustworthiness of strangers belonging to this type of social community.
- **Trust in Economics.** Economists generally suggest trust as a form of “implicit contracting” [83] between two individuals or firms who will take actions following their made promises. For example, according to the definition provided by the European Commission Joint Research Centre [67], “trust is the property of a business relationship, such that reliance can be placed on the business

---

partners and the business transactions developed with them”. In addition, trust is explained as a self-enforcing agreement implicitly admitted by two parties who then take actions in future transactions according to the agreement [13]. By contrast, some economists claim trust is an “externality” that can not be directly traded but contains real economic value (i.e., increased efficiency in transactions) [165]. Moreover, regarding the negative impacts caused by information asymmetry in economic activities, trust is considered by some economists as a useful mechanism that helps suppress opportunistic behaviours [4, 10].

- **Trust in Computer Science.** In the field of computer science, a well-known definition of trust is given by Mui et al. [103], i.e., trust is “a subjective expectation an agent has about another’s future behavior based on the history of their encounters”. Similarly, Jøsang et al. [71] state that “Trust is the subjective probability by which an individual expects that another performs a given action on which its welfare depends”. In fact, most of the trust definitions proposed in the formal three disciplines, i.e., Psychology, Sociology, and Economics, can be extended to computer science because the activities happened in computer-based environments are actually extended from those human activities happened in traditional environments. In particular, personal psychology variations, social activities, and economic transactions are currently connected with computers and deliver the new attributes in the form of informatisation at every moment. Differently, scholars from computer science focus more on the quantitatively formulating the trust of either an individual or a community [95, 16]. Thus, in computer science, trust is accordingly formulated by policy-based mechanisms [107, 144, 106], reputation-based mechanisms [18, 72, 163], and prediction-based mechanisms [49, 51, 46, 59], respectively. In particular, policy-based mechanisms let participating parties build trust via exchanging credentials, such as digital signatures, resumes, and text-chats, following preset transaction policies [8]. By contrast, without the exchanging credentials, reputation mechanisms commonly aggre-

---

gate the information derived from the observations on an entity's past behaviours to calculate a trust score or a trust rank. In general, a trust score or a trust rank is further used to indicate the degree to which the entity can be relied on by others [1]. Moreover, the prediction-based mechanisms commonly predict the trust relationship between two individuals/parties via training a probability model [116] or a learning model [59] by utilizing the information correlating to trust. In general, the information includes the personal characteristics (e.g., preference and domain expertise) and the mutual relations (e.g., existing trust relationship, social intimacy and interaction context).

## **2.2.2 Properties of Trust**

In order to thoroughly explain the activities happened in various environments, scientists from different disciplines correspondingly define trust in many different ways. To extract the common properties of trust based on such a significant amount of trust definitions is extremely significant for effectively formalizing trust. As such, scientists have conducted long-term observations and experimental verifications on human activities for investigating the properties of trust. In this subsection, these trust properties are introduced in detail.

### **2.2.2.1 Context Dependency of Trust**

In past decades, many studies have pointed out that trust is highly context-aware [100, 3, 2]. According to the definition given by Oxford Dictionary, contexts are “The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood.”. In other words, all the factors that influence an action or an event happens can be regarded as contextual factors. Dey et al. state that “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves”,

---

which has been widely adopted in the field of computer science [36]. Taking the trust relationship between two participants in a recommender system as an example, a person may trust a recommended person to different extents as the recommender's expertise varies with the domain of the recommendation changing [141]. In addition, in the definition of personal trust proposed by McKnight et al. [100], the influence of context is further emphasized, namely, "one person trusts another specific person, persons, or thing(s) in the specific situation". A vivid example is then given out by Marsh [95], i.e., "Whilst I may trust my brother to drive me to the airport, I most certainly would not trust him to fly the plane". The view conveyed by the example is that the trust evaluation on the same person should be differentiated in different contexts. Refer to the specific studies in computer science literature, considering the context of publications, the social relationships between scholars (e.g., co-authors, a supervisor, and his/her students) are mined. Similarly, considering the context of the content topic in a scholar's homepage, the role of a scholar (e.g., a professor in the field of data mining) can be mined [130].

More specifically, in crowdsourcing environments, contexts are mostly related to the task itself and accordingly influence the trust relation between a trustor and a trustee. In particular, regarding the influence of the task type, a trustor who trusts a person in a programming task may not trust the person in a T-shirt design task. In addition, regarding the influence of the task reward amount, a trustor who trusts a person in a task with \$1 reward amount may not also trust the person in a task with \$10 reward. Moreover, a worker's geography location may imply its trustworthiness to some extents, and the temporal factors also influence on a worker's trust. In the studies on the location-based crowdsourcing systems, e.g., [6, 101, 48], crowdsourcing workers' behaviours have been demonstrated to be heavily influenced by their geographical positions, and thus the trusting behaviours of them vary with the context changing. In order to help requesters build trust relations with workers, from the view of recommending appropriate tasks to workers, David Geiger et al. state that a worker's preference in choosing a task in crowdsourcing shifts with the time and location

changing [45]. The significant influences of contexts on the trust evaluation in crowdsourcing environments have been emphasized. However, the study on investigating the specific ways that the contexts influence the trust in crowdsourcing environments is still at the embryonic stage. Moreover, it is very challenging to exactly evaluate a crowdsourcing participant's trust in new contexts of crowdsourcing.

#### 2.2.2.2 Transitivity of Trust

As one of the most important properties of trust, the transitivity is commonly leveraged to infer the trust relationship between two unfamiliar or even unknown entities via transmitting the trust messages the intermediaries who indirectly connect the two entities. A well-known example is the “friend of a friend (FOAF)” [47]. In particular, if Clark is trusted by Bob who is trusted by Aaron, Clark will be trusted by Aaron to some extent. This example can be further explained by a recommendation scenario where Bob recommends his friend Clark to be the friend of Aaron [26]. Precisely, the transition of trust is not perfect in the mathematical sense, namely, if Aaron trusts Bob to 100% and Bob trusts Clark to 100%, it does not always mean that Aaron will trust Clark to 100% [70]. In the literature, a wide variety of trust transition methods have been discussed, such as direct propagation (i.e., FOAF), co-citation (i.e., given  $a$  trusts  $c$ , and  $b$  trusts  $c$  and  $d$ ,  $a$  also trusts  $d$  to some extent), transpose trust (i.e., if  $a$  trusts  $b$  then trusting  $b$  imply trusting  $a$ ), and trust coupling (i.e.,  $a$  and  $b$  trust  $c$ , then trusting  $a$  implies trusting  $b$ ) [64, 82]. In general, a person will seek suggestions from his/her trustees or the trustees of his/her trustees for evaluating if an unknown person is trustworthy or not. When the distance between the person and a trustee increases, i.e., the length of transitive path increases, the transitive trust generally decays [70]. Modelling the decay of trust is generally challenging as the decay nonlinearly varies in different trust networks with multiple implicit variants [164, 161]. In crowdsourcing environments, the requesters have no direct transactions with each other, and thus there are no direct connections between any two requesters. Likewise, there are no direct transactions between any two workers. Thus, in this thesis, the transitivity of trust of

---

crowdsourcing participants needs to be reconsidered by taking the new characteristic of the crowdsourcing network into account.

### **2.2.2.3 Personalisation of Trust**

The personalisation of trust is derived from the psychological experiences of the individual who bestows it in social psychology [54, 93]. In other words, the personalisation of trust can be explained as the personal attitude of an individual under different evaluation criteria in different domains. In particular, different people may often hold different opinions on the same person, which is essentially caused by the differences existing in their evaluation systems. For example, given two crowdsourcing workers Aaron and Bob and a crowdsourcing requester Clack, Aaron may trust that Clack will fairly organize voters to determine the final approval answer even if Aaron himself has never succeeded in the tasks published by Clack. By contrast, based on the evaluation system of Bob, he may choose to distrust Cathy once he fails in a task published by Clack. Essentially, the differences derived from their different trust evaluation criteria. Likewise, when the population are asked a question that “do you trust the current President to effectively lead the country?” [46], people holding different opinions will be naturally split. In particular, some of them will choose to trust the president very highly, and the others will only show very little trust in the capacity of the president.

“Trust is personalized” also implies that “Trust is context-aware”. Essentially, different people differently evaluate the trust of a person because they believe that different contexts possess different degrees of importance. For example, when evaluating if a person is trustworthy, some people pay more attention to the educational background and wealth of the person while others may pay more attention to the temperament of the person. Another example is that a PhD student may trust his/her supervisor more than the supervisor trust on him/her, which is called “one-way trust” [41]. Jøsang [68, 69] states that a person’s trust opinion on another one is derived from the person’s subjective logic, i.e., the logic that operates on cognition of the world. The personalisation property of trust has been leveraged to build a trust evaluation model for finding

---

trustworthy vendors in e-commerce [85]. However, there is not a universal measure of the personalized trustworthiness of a person. Compared to the global trust evaluation on a person, personalized trust evaluation is more important in the case to build a trust relationship between two crowdsourcing participants.

### **2.2.3 The Influence of Trust in Crowdsourcing Activities**

In crowdsourcing environments, trust is the foundation for promoting a successful transaction happened between a requester and a worker because each participant has the willingness to ensure their benefits can be guaranteed under a low risk. Thus, the trust directly influences the ways in that a crowdsourcing participant chooses its transacting counterparties. In addition, trust is the important basis for differentiating voters' opinions in the answer approving stage. Moreover, trust also influences the decisions made by both requesters and workers to honestly or dishonestly participate in crowdsourcing activities.

#### **2.2.3.1 The Influence of Trust on Choosing Transaction Counterparty**

From the view of a person, in order to guarantee himself/herself can benefit from the transactions with others, the person may only choose to transact with those people who are evaluated as trustworthy in his/her own evaluation system. This is because a person prefers to accept the suggestions from his/her trustees rather than from those untrustworthy people who are reported by others [12]. In crowdsourcing environments, the demand for transacting with trustworthy workers becomes much more urgent because the untrustworthy workers may provide vague or even harmful answers to prejudice the benefits of a requester. For instance, in the DARPA Network Challenge (a.k.a. the Red Balloon Challenge), the majority of balloon sights submitted by the workers to the winning team from MIT are finally proved to be erroneous. Furthermore, among the workers who submit these erroneous answers, either the unintentional error makers and the intentional attackers are found [105]. In addition, a large number of extra

---

resources are consumed for dealing with the low-quality or even erroneous answers in crowdsourcing tasks [74]. As such, any honest requester has the urgent demand to identify the workers who can be trusted. Regarding the crowdsourcing workers, in fact, they are commonly influenced by the trust in making decisions on choosing whom as their transaction counterparties. A worker prefers to participate in the tasks published by the requesters who are trusted by himself/herself because he/she possesses a strong willingness to be fairly treated in the tasks. Moreover, a worker requires to obtain reasonable payment for his/her provided service. For example, Wang et al. [137] suggest that a reputation-awareness-based auction mechanism can help an SIoT service provider correctly choose a reliable service demander to transact with and thus effectively protects the benefits of the service provider. Essentially, the fairness is the foundation for guaranteeing the fact that a worker itself can always gain the returns correspondingly matching its paid efforts.

### **2.2.3.2 The Influence of Trust on Approving Answers**

In a crowdsourcing process, voting-based consensus mechanisms are commonly applied to determine which answer should be approved. In a voting-based consensus mechanism, the votes from different workers influence the final consensus result to different extents [38]. In other words, in a crowdsourcing task, the significance of two votes from two different voters should be differentiated because they possess different trust levels in the task. In a general practical case, a worker's trust level is calculated and then be used as a weight to indicate the extent to which the worker's opinion should be considered in approving an answer [99]. For example, in most of the spatial crowdsourcing applications [43, 132, 131], the answers of different workers are directly differentiated by taking the workers' trust levels as the reference. Essentially, a weight associating with a worker is an implicit form of votes, which indicates the significance of the answer submitted by the worker. In practice, when facing a large number of workers who all can be trusted, the differentiated trust help a requester effectively differentiate the values of the answers submitted by different workers. As



---

such, using a trust to differentiate workers has become a new challenging problem in crowdsourcing environments.

### **2.2.3.3 The Influence of Trust on Motivating Participants**

Trust evaluation provides an incentive mechanism for encouraging an honest worker to continually and considerably participate in the crowdsourcing processes so as to maximize his/her possible gain. In particular, an effective trust evaluation model enables the workers with sharp ability and responsible working attitude to obtain more opportunities than those low-quality and irresponsible workers to participate in the tasks. As a result, it promotes the workers to believe that following the rules to participate in crowdsourcing processes is the best choice for maximizing their benefits rather than using tricks. In addition, with an effective trust evaluation model, the trustworthy workers can be always permitted to participate in tasks, and thus improve the high-quality answers in all the tasks. This will convince requesters that crowdsourcing is cost-effective so that the existing requesters may choose to keep using crowdsourcing for solving various tasks and more traditional service demanders can be attracted to join the crowdsourcing. Conversely, an ineffective trust evaluation model can dramatically decrease the willingness of both the requesters and the workers to participate in crowdsourcing processes. First, when a worker finds that dishonest workers can easily succeed in tasks, he/she may choose to either imitate the behaviours of dishonest workers or keep away from crowdsourcing. Second, when a requester finds that crowdsourcing is cheap but always comes with the unsatisfied results, he/she may return to use a traditional problem-solving model so as to guarantee the quality. As a result, the crowdsourcing user base will be destructed. Moreover, from the perspective of system trust, Marian MG [96] demonstrates that crowdsourcing participants' trust in the crowdsourcing platform host mediates the interaction between intrinsic motivation and participation intention. Accordingly, he extends knowledge by incorporating system trust as a positive influence in knowledge contribution.

To sum up, trust acts a significant role in influencing the crowdsourcing partic-

---

ipants' behaviours. In particular, trust not only influences the short-term transaction behaviours of the crowdsourcing participants but also determines if a crowdsourcing system can be successful or not. In this thesis, we mainly focus on solving the trust-relating challenges in helping crowdsourcing requesters select reliable, truthful, and capable transaction counterparties. By providing the effective solutions for selecting trustworthy workers, the benefits of all the honest participants can be protected.

## 2.3 Trustworthy Worker Selection in Crowdsourcing

In order to widely collect human intelligence from as many diverse workers as possible, crowdsourcing models have been deployed in many online crowdsourcing platforms so that a requester can easily publish tasks for collecting the answers from the workers all around the world. As such, in a crowdsourcing platform, a requester has to face more than ten thousands of unknown workers, any of whom may show its willingness to participate in the requester's tasks. Moreover, the purposes of a worker can be well-intentioned or malicious, and the abilities of a worker can be satisfied or unsatisfied, leading a high level of uncertainty in the transactions between workers and a requester. A requester always hopes that trustworthy workers with well-intentioned purposes and satisfied abilities can participate in the tasks. This is because requesters commonly believe that a trustworthy worker can submit an accurate answer with a high probability.

Recalling our proposed definition of a trustworthy worker, i.e., a trustworthy worker should be the one who is *responsible*, *truthful*, and *capable*, in this thesis, we actually discuss how to select the workers who satisfy the three aspects of trust. By selecting trustworthy workers to participate in tasks, the effectiveness of crowdsourcing can be maximized, and the benefits of the crowdsourcing participants can be effectively protected. In particular, considering the three aspects of trust, we have to discuss three challenging sub-problems in selecting trustworthy workers. They are (1) *context-aware trust evaluation*, (2) *spam worker defense*, and (3) *trustworthy worker*

---

*recommendation*, respectively. The decomposition of the main issue (i.e., trustworthy worker selection) helps understand the ways to select the *responsible*, *truthful*, and *capable* workers from different perspectives. Thus, in this chapter, we review the existing studies that have been proposed to target the three challenging problems.

### 2.3.1 Trust Evaluation in Crowdsourcing

In traditional online systems, e.g., e-commerce, P2P file sharing systems, and multi-agent systems, trust evaluation has been demonstrated to an effective way for encouraging users to responsibly act in the interactions with the others in the systems [71, 159]. Moreover, Li et al. [84] suggest that a simple trust value may not be able to depict the trust history exactly and may leave misleading information to service customers. Accordingly, they propose a trust vector containing final trust level, service trust trend, and service performance consistency level to indicate a service provider's global trust. Considering the impacts of contexts on a person's trust, Caballero et al. [20] devise a trust model to evaluate a person's trust in multiple contexts. In particular, this trust model provides references to buyers based on three transaction dimensions (i.e., timeless, quality, and cost). In multi-agent systems, Samek et al. [120] further investigate the impacts of contexts on participants' trust. In particular, through identifying possible hierarchical structures of multi-agent systems, they propose a context-aware trust model that is demonstrated to be effective in selecting cooperators.

As we have discussed in Chapter 1, in crowdsourcing environments, an honest worker may also irresponsibly behaves in tasks if he/she believes that more benefits can be obtained by taking irresponsible actions rather than taking responsible actions. To prevent this phenomenon from happening, trust evaluation is a promising method because it actually offers a promise to workers, i.e., the workers who responsibly and conscientiously work can obtain more opportunities to participate in crowdsourcing tasks. From this view, several trust evaluation models have been proposed to adapt to crowdsourcing systems in both industry and academia.

In industry, a worker’s overall answer approval rate is commonly calculated to indicate the probability that the worker will provide a correct answer in the next task [155]. For example, in Amazon Mechanical Turk, a requester commonly sets a minimum requirement on a worker’s overall answer approval rate. In other words, only the workers whose answer approval rates are above the minimum requirement can participate in the tasks published by the requester. In general, a worker’s overall answer approval rate equals to the percentage of the accepted answers out of all the worker’s submitted answers. Likewise, CrowdFlower leverages a worker’s overall accuracy in past transactions to grant a corresponding trust level badge for the worker. In a task with a minimum requirement on a worker’s trust level, only a worker whose trust level is satisfied can participate in the task. Though the overall answer approval rate or the overall accuracy is intuitive for understanding, it neglects the *context dependency* property of trust. Essentially, the overall answer approval rate a global summarize on a worker’s historical performance in all types of tasks in all contexts, which cannot indicate a worker’s trustworthiness in a specific context, e.g., a specific task type or a task with a specific reward amount.

In academia, from the perspective of maximizing the social welfare of a crowdsourcing system, Yu et al. [153] extend three existing trust evaluation models, i.e., beta reputation model [63], interaction-based knowledge degree model [104], and sequential performance-based trust model [56], to adapt to crowdsourcing environments. However, they do not discuss the influences of the changing contexts on a worker’s trust level. Furthermore, Yu et al. [154] propose a Social Welfare Optimizing Reputation-aware Decision-making (SWORD) approach to striking a balance between the overloaded workload of trust workers and the quality of the submitted answers. In addition, Matteo Venanzi et al. [132] model the calculation of a worker’s trust based on the estimated accuracy  $\theta_{i,j}$  of worker  $i$  in reporting an estimate of a target  $j$ , e.g., reporting a GPS location estimate of a target. In particular, they propose a likelihood model to compute a worker’s trust levels in reporting different estimates in crowdsourcing environments, which are further applied to fuse the untrustworthy information existing

among the reported information for locating cell towers. In this model, a worker's trust levels are heavily related to the estimated accuracy  $\theta_{i,j}$  for the worker, which may be referred to the worker  $i$ 's confidence level about its reported value, the precision of the measuring tool, or the variance of some repeated measurements. However, they do not provide any method to quantify the value of  $\theta_{i,j}$ . In addition, in a real-time ubiquitous crowdsourcing platform [97], the mobility patterns and historical performance are combined for automatically assessing a worker's reputation. Based on the calculated reputations, each worker's final confidence is calculated and further applied as the weight of its answer in those tasks where the initially submitted answers are directly aggregated to obtain a final answer. Moreover, Allahbakhsh et al. [5] take the task value, the credibility of evaluators, and the temporal factor into account for designing the trust metrics for evaluating both the workers and evaluators. In particular, they design two trust metrics, i.e., Local Pairwise Trust (LPT) and Reputation Management Systems (RMS). The LPT is conceptualized based on the trust relationship between every pair of directly transacted users (i.e., a requester and a worker) in each time interval. The RMS is conceptualized based on the credibility of evaluations provided by the majority of evaluators on the contributions of a worker. Furthermore, in a participatory sensing crowdsourcing platform, Davami et al. [28] model the workers' trustworthiness by using four different strategies: maximum likelihood estimation, Bayesian, beta reputation, and Gompertz functions. From the perspective of testing-based verification, a worker's performance in gold-standard tasks (tasks with known answers) is also used to evaluate the worker's trustworthiness in a crowdsourcing data analytics system CDAS [90]. By contrast, from the perspective of using a worker's consistency in past transactions to indicate the worker's reliability, the answers of each two workers in tasks are compared for judging if a worker is trustworthy [136]. The consistency is further discussed by Jagabathula et al. [64], who calculate the disagreement level of a worker's answers in past tasks as the penalty for the worker's trustworthiness.

Though context-aware trust evaluation has been proved to be effective in traditional online systems [120, 158], few systematical works targeting crowdsourcing environ-

ments have been reported in the literature. In the few existing studies in crowdsourcing environments, the spatial and temporal contexts have been demonstrated to be important in modelling a crowdsourcing worker's trustworthiness [152, 29]. However, both the influence of task type or the influence of task reward amount on a worker's trustworthiness are neglected. In a multi-dimensional trust model proposed by Liu et al. [89], a worker has been demonstrated to possess different trust levels in different domains. Therefore, it is significant to build a comprehensive context-aware trust model that can differentiate a worker's trustworthiness by taking the new crowdsourcing contexts into account, e.g., the task type and the task reward amount.

### 2.3.2 Spam Worker Defense in Crowdsourcing

In most of the crowdsourcing platforms, the voting-based consensus mechanisms, such as the half voting-based consensus mechanism, the majority voting-based consensus mechanism, and the probability-based consensus mechanism, are commonly applied to find the approximately correct answer for a published task [38]. Essentially, the correctness of a voting-based consensus mechanism in approving an answer in a task is guaranteed by a condition that the number of honest workers (i.e., workers who tell the truth) is larger than the number of spam workers (i.e., workers who lie) in the task. In other words, to guarantee the finally approved answer is most likely to be correct, the consensus achieved by workers on an answer of a task should reflect the opinions of honest workers rather than the opinions of spam workers. In order to satisfy such a condition, various defense models have been proposed to prevent spam workers from participating in tasks. In particular, the existing models can be classified into two types: *verification-based spam worker defense model* and *trust-aware spam worker defense model*.

### 2.3.2.1 Verification-based Spam Worker Defense Model

In general, a verification-based spam worker defense model judges if a worker is a spam worker according to the assessment results of executing automatic or manual verification criteria in a crowdsourcing process. For example, setting testing questions, making robust verification mechanisms and organizing reviewers to re-check the submitted answers are commonly applied to verify if a worker is a spam worker.

In particular, Kittur et al. [74] point out that publishing tasks with preset testing tasks (i.e., the tasks with known answers) together can effectively differentiate the quality of different workers in a task. This is because the testing tasks possess the pre-known answers so that the corresponding verification on a worker's trustworthiness is feasible and efficient. However, a common drawback of testing tasks-based verification models is that it is extremely hard to determine the number of testing tasks and the difficulty of testing tasks in practice. An overly large number of testing tasks or overly tricky testing tasks will make both the workers and the requesters feel overloaded because both of them need to pay more resources for completing a task. However, setting a small number of testing tasks and easy testing tasks may be ineffective. Differently, from the perspective of consistency-based trust verification, Chen et al. [24] design a verification mechanism that regards workers who always give consistent answers as trustworthy workers. Correspondingly, the workers who commonly submit contradictory answers are marked as suspiciously spam workers. However, one glaring limitation of this work is that it is only effective in dealing with binary-choice problems that the answers can be easily compared in pairs. Doan et al. [38] suggest applying trust management schemes, e.g., blocking suspicious workers and manual monitoring workers' behaviours, to prevent workers with suspiciously untrustworthy behaviours from participating in tasks. However, in this work, no particular criteria have been proposed for defending against spam workers. Hirth et al. [55] propose two mechanisms (MD and CG) to detect the cheating behaviours of workers when applying both verification questions and manual re-checking is ineffective or costly. They assume

that all workers have the same probability to correctly evaluate an answer. However, the assumption is too restrictive to be supported in real crowdsourcing platforms.

To sum up, a well-design verification-based defense model can increase the robustness of a crowdsourcing process by filtering out the answers of suspicious spam workers. However, such a defense model may either easily be limited to a specific type of crowdsourcing tasks and consume a large number of extra resources to verify the quality of all the workers who participate in a task [38]. In addition, the verification-based spam worker defense models evaluate the trustworthiness of a worker during a crowdsourcing process rather than at the beginning of a crowdsourcing process. Thus, such a type of models cannot prevent spam workers from participating in tasks beforehand.

### 2.3.2.2 Trust-aware Spam Worker Defense Model

A trust-aware spam worker defense model prevents a worker from participating in tasks if the trust assessment on the worker exhibits some evidence for indicating the worker is a spam worker [136, 64]. In general, the trust evaluation-based spam worker defense models can be further classified into two categories: *reputation-based spam worker defense model* and *trust network-based spam worker defense model*.

- **Reputation-based spam worker defense model.** A reputation-based spam worker defense model regards a worker with a low trust level as a spam worker, and then prohibits such a spam worker to participate in any task. Based on this idea, nearly all those reputation-based trust evaluation models reviewed in Section 2.3.1 can be leveraged to defend spam workers. For example, Olusegun et al. [42] first calculate the reputations of workers in crowdsourcing environments. Based on the reputations, they then devise a fuzzy trust-based worker access model to defend against the low-quality answers submitted by untrustworthy workers. In addition, some studies have also been proposed to directly defend against spam workers by inferring a worker's reputation in a continuous



---

crowdsourcing process. For example, Karger et al. [73] propose a method to iteratively learn the crowdsourcing outcomes for assigning tasks to the appropriate workers (i.e., the workers with a high probability to submit correct answers) based on a known probability that a worker behaves as a spammer. However, this study is only can be applied in a specific scenario where a fixed group of workers continuously participate in a crowdsourcing process, which has a distinct limitation that a worker is allowed to join and leave freely under the form of an open call. Whitehill et al. [143] use some observed data and Gaussian priors to infer the expertise of each worker who is asked to label a data item. Ipeirotis et al. [62] follow the EM method [30] to calculate each worker's confusion matrix and then quantify a worker's quality based on the probability that it will classify one object of class  $i$  into other classes. Raykar et al. [115] propose an empirical Bayesian algorithm based method SpEM to estimate parameters for separating good annotators (honest workers) from spam workers. Rzeszotarski et al. [119] investigate each worker's behaviour data (e.g., logging activities and mouse movements) to find features for detecting spam workers. To sum up, most of the reputation-based spam worker defense models commonly focus on investigating the "reliability" of a worker from its past transactions that are implicitly assumed to be true. Namely, the common intuition of these models is that a worker's historical transaction records can truly indicate the worker's trustworthiness in future tasks. However, they neglect that a worker may collude in past transactions to obtain fake but "good" transaction records. As a result, these models that neglect to consider the "truthfulness" of workers' transaction records are vulnerable to the spam workers who obtain "good" reputations via colluding with their partners. For example, in Amazon Mechanical Turk, a spam worker can participate in the shadow tasks published by his/her accomplices to boost its overall answer approval rate, so that obtain a good reputation, i.e., a high overall answer approval rate at Turk. More importantly, the low or even free transaction fee [22], (e.g., in general \$0.01 with a minimum of \$0.005 per

task at Turk) and the availability of high-degree anonymity [155, 146] in crowdsourcing environments could allow this type of collusion to be easily conducted by spam workers with a low cost for masquerading themselves as “honest” workers [5].

Recall our definition of a trustworthy crowdsourcing worker, a trustworthy worker should contain three properties, i.e., reliability, truthfulness, and capacity. Reputation-based spam worker defense models commonly neglect the significant influence of the truthfulness on a worker’s trust and thus are vulnerable to a worker who colludes with partners to counterfeit a good “reputation” that can make itself more competitive than other normal workers.

- **Trust network-based spam worker defense model.** Reputation-based spam worker defense models are effective in defending against the threats from lone-wolf spam workers. However, colluding spam workers may bypass the reputation-based spam worker defense models by counterfeiting the favourable information used for calculating their trust levels, and then mount attacks [62]. Targeting these spam workers, reputation-based spam worker defense models focusing on “reliability” are not effective. Thus, we need to pay more attention to leverage the aspect of “truthfulness” of a worker to judge if a worker is a spam worker. Accordingly, we below review some trust network-based spammer defense models that have shown surprising effectiveness in traditional online systems by attempting to discover if a worker counterfeits fake records. For example, SybilLimit [151] admits that a spam worker can obtain fake trust edges to connect itself with the honest region. Based on it, SybilLimit limits the number of permitted spammers to  $g * w$  in the user network, where  $g$  is the number of attack edges (i.e., the edges that directly connects honest users and spam users) and  $w$  is the mixing time of the network. SybilInfer [27] separates the honest users from spammers by directly estimating the minimum-quotient cut between honest networks and a spam network. SybilRank [23] infers an unknown user-

---

s reliability based on the distributed trust scores from the verified users to the unknown user. SybilDefender [142] claims that a user is a spammer if it tightly links to a small number of users. Commonly, these studies limit the number of spammers based on the underlying assumption that the number of attack edges is relatively small. However, this assumption can be hardly supported in crowdsourcing environments because a spam worker can easily obtain many attack edges via collusions in shadow tasks. From another perspective, community detection models, e.g., Louvain [14] and Infomap [117], that leverage clustering to detect communities in bipartite networks can be referred to detect spam workers. In addition, Deepwalk [111] classifies nodes in a social network by representing each node as a unique vector is another promising method to identify spam workers. However, in these models, the attack edges and the trust values in edges are not considered and thus are vulnerable to the spam workers who can tightly connect with honest communities via their colluding accomplices.

To sum up, in crowdsourcing environments, verification-based spam worker defense models are effective in particular cases but commonly need to consume extra resources that may diminish or even eliminate the cost-effectiveness of crowdsourcing. Regarding reputation-based spam worker defense models, they are commonly vulnerable to the spam workers who collude with accomplices to counterfeit transaction records and trust edges (i.e., guise **G1** and **G2**). In addition, no trust network-based spam worker defense model has been proposed in crowdsourcing environments although the studies in other domains have shown the effectiveness of trust network-based spam worker defense models in defending against the spam workers with guise **G1**. Moreover, the existing trust network-based spam worker defense models in other domains do not discuss dealing with spam workers with guise **G2**. Therefore, in this thesis, we devise two new trust network-based spam worker defense models to bridge the gap between the demand for defending against the spam workers with both guise **G1** and **G2** and the existing studies.

### 2.3.3 Worker Recommendation in Crowdsourcing

In a crowdsourcing platform, there exists a large number of workers who possess equal or very close reputations. If we only take workers' reputations into account, these workers are homogeneous in a task and thus possess an equal chance to participate in a task. However, in fact, these homogeneous workers have different performances in the tasks published by different requesters. This is because different requesters publish tasks with different requirements and expectations. The differences of requesters lead that different requesters differently trust two homogeneous workers, i.e., the *personalisation of trust*. Namely, two requesters may trust a worker to different extents, and a requester may also differently trust two workers who both have “good” reputations. Thus, how to generate personalized recommendations (i.e., the most appropriate workers) for the tasks published by a requester is another challenging problem in crowdsourcing environments. In fact, this problem can be solved from two perspectives: *worker-oriented recommendation* and *requester-oriented recommendation*.

- **Worker-oriented recommendation.** A worker-oriented recommendation model is the one that analyzes the information explicitly or implicitly provided a worker to obtain its preferences in participating tasks and then accordingly recommends the appropriate tasks published by a requester to the worker. In crowdsourcing environments, a few worker-oriented recommendation models have been discussed in [7, 75, 156, 94]. Ambati et al. [7] present a content-based recommendation system that models a worker's skills and interests as the basis of recommendations. By matching the interests of a worker in social networks and the descriptions of the published tasks in crowdsourcing environments, Difallah et al. [37] propose a recommendation framework to pick a suitable crowd for each crowdsourcing task. In [75], Konomi et al. take the mobile contexts into account for generating task recommendations for a target worker. In [156], Yuen et al. propose a task recommendation framework that leverages probabilistic matrix factorization to fit the matrices derived from historical transaction

---

records and then accordingly recommends tasks to workers in dynamic scenarios. In [94], Mao et al. employ content-based recommendation methods to automatically recommend tasks of software development for workers. Worker-oriented recommendation models help significantly improve the user experience of workers. Moreover, from the worker's perspective, Schnitzer et al. [122] design a survey to investigate the demands of workers on crowdsourcing tasks. Based on their study, the reward amount and the time required to complete a task are two factors of most concern to the crowdsourcing workers. However, they neglect that the workers tend to provide the information that can benefit themselves to obtain as many recommendations as possible.

- **Requester-oriented recommendation.** A requester-oriented recommendation model is the one that recommends workers to the tasks published by a requester according to the information explicitly or implicitly provided the requesters. This type of recommendation model suggests that it is more important to recommend a worker who is most likely to submit a satisfying answer in a task than to recommend a worker who is interested in the task. Targeting the spatial crowdsourcing tasks that require a group of workers to collaboratively complete the tasks, Gao et al. [44] propose a two-level-based recommendation framework to discover  $k$  cheapest worker groups that can satisfy both the constraints of spatial range and the skill requirements. However, in this work, they do not provide the method to evaluate the capacity of a worker. In the literature, there is a huge gap between the demand of requester-oriented recommendation and the existing works.

From the perspective of recommending suitable workers to a task published by a target requester, in this thesis, we propose a novel trust-aware recommendation model. This model solves the limitations of applying the existing recommendation techniques in crowdsourcing environments, which are reviewed below. Typically, traditional recommendation techniques can be classified into four types: *demographic factor-based*

---

*recommendation, content-based recommendation, collaborative filtering-based recommendation, and trust-based recommendation* [15, 147].

- **Demographic recommendation.** This type of recommendation technique classifies users into groups according to their personal attributes, e.g., age, gender, country, etc. After obtaining the groups, given a target user, the opinions of other users who belong to the same group with the target user are aggregated to generate recommendations for the target user [121]. However, the crowdsourcing workers may have different preferences though they have the common personal attributes, which makes demographic recommendation be ineffective in crowdsourcing environments.
- **Content-based recommendation.** This type of recommendation technique generates recommendations by matching an item's descriptions with a target user's preferences. Content-based recommendation techniques have been demonstrated to be effective in recommending unrated items to users, such as web pages [19], news [112], and books [102]. However, when an item has no available description or inaccurate descriptions, this type of technique becomes less effective. In crowdsourcing environments, the problem is salient as it is infeasible to wholly and accurately to describe each worker's capacity in a standard form.
- **Collaborative Filtering (CF).** CF is currently the most popular recommendation technique that generates recommendations based on users' similarity or items' similarity. For example, Chen et al. [25] present a multi-collaborative filtering recommendation algorithm in a Web 2.0 platform by mining different data sources.
- **Trust-based recommendation.** In recent years, trust-based recommendation techniques have been widely discussed based on a fact that a user tends to accept suggestions from trustworthy users/friends [65, 34, 66]. In [65], Jamali et al. combine trust with CF to improve the quality of recommendations in social

---

networks. In [34], Deng et al. propose a trust-based recommendation method for web services. In [66], Jia et al. calculate a trust degree by integrating the results derived from adopting different types of CF, and then they propose a multi-dimensional trust model for generating recommendations. In social networks, Liu et al. [87] propose a method to effectively find the optimal social path to evaluate the trust of a target service provider. Based on the trust evaluation results, the most trustworthy service providers are then recommended to a target service consumer. Furthermore, they propose a novel multiple foreseen path-based heuristic algorithm to further improve the utility of a found optimal social path [88].

## 2.4 Conclusion

In this chapter, we have broadly reviewed the studies of crowdsourcing models, the trust related works, and the trust issues in crowdsourcing environments. In particular, we have first classified the existing crowdsourcing systems into two categories and accordingly concluded the typical characteristics of the existing crowdsourcing systems. We have presented the meaning of trust and the properties of trust, which are the basis of understanding trust in crowdsourcing environments. Moreover, we also have discussed the influences of trust in crowdsourcing activities followed by the discussion on the necessity of selecting trustworthy workers. Finally, we have proposed our scheme for solving the trustworthy worker selection problem, which particularly decomposes the complex trustworthy worker selection problem into three challenging subproblems: *trust evaluation*, *spam worker defense*, and *worker recommendation*. Regarding each of the challenging problems, we also have reviewed the existing studies and pointed out their limitations that are solved in our proposed models.

## Chapter 3

---

# Trustworthy Worker Selection based on Context-aware Trust Evaluation

---

In crowdsourcing environments, when workers can easily obtain the permission for participating in tasks, some workers tend to quickly submit ambiguous answers to as many tasks as possible for pursuing the maximal profits rather than conscientiously working on tasks [109, 39, 137]. We name this behaviour as *distorted pursuit*. These untrustworthy workers can easily succeed in open-ended tasks and multi-choice tasks, such as completing a survey and choosing a label for an image. In addition, an untrustworthy worker may boost its reputation via participating in easy tasks or in the tasks published by its accomplices, which is called as *rank boosting* [61]. To alleviate the negative impacts of these untrustworthy workers on the reliability of a crowdsourcing system, context-aware trust evaluation is a promising method as it can thoroughly investigate a worker's trust in various contexts. However, in the literature, there lacks studies to investigate crowdsourcing contexts for preventing untrustworthy workers with the *distorted pursuit* and *rank boosting* behaviours from participating tasks. Moreover, the existing trust evaluation models applied in industry, e.g., overall answer approval rate-based trust models at Amazon Turk and CrowdFlower, also neglect the influences of contexts on a worker's trust.

Considering the impacts of contexts on selecting trustworthy worker, this Chapter presents a new context-aware trust evaluation model and a corresponding worker selection method. In particular, two crowdsourcing contexts, i.e., the context of task



---

type and the context of task reward amount, are taken into account for modelling a worker's trustworthiness. Based on a worker's trust in the two contexts, we propose a worker selection method to effectively discover a worker combination where the workers are most trustworthy in a specific context, namely in a task with a particular reward amount.

### **3.1 Two-Dimensional Context-Aware Trust Evaluation**

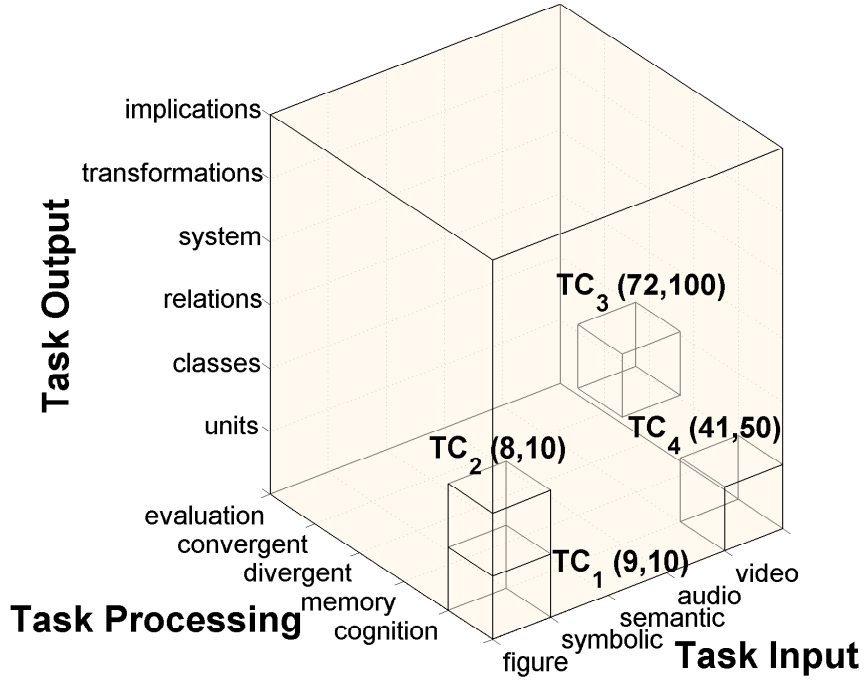
In crowdsourcing environments, there are two critical contexts that explicitly influence a worker's trustworthiness. The two contexts are the types of tasks and the reward amounts of tasks [7, 21, 74]. In general, a worker performs satisfactorily in its familiar types of tasks and the tasks that can be easily tackled. Thus, we first propose two types of crowdsourcing task classifications: *task type-based classification* and *task reward amount-based classification*. Based on them, we devise two types of context-aware trust metrics. In particular, they are Task Type-aware Trust (Tarust) and Task Reward Amount-aware Trust (RaTrust), respectively, for evaluating a worker's trust in the above listed two crowdsourcing contexts.

#### **3.1.1 Task Type-based Trust Evaluation**

##### **3.1.1.1 Task Type-based Classification of Crowdsourcing Tasks**

In crowdsourcing environments, any task has three essential components: *task input*, *task processing*, and *task output*. Taking a task that finding contact information of a toll manufacturer according to a given textual template as an example, this task consists of coordinates in three dimensions: *input* (a textual template), *processing* (finding contract information according to the example), and *output* (text messages). Accordingly, we can use the three components to group tasks to their corresponding class. In particular, we first build a three-dimensional intelligence space for classifying all tasks in Fig.1, where the three task dimensions are *task input*, *task processing*, and

task output, respectively.



**Figure 3.1:** An Intelligence Space for Human Intelligence Tasks Classification

In the dimension of task input, there are 5 types: *figure*, *symbolic*, *semantic*, *audio* and *video*, which are summarized from the task requirements at Amazon Turk. In addition, according to the structure of Guilford’s SI model [52] and the human processing in solving various crowdsourcing tasks, we conclude 5 basic types of task processing in crowdsourcing environments. In particular, they are *cognition*, *memory*, *divergent production*, *convergent production*, and *evaluation*, respectively. Moreover, we find 6 types of task output in crowdsourcing environments: *units*, *classes*, *relations*, *system*, *transformations*, and *implications*. Based on the task input, task processing and task output, a three-dimensional intelligence space, consisting of 150 (5\*5\*6) cubes, is established as a container for classifying tasks (see. Fig. 3.1). In the three-dimensional intelligence space, each cube represents a type of human intelligence requirements for workers, namely a type of crowdsourcing tasks requiring the same intelligence. For instance, a task that requires workers to find product in-

formation and a task that finds contact information are classified into the same cube  $C = \{semantic, cognition, units\}$  because they have the same input (semantic examples), the processing (cognition for examples), and the output (information units).

Moreover, based on the 3D intelligence space, a worker's historical task records in different types of tasks can be separately stored in the corresponding cubes. We call such a cube as a Trust Cube (TC). With trust cubes, we can comprehensively and specifically evaluate a worker's trust in the tasks belonging to different task types. In particular, given a worker, we can build a cube set  $TCs$  where each cube  $TC$  stores the number of the approval answers  $ha$  of the worker in the task type of the cube and the number of the submitted answers  $hs$  of the worker in the same task type. Based on it, a worker's answer approval rate of tasks  $hr$  in a cube can be calculated as  $hr = ha/hs$ . Consequently, a worker may have many fine-grained answer approval rates. Below, we further discuss how to use the fine-grained answer approval rates to evaluate a worker's trust in different types of tasks by using a trust metric called Task Type-aware Trust (TaTrust).

### 3.1.1.2 Task Type-aware Trust (TaTrust)

Suppose there is a worker  $w_i$  with historical records in  $TC_1, TC_2, TC_3$  and  $TC_4$ , we can build a trust cube set for it as depicted in Fig. 3.1. When the worker applies for a task, the value of the worker's Task Type-aware Trust (TaTrust) is differently influenced by records in  $TC_1, TC_2, TC_3$  and  $TC_4$ . We define the influence factor  $inf$  to represent the different degrees of influence. For example,  $inf_{(TC_1, TC_2)}$  represents the degree that the approval rate in  $TC_1$  influences the value of  $TaTrust$  when the upcoming task belongs to  $TC_2$ . The values of influence factor  $inf$  range from 0 to 1. The value 0 represents a worker's approval rate in  $TC_j$  does not influence the worker's trustworthiness, and 1 represents the influence is the max. We formalise the task type

based trust of worker  $w_i$  as follows:

$$TaTrust = \sum_{k=0}^3 \left( \frac{\sum_{i=1}^m \sqrt[4-k]{ha_{(k,i)}} \mathit{tinf}_{(k,i)}}{\sum_{i=1}^n hs_{(k,i)}} \right), \quad (3.1)$$

where  $m$  is the number of trust cubes in which the worker  $w_i$  possesses historical records. And  $k$  is the number of the same coordinates between two trust cubes. For example, when we need to evaluate  $w_i$ 's  $TaTrust$  in an upcoming task whose type is same with that of  $TC_1$ ,  $k$  of  $TC_2$  is set to 2 as the coordinates of  $TC_2$  and  $TC_1$  are same in 2 dimensions: task input and task processing.

The influence factor  $\mathit{tinf}$  is formalized as a function of  $k$  and  $ha$  by using the sigmoid function in Eq. (3.2).

$$\mathit{tinf}_{(k,i)} = \frac{1}{1 + \exp(-g_i(ha, ha_{(k,i)}, k))}, \quad (3.2)$$

where,  $g(ha, ha_{(k,i)}, k)$  is regarded as the independent variable of  $\mathit{tinf}$ , and  $g(ha, ha_{(k,i)}, k)$  is a monotonically increasing function of  $k$  and  $ha$ .

Influence factor  $\mathit{tinf}$  is formalised based on three characteristics. First, the marginal influence of task records belonging to one  $TC$  should be diminishing on a real crowdsourcing platform. Thus, the gradient changes should be narrowing when the value of  $\mathit{tinf}_{(k,i)}$  approaches to the border. We use the sigmoid function to model this characteristic in Eq. (3.2).

In addition, the influence factor  $\mathit{tinf}$  between two  $TC$ s is determined by  $k$  which represents the number of the same coordinates between  $TC$ s. For example, if an upcoming task belongs to  $TC_1$  in Fig. 3.1, the influence factor of each approval task in  $TC_2$  where  $k = 2$  should be larger than each approval task in  $TC_4$  where  $k = 1$ , i.e.,  $\mathit{tinf}_{(TC_2, TC_1)} > \mathit{tinf}_{(TC_4, TC_1)}$ . Because, the coordinates of  $TC_2$  and  $TC_1$  are same in 2 dimensions: task input and task processing, while the coordinates of  $TC_4$  and  $TC_1$  are same in one dimension: tasks processing.

Moreover, the differences, existing in the numbers of approval answers in different

$TCs$ , may affect the value of the influence factor  $tinf$ . Suppose worker  $w_a$ 's answer approval rate in  $TC_2$  is 80% ( $ha_{w_a} = 8$  and  $hs_{w_a} = 10$ ) and worker  $w_b$ 's answer approval rate in  $TC_2$  is also 80% ( $ha_{w_b} = 80$  and  $hs_{w_b} = 100$ ), worker  $w_b$  is more trustworthy than worker  $w_a$  because worker  $w_b$  completes more tasks with the same approval rate.

We assume the influence generated by the number of approved tasks exponentially drops with the change of  $k$ . Based on the independent variables  $k$  and  $ha$ , the independent variable  $g(ha, ha_{(k,i)}, k)$  is defined in Eq. (3.3).

$$g_i(ha, ha_{(k,i)}, k) = \begin{cases} \frac{{}^{(4-k)}\sqrt{ha_{(k,i)}} - ha}{MIN(ha, {}^{(4-k)}\sqrt{ha_{(k,i)}})}, & ha \neq 0 \\ {}^{(4-k)}\sqrt{ha_{(k,i)}}, & ha = 0 \end{cases} \quad (3.3)$$

where  $k \in \{0, 1, 2, 3\}$  depends on the same coordinates among  $TCs$ . Based on Eqs. (3.1), (3.2), and (3.3), the calculation of task type based trust  $TaTrust$  is defined as follows,

$$TaTrust = \sum_{k=0}^3 \left( \frac{\sum_{i=1}^m \frac{{}^{(4-k)}\sqrt{ha_{(k,i)}}}{1 - \exp(-g_i(ha, ha_{(k,i)}, k))}}{\sum_{i=1}^m hs_{(k,i)}} \right). \quad (3.4)$$

Regarding that a worker has different degrees of trustworthiness in tasks belonging to different task types, we reasonably model the calculation of a worker's  $TaTrust$  by presenting the rationale of aggregating a worker's historical records in different trust cubes. In fact, we calculate a worker's  $TaTrust$  in each type of tasks to specifically reflect the worker's trustworthiness in each context.

### 3.1.2 Reward Amount-based Trust Evaluation

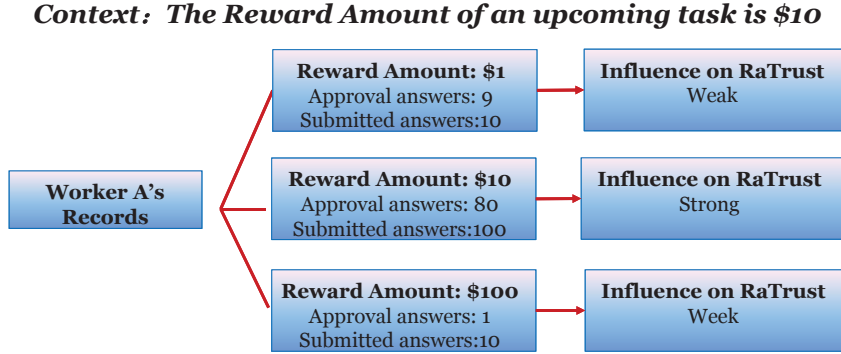
A worker's trustworthiness may vary in the upcoming tasks with different difficulty levels. However, there is no effective indicator to directly quantify difficulty levels for tasks in crowdsourcing environments. Even the tasks belonging to the same task type

may also have different difficulty levels. Nevertheless, the reward amount of a task can indirectly reflect the difficulty level of the task. From this perspective, we first propose another type of task classification based on the reward amounts of tasks.

### 3.1.2.1 Task Reward Amount-based Classification of Crowdsourcing Tasks

In e-commerce, the transaction amount (price) has been proved to be a vital attribute in evaluating a seller's contextual trust level [139, 159]. For example, a seller, who usually sells expensive goods with a good reputation, is regarded to be trustworthy in an upcoming transaction with a lower price. In our trust model, we go further to consider that all task records with different reward amount jointly influence a worker's reliability in a task with a fix reward amount. In crowdsourcing, a worker, who performs well in a range of reward amounts, is likely to be trustworthy in the tasks belonging to the same range. However, once the reward amount of an upcoming task is much higher or lower than the reward amount of tasks that the worker used to participate in, the worker's performance may change with a high probability. Thus, given an upcoming task with a reward amount of  $re^*$ , then those tasks rewarded between  $\alpha * re^*$  and  $\beta * re^*$  are classified into one type, where  $\alpha$  and  $\beta$  are constants. Here, we have to guarantee that those tasks with closely reward amounts should be classified to close groups. To realize this, we calculate a relative distance as  $d = \frac{\max(re^*, re_i)}{\min(re^*, re_i)}$ . Given an upcoming task with value  $re^*$ , we can calculate the relative distance  $d$  from it to another task with value  $re_i$  as the classification indicator. Based on the value of  $d$ , we classify other tasks to the corresponding groups where each group has a unique distance-based similarity  $d'$  that is calculated in Eq. (3.5).

$$d' = \begin{cases} 1, & \text{if } 0 < \frac{\max(re^*, re_i)}{\min(re^*, re_i)} < 1 \\ 2, & \text{if } 1 < \frac{\max(re^*, re_i)}{\min(re^*, re_i)} < 10 \\ 3, & \text{if } 10^1 < \frac{\max(re^*, re_i)}{\min(re^*, re_i)} < 10^2 \\ \dots, & \dots \\ h, & \text{if } 10^{(h-1)} < \frac{\max(re^*, re_i)}{\min(re^*, re_i)} < 10^h \end{cases}, \quad (3.5)$$



**Figure 3.2:** An Example of Differentiating Reward Amount-based Distance

where, the ranges for  $d$  are set as the times of 10 as we get the maximum ratio between two task with different reward amount is below 1000 through investigating the tasks at Amazon Turk.

### 3.1.2.2 Task Reward Amount-aware Trust (RaTrust)

Based on the task reward amount-based task classification, we calculate a worker's Reward Amount-aware Trust (RaTrust) by aggregating its historical approval records in tasks with different reward amounts. In particular, given a worker, we can first represent its records in one type of task as  $H_i = \{\widetilde{ha}_i, \widetilde{hs}_i, d'\}$ , in which  $\widetilde{ha}_i$  is the approval number of answers,  $\widetilde{hs}_i$  is the number of submitted answers, and  $d'$  is the distance-based similarity.  $RaTrust$  represents the trustworthiness of a worker in an upcoming task with a specified reward amount, which is defined in Eq. (3.6):

$$RaTrust = \sum_{d'=1}^h \left( \sum_{i=1}^{n_{d'}} \frac{\sqrt[p]{\widetilde{ha}_{(d',i)} \text{rinf}_{(d',i)}}}{\widetilde{hs}_{(d',i)}} \right), \quad (3.6)$$

where  $n_{d'}$  is the number of tasks in the task group with  $d'$  value, and  $0 < \text{rinf} < 1$  is a reward amount-aware influence factor determined by  $d'$  and  $\widetilde{ha}_{(d',i)}$ . First, approval answers in those tasks with low values of  $d'$  should influence the  $RaTrust$  more than those tasks with high values of  $d'$ . For example, in Fig. 3.2, given an upcoming task with the reward amount \$10, a worker A's past transaction records in the tasks with

reward amount \$10 should influence the final trustworthiness of the worker in this context rather than those records in the tasks with \$1 or \$100 reward amounts. Second, the influence of each approval task increases when the total number of approval tasks increases. Besides, the marginal influence of records diminishes when the value of  $rinf_{(d',i)}$  approaches the border. Accordingly,  $rinf_{(d',i)}$  is defined in Eq. (3.7):

$$rinf_{(d',i)} = \frac{1}{1 + \exp(-(z_i(\widetilde{ha}, \widetilde{ha}_{(d',i)}, d')))}, \quad (3.7)$$

where,  $z_i(\widetilde{ha}, \widetilde{ha}_{(d',i)}, d')$  is the independent variable of  $rinf_{(d',i)}$ . In addition, the function for calculating  $z_i(\widetilde{ha}, \widetilde{ha}_{(d',i)}, d')$  is defined as a monotonically increasing function of  $d'$  and  $\widetilde{ha}$  in Eq. (3.8).

$$z_i(\widetilde{ha}, \widetilde{ha}_{(d',i)}, d') = \begin{cases} \frac{d' \sqrt{\widetilde{ha}_{(d',i)} - \widetilde{ha}}}{\min(\widetilde{ha}, d' \sqrt{\widetilde{ha}_{(d',i)}})}, & \widetilde{ha} \neq 0 \\ d' \sqrt{\widetilde{ha}_{(d',i)}}, & \widetilde{ha} = 0 \end{cases} \quad (3.8)$$

The rationale behinds the RaTrust is that a worker who performs well in tasks whose reward amounts are close to that of an upcoming task are reasonably own higher trust values in the upcoming task than another worker who performs well in tasks whose reward amounts are far away from that of the upcoming task. By adopting this trust evaluation method, a worker's trustworthiness is effectively distinguished in different contexts where the task reward amounts are different.

### 3.2 A Worker Selection Algorithm based on Context-aware Trust

Regarding a group of published crowdsourcing tasks, multiple workers are needed to participate in the crowdsourcing processes of the tasks. An intuitive way is to select trustworthy workers is to rank them according to with their trust scores derived from

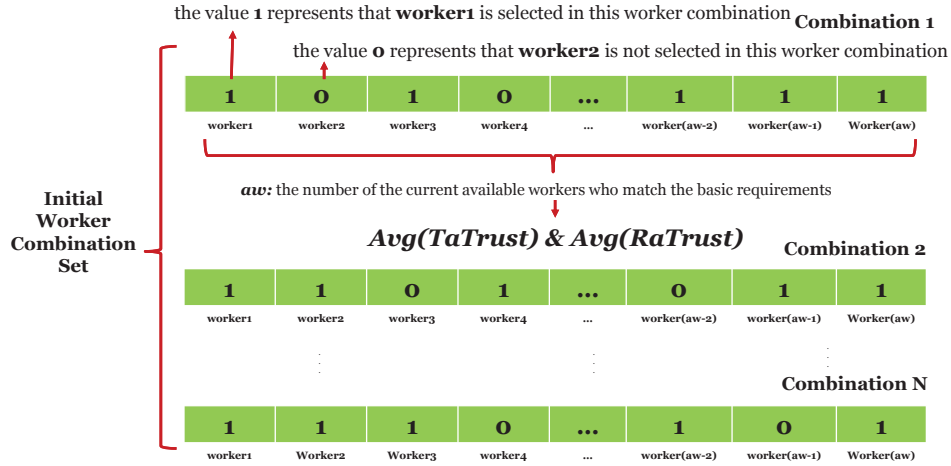


past transaction records. For example, the trust models proposed in [139, 140, 159] commonly calculate a trust score for each user with preset weights for aggregating the influence of different factors on the trust score and then regard a worker owning a high trust score as a trustworthy worker. The advantage is that different users' trustworthiness can be directly compared according to their trust scores. However, one obvious drawback is that it is nearly impossible to eliminate the subjective bias on the trust scores caused by the preset weights. Thus, based on our proposed two context-aware trust metrics: TaTrust and RaTrust, we model the trustworthy worker selection problem as a multi-objective combinatorial optimization problem that can be solved without setting subjective weights.

In order to find a worker combination whose values in the objectives cannot be optimized anymore, we propose a modified evolutionary algorithm based on NSGA-II [31]. In particular, there are two objectives in our model, i.e., the average value of TaTrust and the average value of RaTrust of a worker combination. A worker combination is called an efficient solution when the value of any one of TaTrust and RaTrust cannot be improved anymore without degrading the values of the other one. Our proposed algorithm is based on NSGA-II because it has been demonstrated to be efficient in solving multi-objective optimization problems.

### 3.2.1 Modelling Multi-Objective Worker Selection Problem

When a task is published on a crowdsourcing platform, a corresponding task status vector  $TSV = \{wn, aw, ar, TC, re\}$  is also generated. Here,  $wn$  represents the fixed number of the workers required by a requester for solving the tasks and  $aw$  is the number of the currently available workers who match the basic requirement  $ar$  (i.e., the answer overall approval rate). In addition,  $TC$  and  $re$  represent the trust cube to which the task belongs and the reward amount of the task, respectively, which are used to calculate context-aware trust  $TaTrust$  and  $RaTrust$ . Then, we formalise the trustworthy worker selection problem into a multi-objective combinatorial optimization in



**Figure 3.3:** An Example of the Initial Worker Combinations

Eq. (3.9),

$$f(X) = \text{minimize} \left( \frac{wn}{\sum_{i=1}^{wn} TaTrust_{(i)}}, \frac{wn}{\sum_{i=1}^{wn} RaTrust_{(i)}} \right), \quad (3.9)$$

*s.t*  $X \in D$

where  $0 < wn \ll aw$ , and  $X$  represents a feasible solution, i.e., a worker combination with number  $wn$  selected workers and number  $(aw - wn)$  refused workers. In a feasible solution  $X = \{x_1, x_2, \dots, x_i, \dots, x_{aw}\}$ , the value of  $x_i$  is 0 or 1.  $x_i = 0$  represents that  $worker_i$  is not selected in the current feasible solution, and  $x_i = 1$  represents that  $worker_i$  is selected in the current feasible solution. Given a solution  $X_j$ , it has a corresponding image point  $pt_j$  in the objective space, which consists of the average values of two context-aware trust:  $\overline{TaT}_{X_j}$  and  $\overline{RaT}_{X_j}$ , respectively. Given another solution  $X_l$  that has a point  $pt_l = \{\overline{TaT}_{X_l}, \overline{RaT}_{X_l}\}$ , if  $\overline{TaT}_{X_j} \geq \overline{TaT}_{X_l}$  and  $\overline{RaT}_{X_j} \geq \overline{RaT}_{X_l}$ , point  $pt_j$  dominates  $pt_l$ . Corresponding, solution  $X_j$  dominates  $X_l$ . If no  $X_j$  dominating  $X_l$  can be found in  $D$ ,  $X_l$  is called an efficient solution. The image point of  $X_l$  is called a non-dominated point. In general, the efficient solution is not unique. Thus, the set consisting of all efficient solutions is named as the efficient set. The assemble consisting of the image points of efficient solutions in the objective space is named *Pareto front*. Our ultimate objective is to select the trustworthy worker combinations with image points falling in the *Pareto front*.

### 3.2.2 A Multi-Objective Worker Selection Algorithm

The processes of the multi-objective worker selection are as follows.

**Step 1:** Generate an initial worker combination set  $WS = \{W_i\}_{i=1}^N$  that contains number  $N$  initial worker combinations. In the set, each  $W$  satisfies  $W = \{w_j\}_{j=1}^{aw}$  and  $\sum_{j=1}^{aw} w_j = wn$ .  $W$  represents a worker combination, and  $wn$  is the number of required workers. In Fig. 3.3, an example of the initial worker combinations is presented.

In NSGA-II, the initial solutions are randomly generated to provide a strong global search ability for the algorithm. The drawback is that searching from a random initiated solution costs much more time than starting with a better solution set. Thus, we modify this step by increasing the possibility of selecting those workers who possess obvious good reputations. First, a worker set is generated after sorting all workers according to  $\bar{T}$ . The  $SumT$  is the sum of task type based trust TaTrust and reward amount based trust RaTrust, i.e.,  $SumT = TaTrust + RaTrust$ . We set an initial selection probability  $pr_k = \frac{\bar{T}_k}{\bar{T}_{max}}$  for each worker.

**Step 2:** For each worker combination  $W_i$  in  $WS$ , its fitness  $fit_{W_i}$  (non-domination level) and density-estimation metric  $den_{W_i}$  are calculated by adopting the same methods proposed in NSGA-II.

In the first stage, all worker combinations that belong to the first non-dominated front are identified by comparing the trust values of all the number  $N$  worker combinations. Likewise, after stripping out the first non-dominated front, the second front can be identified. All  $W$ s are divided into corresponding non-dominated fronts by repeating this procedure. The value of a non-dominated front level (e.g., 1 represents the first non-dominated front) is used as the  $fit$  of a worker combination in the level. As the priority of  $W$ s in the same level cannot be determined by simply relying on the  $fit$ , we further calculate the density-estimation metric  $den$ , which is determined by the distance of each objective among current worker combination and the nearest

combinations in Eq. (3.10),

$$den = \frac{|TaTrust^+ - TaTrust^-| + |RaTrust^+ - RaTrust^-|}{TaTrust^{max} - TaTrust^{min} + RaTrust^{max} - RaTrust^{min}}. \quad (3.10)$$

**Step 3:** Select number  $\frac{|N|}{2}$  worker combinations set from  $WS$  by using usual binary tournament strategy. Crossover and mutation operators are executed to generate number  $p$  offspring worker combinations  $WS^{off} = \{W_l^{off}\}_{l=1}^p$ . In our devised algorithm, the crossover operator is modified to satisfy the constraint that the number of selected workers is fixed. In addition, in our MOWS\_GA, we modify the mutation operator to be an adaptive variable  $\sigma = \sigma\gamma$ , which is calculated in Eq. (3.11),

$$\gamma_{(i,iter)} = \begin{cases} \min\left(\frac{\sum_{i=1}^{\frac{|N|}{2}} \frac{\bar{T}_{(i,iter-1)} - \bar{T}_{(i,iter-2)}}{\bar{T}_{(i,iter)} - \bar{T}_{(i,iter-1)}}}{n}, 1\right), & \text{if } iter \geq 2 \\ 1, & \text{if } iter < 2 \end{cases}. \quad (3.11)$$

The  $\gamma$  represents the increasing ratio of trust values between two evolutions. If the increase of the ratio is more significant than the last time, which means the mutation promotes the increase of the trust values. Thus, we use  $\gamma$  to decline the value of  $\sigma$  to avoid time consumption in searching other directions. We set 0.2 as the maximum value of  $\sigma$ , because frequent mutations influence the convergence of the algorithm.

**Step 4:** Use elitism to select a worker combination set (size  $N$ ) from  $WS \cup WS^{off}$ . The combinations in the set are stored in  $WS^{new}$ . Adopt the same strategy to cross and mutate  $WS^{new}$ , and let the result to replace  $WS$ , i.e.,  $WS = WS^{new}$ .

**Step 5:** Check whether the termination condition is satisfied. Once the number of iterations  $iter$  reaches the preset maximal value or no new dominated solution appears during 10 consecutive iterations, the algorithm is terminated. Otherwise, go to **Step 2**.

The complexity of non-dominated sorting is  $\mathcal{O}(M * (2N)^2)$ , in which  $M$  is the number of optimal objectives and  $N$  is the number of worker combinations. As the number of optimal objectives is commonly a small positive integer, the overall complexity of our algorithm is  $\mathcal{O}(iter_{max} * N^2)$ , where  $iter_{max}$  is the maximum number

of iterations.

### 3.3 Experiments and Analysis

In this section, we evaluate our proposed context-aware crowdsourcing worker selection model *CrowdTrust* in a scenario where a requester publishes a group of tasks where 100 workers from 1000 workers are required. The overall answer approval rates of the 1000 workers are preset to satisfy the requirements in the published tasks. Below, we present the experiment settings first followed by the experimental results and the analysis of the results.

#### 3.3.1 Experiment Setting

As no available dataset includes completed historical transaction records of workers in crowdsourcing environments, we generate synthetic data for simulations. In the synthetic dataset, some workers are preset to have *distorted pursuit* and *rank boosting* behaviours. The simulations are conducted for selecting trustworthy worker combinations from the dataset.

**Table 3.1:** Constraints for Generating 1000 Workers

Behaviour	Percentage	Constraints
Distorted Pursuit	15% (obvious)	$k = rand(0,1,2),$ $d' = rand(2,3,4)$
	5% (marginal)	$k = rand(1,2,3),$ $d' = rand(1,2,3)$
Rank Boosting	20%	$k = rand(0,1), d' =$ $rand(3,4)$ or $k = rand(0,1,2),$ $d' = rand(3,4)$
Honest	60%	$k = rand(2,3)$ $= rand(1,2)$

In the synthetic dataset, workers' historical task records are randomly generated

with a series of constraints listed in Table 3.1. We generate the transaction records for 1000 workers in different types of tasks (i.e., in different  $TCs$ ) and in different reward amount ranges as follows.

In particular, the numbers of workers' submitted tasks are randomly generated from a normal distribution. And, each worker's answer overall approval rate is randomly generated in the range of 90% and 95%, because only a worker with an overall answer approval rate of 90% or above can participate in the tasks on a crowdsourcing platform, e.g., Amazon Turk.

Among the 1000 workers, 150 workers (i.e., 15% of the 1000 workers) with obviously *distorted pursuit* behaviours and 50 workers (i.e., 5% of the 1000 workers) with marginally *distorted pursuit* behaviours are generated. Setting marginal workers is because that there are some workers with *distorted pursuit behaviours* may apply for tasks that they are good at in a period. Thus, the marginal workers who possess a certain number of honest records in the tasks similar to the upcoming one are generated in the simulations. In addition, 200 dishonest workers (i.e., 20% of the 1000 workers) with *rank boosting* behaviours are generated.

Furthermore, we generate 600 honest workers (60% of 1000 workers). Honest workers' records are generated in those tasks that have similar types and reward amounts to the upcoming task.

The parameters applied in the multiple-objective worker selection algorithm are listed in Table 3.2.

**Table 3.2:** Parameters for the Multiple-Objective Worker Selection Algorithm

Objective Variable	Decision Variable	Population Size (N)
2	1000	20
<b>Max Iterations</b> ( $iter_{max}$ )	<b>Crossover probability</b> ( $\zeta$ )	<b>Mutation probability</b> ( $\sigma$ )
500 / 1000	0.9	0-0.2

In our simulations, there are 2 objective variables: TaTrust and RaTurst and 1000 decision variables. In order to increase the capability of global search in our devised algorithm, the value of crossover probability  $\zeta$  is set to 0.9. Mutation probability  $\sigma$  is

modified to be an adaptive one in the range of 0-0.2, which can avoid time consumption caused by excessive mutations. Considering the efficiency of our proposed algorithm, the population size of initial worker combinations is set to a relatively small size 20, and the maximum numbers of iterations are set to 500 and 1000, respectively.

### 3.3.2 Performance Comparison in Trustworthy Worker Selection

In the literature, no context-aware solution has been reported for selecting trustworthy workers in crowdsourcing environments. Thus, we compare the performance difference between the Answer Approval Rate-based Random Worker Selection model ARS and our proposed Context-aware Trust Evaluation based Worker selection model CrowdTrust. In a crowdsourcing platform, e.g., Amazon Turk, if the overall answer approval rate of a worker who wants to participate in a task satisfies the preset requirement (in general 90%), they are automatically selected on the first-come-first-serve basis. Because all workers in our synthetic dataset are randomly generated with an overall approval rate above 90%, ARS first selects 20 random worker combinations. Each combination has 100 workers. Then, we calculate TaTrust and RaTrust for each worker combination, and compare the values of them with the results of the worker combinations selected by CrowdTrust.

**Result 1.** Fig. 3.4 plots the trust values of worker combinations selected by ARS and CrowdTrust, respectively. From Fig. 3.4, we can observe that the best TaTrust and RaTrust values in ARS are 0.71 and 0.73 respectively. By contrast, the best TaTrust and RaTrust values in the worker combinations selected by CrowdTrust are 0.805 and 0.81 respectively, which are 13.4% and 10.9% higher than the ones delivered by ARS. Thus, our proposed CrowdTrust can select worker combinations with on average 10% higher context-aware trust values than the trust values of worker combinations selected by ARS.

**Result 2.** Fig. 3.5 plots the average numbers of untrustworthy workers and honest workers in the worker combinations selected by ARS and CrowdTrust respectively.

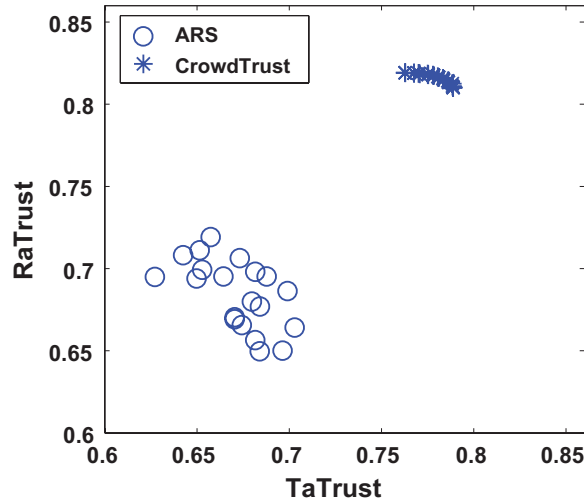


Figure 3.4: The Comparison in TaTrust and RaTrust

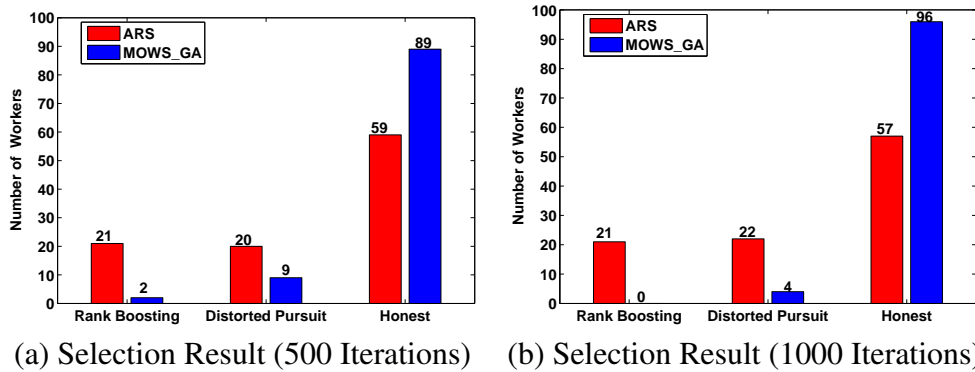


Figure 3.5: The Comparison in Trustworthy Worker Selection

From Fig. 3.5, we can see that compared to ARS, CrowdTrust selects fewer workers with untrustworthy behaviours and more workers with honest behaviours.

In Fig. 3.5(a), after 500 iterations, 21 workers with *rank Boosting* behaviours and 20 workers with *distorted Pursuit* behaviours, are selected by ARS. This is close to 40% - the percentage of the workers with frauds in the dataset. By contrast, the total number of workers with these behaviours selected by CrowdTrust is 11 only, which is 73.2% less than the one selected by ARS. In addition, 89 honest workers are selected by CrowdTrust, which is 50.9% more than the number delivered by ARS.



Fig. 3.5(b) plots the results of 1000 iterations. From Fig. 3.5(b), we can see that the numbers of workers who perform *rank boosting* or *distorted pursuit* frauds are 21 and 22 respectively in ARS. By contrast, the numbers are 0 and 4 respectively in CrowdTrust. Through observing the trust values of the 4 selected workers with *Distorted Pursuit* behaviours, we find that the 4 workers are all marginal workers, who possess a certain number of honest records in the tasks similar to the upcoming one. In addition, the number of honest workers selected by CrowdTrust is 96, which is 68.4% more than that of ARS.

From the above results, we can conclude that our proposed CrowdTrust can select worker combinations with higher TaTrust and RaTrust values than ARS and effectively prevent workers with untrustworthy behaviours from participating in tasks.

### 3.4 Conclusions

In this chapter, we have proposed two methods for classifying crowdsourcing tasks based on task type and task reward amount, respectively. Based on the classifications, a context-aware trust model CrowdTrust, for calculating the task type based trust *TaTrust* and task reward amount based trust *RaTrust*, has been proposed. To the best of our knowledge, this is the first solution in the literature to evaluating workers' trust from the perspective of contextual awareness. For solving the trustworthy worker combinations selection problem with two context-aware trust objectives, which is NP-Hard, we have proposed a new evolutionary algorithm based on NSGA-II. The results of experiments conducted on a synthetic dataset have demonstrated that CrowdTrust can effectively identify dishonest workers. However, our proposed approach may be vulnerable to attack when workers counterfeit fake records and apply for tasks in a specific type of tasks. Thus, in the following two chapters, we further discuss the models for defending threats from these harmful workers.

## Chapter 4

---

# Spam Worker Defense based on Trust Vector

---

In the last chapter, we have discussed how to select reliable and capable workers based on the context-aware trust metrics. However, some spam workers may collude with their accomplices to obtain “good” reputations in particular contexts, and then cheat in the tasks under the contexts. The effective defense against spam workers has become a top-priority problem in crowdsourcing environments. However, this problem is extremely challenging because the low or even free transaction fee [22] and the availability of high-degree anonymity [155, 146] in crowdsourcing environments could allow spam workers to masquerade as “honest” workers via low-cost collusions [5]. In general, a spam worker can obtain two types of guises:

- **G1:** a spam worker can obtain “successful” transaction records, and thus possess a “good” reputation, by colluding with some requesters to manipulate the crowdsourcing outcomes in *shadow tasks*. A shadow task is one whose answer is preset and revealed to the colluding spam workers beforehand to ensure that they can succeed in the task.
- **G2:** a spam worker can collude with some requesters and workers who have obtained direct trust from some honest workers and requesters, and thus indirectly connects itself to the honest requesters via the edges between their colluding partners and the honest requesters. Such an edge is called an *attack edge* be-

cause the spam worker can leverage it to deceive honest requesters and then mount attacks in the tasks published by such requesters.

To defend against the threats from the spam workers, verification-based spam worker defense models and trust-aware spam worker defense models have been proposed in crowdsourcing environments.

However, regarding the 1st category (i.e., verification-based spam worker defense models), they become ineffective and expensive when facing numerous spam workers with the two guises because a large number of human resources are consumed in verifying the answers submitted by the spam workers [97].

In the 2nd category (i.e., trust-aware spam worker defense models), reputation-based spam worker defense models have been proposed to particularly target spam workers in crowdsourcing environments while trust network-based spam worker defense models have not attracted enough attention in crowdsourcing environments. In the literature, reputation-based spam worker defense models, such as sequential performance-based defense models [153, 90] and a consistency-based defense model [64], have a common intuition that a worker’s historical transaction records can genuinely indicate the worker’s reliability in future tasks. However, this intuition also determines that these models are vulnerable to the spam workers who possess many “successful” transaction records (i.e., guise **G1**). Regarding the trust network-based spam worker defense models, we are the first one to defend spam workers from this perspective. In fact, trust network-based defense models, such as SybilLimit [151] and SybilInfer [27], have shown the high effectiveness in limiting spammers in P2P networks with guise **G1** when a spammer only possesses few attack edges. In addition, In [46], Golbeck et al. calculate the average trust values along social trust paths to help a source participant infer whether the target participant is trustworthy. In [82], Jure. Leskovec et al. show an impressive result that the trust relationship between two nodes is more likely to be positive if the two nodes are trusted by many common neighboring nodes. In [138], Wang et al. take participants’ characteristics, mutual relations and contexts into account to infer the trust relation between two indirectly

connected participants. These works also are effective in helping a participant to infer if another participant is a spammer or not. However, they commonly neglect that spam workers with guise **G2** may receive the positive trust information propagated via the attack edges. These positive trust information can help the spam worker obtain a high trust value and thus look like an honest worker. Due to the drawbacks of the existing models, a new model that can effectively defend against the spam workers with both guises **G1** and **G2** is in a high demand.

Considering the drawbacks of the existing spam worker defense models proposed in crowdsourcing environments and the limitations of the spam worker defense models introduced in other online environments, in this chapter, we propose a novel trust vector-based spam worker defense model called CrowdDefense for preventing spam workers with guise **G1** and guise **G2** from participating in tasks. In particular, we first build a transaction-based Crowdsourcing Trust Network (CTN) where a requester and a worker who has transacted with him/her are directly connected via an edge recording their past transaction records. Based on such a CTN, we analyze three threat patterns of spam workers where spam workers collude with different types of accomplices to boost their reputations. According to the analysis of the three threat patterns, we propose a novel trust inference method to measure the trust relations between a worker and his/her indirectly connected requesters in the CTN. Then, we compute a worker's trust relations with different requesters and present them in a new Worker Trust Vector (WTV) to indicate the worker's global trust level. Based on it, we propose a novel worker selection method that investigates the WTVs to determine which worker should be permitted to participate in tasks. As spam workers only succeed in the transactions with their accomplices, they cannot obtain good trust values in their WTVs. Thus, spam workers can be prevented from participating in crowdsourcing tasks. The experiments demonstrate that CrowdDefense significantly outperforms the state-of-the-art approaches in terms of preventing spam workers from participating in the tasks published by honest requesters.

---

## 4.1 Crowdsourcing Trust Network-based Threat Analysis

### 4.1.1 Crowdsourcing Trust Network (CTN)

A Crowdsourcing Trust Network (CTN) is one that consists of requester nodes and worker nodes. In a CTN, only a requester and a worker who has transacted the requester are directly connected via an edge. Note that, in a CTN, there is no edge between any two requesters because they cannot transact with each other. Likewise, there is no edge between any two workers. In addition, a crowdsourcing user may have the roles of both a *requester* and a *worker*, we use a requester node and a worker node to represent the same user's requester role and worker role, respectively. Below, we first present a requester taxonomy and a worker taxonomy in crowdsourcing based on their identities that are determined by their past transaction behaviours and determine their future transaction behaviours. Then we present the definitions on the edges that connecting requesters with workers.

#### 4.1.1.1 Requesters in a CTN

We define three types of requester identities: (1) *Honest Requester*: an honest requester is one who publishes normal tasks, and then fairly verifies, approves, and rewards the answers submitted by workers; (2) *Grey Requester*: a grey requester is one who publishes normal tasks when it is not colluding with other workers. However, a grey requester also publishes shadow tasks to assist its colluding workers in obtaining good reputations (i.e., guise **G1**) and attack edges (i.e., guise **G2**); (3) *Spam Requester*: a spam requester is one who only publishes shadow tasks to assist its colluding workers in obtaining guises **G1** and **G2**.

#### 4.1.1.2 Workers in a CTN

We define three types of worker identities: (1) *Honest Worker*: an honest worker is one who properly participates in tasks, and then honestly submits answers that are believed by itself as the correct answers; (2) *Grey Worker*: a grey worker is one who honestly behaves in some tasks to obtain a good reputation and trust edges to honest requesters. But, a grey worker also colludes with spam requesters and grey requesters in shadow tasks to build trust edges between itself and the colluding requesters; (3) *Spam Worker*: a spam worker is one who submits junk answers in the tasks published by honest requesters. In addition, a spam worker colludes with spam requesters and grey requesters in shadow tasks to obtain both a good reputation (i.e., guise **G1**) and attack edges (i.e., guise **G2**). Such a spam worker can indirectly connect to some honest requesters and deceive them if the honest requesters trust the grey workers who possess trust edges to the spam worker's colluding requesters.

Note that, either a requester or a worker owns only one type of requester identities or worker identities in a time point.

#### 4.1.1.3 Transaction-based Edges in a CTN

**Definition 4.1. Direct Trust** is a trust metric that indicates the local trust relation between a requester and a worker, which is measured by the worker's reputation in the tasks published by the requester.

As the answer approval rate has been widely adopted as a reputation in many crowdsourcing platforms (e.g., Amazon Turk), we calculate the answer approval rate of worker  $w_j$  in the tasks published by requester  $r_i$  to indicate the direct trust relation between  $r_i$  and  $w_j$ , i.e.,  $dt_{(r_i, w_j)} = \frac{n_{apv(r_i, w_j)}}{n_{sub(r_i, w_j)}}$ ,  $0 \leq dt_{(r_i, w_j)} \leq 1$ . Here,  $n_{sub(r_i, w_j)}$  denotes the total number of the answers submitted by  $w_j$  in the tasks published by  $r_i$ , and  $n_{apv(r_i, w_j)}$  denotes the number of the approved answers submitted by  $w_j$  in the tasks. Based on the values of  $dt$  on edges in a CTN, we define two types of edges.

**Definition 4.2. Trust Edge** is a type of edges that connect a requester  $r_i$  and a worker

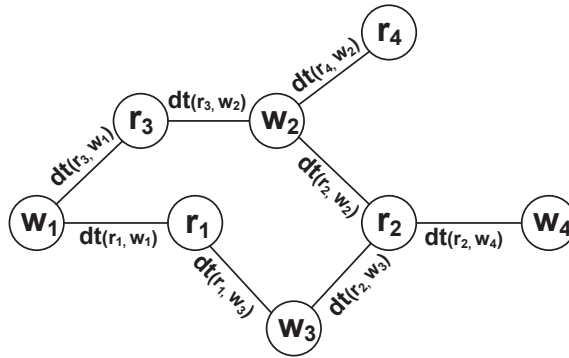
$w_j$  who directly trust each other in past transactions, on which  $dt_{(r_i,w_j)} \geq \varepsilon$ ;

**Definition 4.3. Distrust Edge** is a type of edges that connect a requester  $r_i$  and a worker  $w_j$  who directly distrust each other in past transactions, on which  $dt_{(r_i,w_j)} < \varepsilon$ .

Here,  $\varepsilon$  is a preset threshold. In practice,  $\varepsilon$  can be set as the value of the average answer approval rate of honest workers in normal tasks.

#### 4.1.1.4 The Formulation of a CTN

As a crowdsourcing user may have the roles as both a *requester* and a *worker*, we use two nodes to represent the same user’s worker role and requester role, respectively. Let  $R = \{r_i\}_{i=1}^{|R|}$  denote all the requester nodes,  $W = \{w_j\}_{j=1}^{|W|}$  denote all the worker nodes,  $TE = \{te_k\}_{k=1}^{|TE|}$  denote all the trust edges containing direct trust values, and  $DTE = \{dte_k\}_{k=1}^{|DTE|}$  denote all the distrust edges containing direct trust values, we can obtain a bipartite Crowdsourcing Trust Network  $CTN(R \cup W, TE \cup DTE)$ . A CTN example is depicted in Fig. 4.1



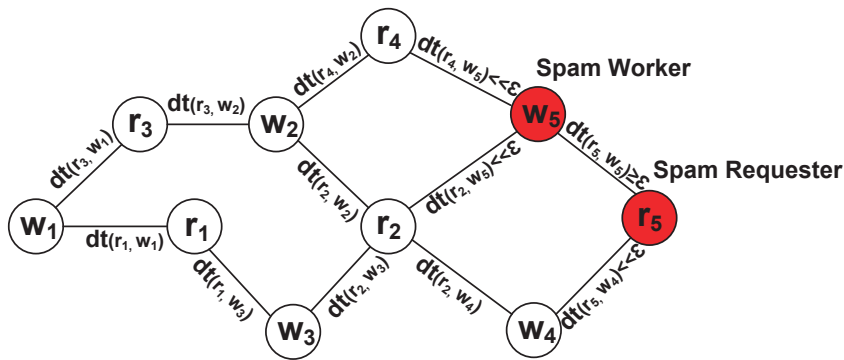
**Figure 4.1:** An Example of a Crowdsourcing Trust Network (CTN)

#### 4.1.2 Three Threat Patterns in a CTN

Typically, there are three threat patterns of spam workers who collude with different types of accomplices. Below, we present the three threat patterns in detail.

**4.1.2.1 Threat Pattern A (Colluding with Spam Requesters)**

In this threat pattern, a spam worker continuously succeeds in the shadow tasks published by its colluding spam requesters to maintain a “good” reputation. Based on it, the spam worker can bypass the reputation systems to submit random and erroneous answers in the tasks published by an honest requester. Fig. 4.2 depicts an example of Threat Pattern A in which spam worker  $w_5$  boosts his/her reputation by obtaining a high direct value from spam requester  $r_5$  in shadow tasks, i.e.,  $dt_{(r_5, w_5)} \geq \epsilon$ . By contrast, honest worker  $w_4$  who does not know the preset answer of a shadow task and thus is most likely to fail in the shadow tasks published by spam requester  $r_5$ , i.e.,  $dt_{(r_5, w_4)} \ll \epsilon$ . As  $w_5$  succeeds in a large number of shadow tasks,  $w_5$  has a “good” reputation, i.e., a high answer approval rate. If only this reputation is considered,  $r_2$  and  $r_4$  may choose to trust  $w_5$ . But,  $w_5$  is a spam worker who submits random and erroneous answers, leading to  $dt_{(r_2, w_5)} \ll \epsilon$  and  $dt_{(r_4, w_5)} \ll \epsilon$ . In addition, honest requesters  $r_1$  and  $r_3$  may still take  $w_5$  as an honest worker if  $w_5$  colludes in more shadow tasks to maintain a high answer approval rate. However, one of the characteristics of such a spam worker is that he/she cannot connect to other honest requesters via any edge with a high direct trust value.

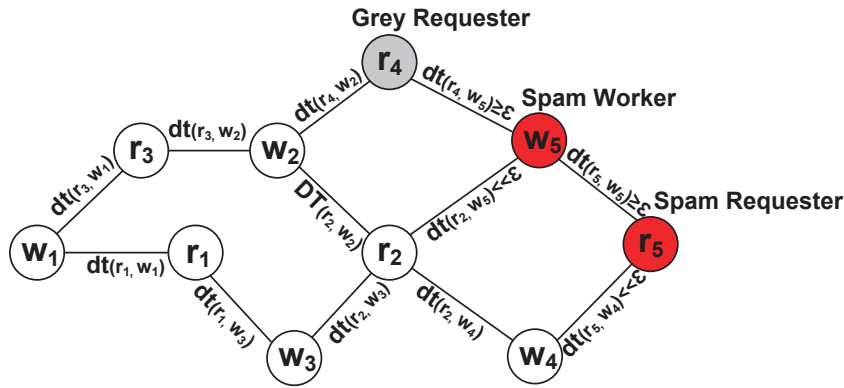


**Figure 4.2:** An Example of Threat Pattern A



**4.1.2.2 Threat Pattern B (Colluding with Spam Requesters and Grey Requesters)**

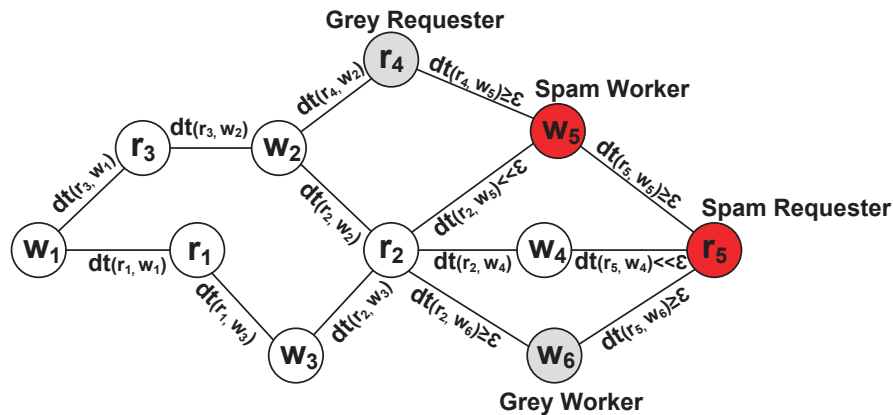
In Threat Pattern B, a spam worker transacts with both spam requesters and grey requesters in shadow tasks to boost his/her reputation. As a grey requester directly connects with some honest workers via trust edges with high direct trust values, the grey requester can indirectly connect spam workers with honest participants by also building trust edges connecting itself with spam workers via publishing some shadow tasks. Fig. 4.3 depicts an example of Threat Pattern B where spam worker  $w_5$  colludes with grey requester  $r_4$  in shadow tasks, and thus a trust edge with high direct trust value is built between them, i.e.,  $dt_{(r_4, w_5)} \geq \epsilon$ . Via this edge, the spam worker  $w_5$  can indirectly connect with honest requesters  $r_1, r_2$  and  $r_3$ . The existence of  $dt_{(r_4, w_5)} \geq \epsilon$  and  $dt_{(r_5, w_5)} \geq \epsilon$  makes the spam worker  $w_5$  look like an honest worker, which leads the spam worker  $w_5$  hardly to be recognized. In addition, with the assistance of grey requesters, spam workers and spam requesters can avoid forming a small clique that can be easily detected in a CTN.



**Figure 4.3:** An Example of Threat Pattern B

**4.1.2.3 Threat Pattern C (Colluding with Spam Requesters, Grey Requesters and Grey Workers)**

A spam worker first boosts his/her reputations by colluding with both spam requesters and grey requesters in shadow tasks. Moreover, the owner of the spam workers may bribe or create some grey workers to deeply conceal the spam worker via joining the spam worker’s colluding network. Consequently, such a spam worker indirectly connects with other honest requesters via both grey requesters and grey workers, which make itself much hard to be discovered. Fig. 4.4 depicts an example of Threat Pattern C where spam worker  $w_5$  obtains a good reputation, and directly connect with grey requester  $r_4$  via a trust edge with a high direct trust value (i.e.,  $dt_{(r_4,w_5)} \geq \epsilon$ ) by colluding in the shadow tasks published by grey requester  $r_4$ . In the meantime, spam requester  $r_5$  publishes shadow tasks to boost the reputations of spam worker  $w_5$  and grey worker  $w_6$ , and thus connect  $w_5$  with  $w_6$  via paths containing trust edges with high direct trust values (i.e.,  $dt_{(r_5,w_5)} \geq \epsilon$  and  $dt_{(r_5,w_6)} \geq \epsilon$ ). These edges help spam worker  $w_5$  indirectly connect with honest requesters  $r_1, r_2$  and  $r_3$  via paths containing trust edges with high direct trust values. The action provides camouflage for spam worker  $w_5$ , and also avoid spam worker  $w_5$  and their accomplices forming a clique.



**Figure 4.4:** An Example of Threat Pattern C

**Summary:** A spam worker indirectly connects with other honest participants via some fixed consisting of fixed trust edges. By contrast, an honest worker indirectly

connects to honest participants via multiple paths consisting of trust edges with high direct trust values. In addition, a spam worker submits random or erroneous answers to the tasks published by honest requesters. This leads to a high possibility that an honest requester indirectly connects with the spam worker via the paths that contain distrust edges with low direct trust values. Thus, in the following section, we aggregate the direct trust values along the paths connecting a worker with a requester to infer their trust relation. By analysing a worker’s trust relations with different requesters, we can judge if the worker is a spam worker.

## 4.2 Trust Vector-based Threat Defense

The analysis of the three threat patterns reveals that a spam worker connects to honest requesters via particular paths including fixed trust edges and dynamic distrust edges. As these paths cannot be counterfeited, with the direct trust values on the paths, we can infer the trust relation between a spam worker and a requester who are indirectly connected with each other. Based on it, we first investigate a workers’s trust relations with different requesters from the existing connections in a CTN. Then we devise a new worker trust vector (WTV) that represents a worker’s trust relations with different groups of requesters. After analysing a worker’s WTV, we can determine if the worker is a spam worker based on the opinions of different groups of requesters.

### 4.2.1 Trust Pheromone-based Trust Inference

In a CTN, each trust path connecting a requester and a worker contains the trust information between the requester and the worker. Below, we first define two types of paths: *trustworthy path* and *untrustworthy path*. Based on it, we define a new concept Trust Trace (TT) to represent the summarized trust information in each path.

**Definition 4.4. Trustworthy Path**  $tp_{(r_i, w_j)}^k$  is a  $k$ -hop path that starts from a requester  $r_i$  and ends at a worker  $w_j$ , in which each two directly connected nodes trust each

other.

For example, a path  $r_1 \xrightarrow{dt_{(r_1,w_1)} \geq \varepsilon} w_1 \xrightarrow{dt_{(r_2,w_1)} \geq \varepsilon} r_2 \xrightarrow{dt_{(r_2,w_2)} \geq \varepsilon} w_2$  is a trustworthy path between  $r_1$  and  $w_2$ .

**Definition 4.5. Untrustworthy Path**  $utp_{(r_i,r_e,w_j)}^k$  is a  $k$ -hop path that starts from a requester  $r_i$  and ends at a worker  $w_j$ , where any two directly connected nodes trust each other except that the last requester  $r_e$  in the path distrusts the worker  $w_j$ .

For example, a path  $r_1 \xrightarrow{dt_{(r_1,w_1)} \geq \varepsilon} w_1 \xrightarrow{dt_{(r_2,w_1)} \geq \varepsilon} r_2 \xrightarrow{dt_{(r_2,w_2)} < \varepsilon} w_2$  is an untrustworthy path between  $r_1$  and  $w_2$ .

**Definition 4.6. Trust Phormone**  $tp_{p(r_i,w_j)}$  is an aggregation of direct trust values in a trustworthy or an untrustworthy path between a source requester and a target worker, which reflects the trust relation between requester  $r_i$  and worker  $w_j$  in the path  $p_{(r_i,w_j)}$ . It can be calculated in Eq. (4.1).

$$tp_{p(r_i,w_j)} = \begin{cases} \frac{\sum_{l=1}^{len_{p(r_i,w_j)}} dt_l}{len_{p(r_i,w_j)}}, & \text{if } p_{(r_i,w_j)} \in TP_{(r_i,w_j)} \\ -\frac{\sum_{l=1}^{len_{p(r_i,w_j)}-1} dt_l}{len_{p(r_i,w_j)}-1}, & \text{if } p_{(r_i,w_j)} \in UTP_{(r_i,w_j)} \end{cases}, \quad (4.1)$$

where  $len_{p(r_i,w_j)}$  is the number of edges in path  $p_{(r_i,w_j)}$ ;  $TP_{(r_i,w_j)}$  is a set of trustworthy paths that all start from requester  $r_i$  and end at worker  $w_j$ ;  $UTP_{(r_i,w_j)}$  is a set of untrustworthy paths that all start from requester  $r_i$  and end at worker  $w_j$ ; In a trustworthy path  $tp_{(r_i,w_j)}$ , the trust phormone between requester  $r_i$  and worker  $w_j$  is set as the average direct value in the path. In addition, in an untrustworthy path  $utp_{(r_i,w_j)}$ , the trust phormone between requester  $r_i$  and worker  $w_j$  is set as the opposite of the average direct value in the path. Given a requester and a worker, by aggregating all trust phormone relating to the requester and the worker, we can measure the strength of trust (SOT) between the requester and the worker.

**Definition 4.7. Strength of Trust (SOT)** is a trust metric that indicates the degree of trust between two participants based on all the trust pheromone in the paths connecting the two participants.

Below, we present the calculation of SOT between a requester  $r_i$  and a worker  $w_j$ :

$$SOT_{(r_i, w_j)} = \frac{\sum_{u=1}^{|TP_{(r_i, w_j)}|} tph_u + \sum_{v=1}^{|UTP_{(r_i, w_j)}|} tph_v}{\sum_{q=1}^{|TP_{r_i}|} tph_q}, \quad (4.2)$$

where,  $TP_{(r_i, w_j)}$  is a set of trustworthy paths that start from requester  $r_i$  and end at worker  $w_j$ ;  $UTP_{(r_i, w_j)}$  is a set of untrustworthy paths that start from requester  $r_i$  and end at worker  $w_j$ ;  $TP_{r_i}$  is a set of trustworthy paths that start from requester  $r_i$  and end at any worker.

## 4.2.2 The SOT Estimation Algorithm

Given a requester  $r_i$  and a worker  $w_j$ , traversing all the  $r_i$ -started paths can exactly calculate the SOT from  $r_i$  to  $w_j$ . However, the complexity of traversal exponentially grows with the average number of edges  $n$  owned by a participant, i.e.,  $O(n^{hop_{max}})$ . Here,  $hop_{max}$  denotes the maximum number of the hops in a searching path. Thus, we present an SOT estimation algorithm (Algorithm 1) based on random walk below.

### 4.2.2.1 Algorithm Descriptions

Below we present the algorithm for estimating  $SOT_{(r_i, w_j)}$ .

**Step 1 (Line 1):** Initiate the currently searching node  $r_c = r_i$ , the currently searching path  $p_c = \{r_i\}$ , the trust pheromone about  $w_j$  as  $stph_{(r_i, w_j)} = 0$ , and the total trust trace about  $r_i$  as  $stph_{r_i} = 0$ ;

**Step 2 (Line 4):** Fill the set  $TW$  with the workers who are directly trusted by  $r_c$ , i.e.,  $dt_{(r_c, w \in TW)} \geq \varepsilon$ ;

**Step 3 (Line 5):** Calculate  $tph_{(r_i, w_k \in TW)}$  in the current searching path according

**Algorithm 1:** The SOT Estimation Algorithm

---

**Data:**  $CTN, r_i, w_j, dt, maxiter, \varepsilon, hops_{max}$   
**Result:**  $SOT_{(r_i, w_j)}$

- 1 **Set**  $r_c = r_i, stph_{(r_i, w_j)} = 0, stph_{r_i} = 0, iter = 1, TW = \emptyset, CR = \emptyset,$   
 $\overline{SOT}_{(r_i, w_j)} = 0, \overline{SOT}'_{(r_i, w_j)} = 0;$
- 2 **while**  $iter < maxiter$  **do**
- 3     **for** ( $hops = 1$  to  $hops_{max}$ ) **do**
- 4          $TW = \{w, dt_{(r_c, w)} \geq \varepsilon\};$
- 5         **Update**  $tph_{(r_i, w_k \in TW)}$  via Eq. (4.3);
- 6         **if** ( $dt_{(r_c, w_j)} \geq \varepsilon$ ) **then**
- 7             **Update**  $stph_{(r_i, w_j)}$  via Eq. (4.4);
- 8             **Update**  $stph_{r_i}$  via Eq. (4.5);
- 9         **else if** ( $0 \leq dt_{(r_c, w_j)} < \varepsilon$ ) **then**
- 10             **Update**  $stph_{(r_i, w_j)}$  via Eq. (4.6);
- 11             **Update**  $stph_{r_i}$  via Eq. (4.5);
- 12         **else if** ( $dt_{(r_c, w_j)} = null \ \& \ hops = hops_{max}$ ) **then**
- 13             **Update**  $stph_{r_i}$  via Eq. (4.5);
- 14          $cw = w_{random}, w \in TW;$
- 15          $CR = \{r, dt_{(r, cw)} \geq \varepsilon\};$
- 16          $r_c = r_{random}, r \in CR;$
- 17          $SOT'_{(r_i, w_j)} = SOT_{(r_i, w_j)};$
- 18         **Update**  $SOT_{(r_i, w_j)} = \frac{stph_{(r_i, w_j)}}{stph_{r_i}};$
- 19          $\overline{SOT}'_{(r_i, w_j)} = \overline{SOT}_{(r_i, w_j)};$
- 20          $\overline{SOT}_{(r_i, w_j)} = \frac{(iter-1)*\overline{SOT}_{(r_i, w_j)} + SOT_{(r_i, w_j)}}{iter};$
- 21         **if**  $2 \leq mod(iter, maxiter) \leq 5$  **then**
- 22             **Update**  $\sigma$  via Eq. (4.6);
- 23             **if**  $\sigma \leq \epsilon$  **then**
- 24                 **Return**  $SOT_{(r_i, w_j)};$
- 25                 **Break;**
- 26          $r_c = r_i;$
- 27          $iter = iter + 1;$
- 28 **Return**  $SOT_{(r_i, w_j)}$

---

to Eq. (4.3),

$$tph_{(r_i, w_k \in TW)} = \frac{\sum_{w_k \in TW} (\overline{dt}_{pc} * hop_{pc} + dt_{(r_c, w_k)})}{hop_{pc} + 1}, \quad (4.3)$$

**Step 4 (Lines 6-13):** Update the sum of the trust pheromone from  $r_i$  to  $w_j$ , i.e.,  $stph_{(r_i, w_j)}$ ;

1. **Case 1 (Lines 6-8):** If  $dt_{(r_c, w_j)} \geq \varepsilon$ , update  $stph_{(r_i, w_j)}$  according to Eq. (4.4) and update  $stph_{r_i}$  according to Eq. (4.5), and then go to **Step 5**;

$$stph_{(r_i, w_j)} = stph_{(r_i, w_j)} + \frac{\overline{dt}_{p_c} * hop_{p_c} + dt_{(r_c, w_j)}}{hop_{p_c} + 1}, \quad (4.4)$$

$$stph_{r_i} = stph_{r_i} + tph_{(r_i, w_k \in TW)}, \quad (4.5)$$

where,  $\overline{dt}_{p_c}$  is the average direct trust values in the current searching path  $p_c$ ;  $hop_{p_c}$  is the number of hops in the path  $p_c$ .

2. **Case 2 (Lines 9-11):** If  $dt_{(r_c, w_j)} < \varepsilon$ , update  $stph_{r_i}$  according to Eq. (4.5), update  $stph_{(r_i, w_j)}$  according to Eq. (4.6), and then go to **Step 5**;

$$stph_{(r_i, w_j)} = stph_{(r_i, w_j)} - \overline{dt}_{p_c}, \quad (4.6)$$

3. **Case 3 (Lines 12-13):** If  $dt_{(r_c, w_j)} = null$ , check the current searching hops to determine whether update  $stph_{r_i}$ ;

(1): If  $hop_{p_c} = hop_{max}$ , update  $stph_{r_i}$  according to Eq. (4.5), and then go to **Step 8**;

(2): If  $hop_{p_c} < hop_{max}$ , go to **Step 5**;

**Step 5 (Line 14):** Randomly select a worker from set  $TW$ , then fill set  $TR$  with the requesters who directly trust the worker;

**Step 6 (Line 16):** Randomly select a requester  $r'$  from  $TR$ , let the  $r'$  replace the current searching node  $r_c$ , go to **Step 2**;

**Step 7:** Check the current searching hops  $hop_{p_c}$ . If  $hop_{p_c} = hop_{max}$  go **Step 8**; otherwise go to **Step 5**;

**Step 8 (Line 18):** Update the  $SOT_{(r_i, w_j)} = \frac{stph_{(r_i, w_j)}}{stph_{r_i}}$ ;

**Step 9 (Lines 19-27):** Check whether the searching has satisfied any of the termination conditions, if any condition is satisfied, terminate; otherwise, replace the current searching node  $r_c$  with the requester  $r_i$  go to **Step 2**.

#### 4.2.2.2 Termination of Algorithm

This algorithm is terminated when the number of searching times exceeds the preset maximum value or the value of SOT converges. We define the function to judge the convergence in Eq. (4.7),

$$\sigma^2 = (SOT'_{(r_i, w_j)} - \overline{SOT'}_{(r_i, w_j)})^2 - (SOT_{(r_i, w_j)} - \overline{SOT}_{(r_i, w_j)})^2, \quad (4.7)$$

where,  $SOT'_{(r_i, w_j)}$  denotes the  $SOT$  between  $r_i$  and  $w_j$  in the previous searching, and  $\overline{SOT'}_{(r_i, w_j)}$  is the average value of  $SOT$  by the searching;  $SOT_{(r_i, w_j)}$  is the  $SOT$  value between  $r_i$  and  $w_j$  in the current searching, and  $\overline{SOT}_{(r_i, w_j)}$  is the average value of  $SOT$  by the current searching; When  $\sigma^2 \leq \epsilon$ , the algorithm terminates. The  $\epsilon$  can be set as the inverse of the number of edges in a CTN.

The worst time complexity of our algorithm is  $O_{(m * hops_{max} * n)}$ , where  $m$  is the maximum number of iterations,  $hops_{max}$  is the maximum searching hops, and  $n$  is the maximum number of edges starting from a requester.

#### 4.2.3 Worker Selection based on Worker Trust Vector (WTV)

In order to evaluate a worker's global trust, we calculate the SOTs between the worker and *authenticated requesters*, *active requesters*, and *ordinary requesters*, respectively. Based on it, we devise a worker trust vector to present each worker's trust in three trust metrics: *deterministic trust*, *non-deterministic trust*, and *ordinary trust*, respectively.



### 4.2.3.1 Three Types of Crowdsourcing Requesters

- **Authenticated Requester:** An authenticated requester is one who is manually verified by a crowdsourcing platform and marked as a trustworthy requester.
- **Active Requester:** An active requester is one who has a good reputation on a crowdsourcing platform. This type of requesters is not manually verified by crowdsourcing platforms.
- **Ordinary Requester:** An ordinary requester is one who has no remarkable characteristics. This type of requesters has published a limited number of tasks. Ordinary requesters can be randomly selected from requesters who have ordinary reputations.

### 4.2.3.2 Trust Elements in a WTV

Based on the three types of requesters, a worker trust vector is defined as  $WTV_w = \{DeT_w, NDeT_w, OT_w\}$ , where  $DeT_w$ ,  $NDeT_w$ ,  $OT_w$  are *deterministic trust*, *non-deterministic trust*, and *ordinary trust*, respectively.

- **Deterministic Trust (DeT):** A worker's deterministic trust is an aggregation of the SOTs between the worker and a group of authenticated requesters. This type of trust reveals whether a worker is regularly trusted by authenticated requesters. As an authenticated requester is a trustworthy entity, a worker's trust relations with a group of authenticated requesters is a deterministic trust indicator,

$$DeT_{w_i} = \frac{\sum_{r \in R_{aut}} SOT_{(r, w_i)}}{|R_{aut}|}, \quad (4.8)$$

where  $R_{aut}$  is the set of the authenticated requesters and  $|R_{aut}|$  is the number of the requesters in  $R_{aut}$ .

- **Non-Deterministic Trust (NDeT):** A worker's non-deterministic trust is calculated by aggregating the SOTs between the worker and a group of randomly

picked active requesters. This type of trust is a non-deterministic trust indicator as a worker may increase SOT values with a small part of active requesters by colluding with its accomplices,

$$NDeT_{w_i} = \frac{\sum_{r \in R_{act}} SOT_{(r, w_i)}}{|R_{act}|}, \quad (4.9)$$

where  $R_{act}$  is the set of the active requesters and  $|R_{act}|$  is the number of the requesters in  $R_{act}$ .

- **Ordinary Trust (OT):** A worker's ordinary trust is calculated by integrating the SOTs between the workers and a group of ordinary requesters. This trust factor reflects a worker's trust relations with ordinary requesters.

$$OT_{w_i} = \frac{\sum_{r \in R_{ord}} SOT_{(r, w_i)}}{|R_{ord}|}, \quad (4.10)$$

where  $R_{ord}$  is the set of the randomly selected ordinary requesters and  $|R_{ord}|$  is the number of the requesters in  $R_{ord}$ .

#### 4.2.3.3 WTV based Worker Selection

Based on the calculated WTVs of workers, we can select those workers whose DeT, NDeT and OT values are all above average level as honest workers. The details of this approach are presented in Algorithm 2.

#### 4.2.3.4 Summary

A WTV includes a worker's trust relations with authenticated requesters, active requesters and ordinary requesters, respectively. As a spam worker can only increase its SOTs between himself/herself and its colluding accomplices, it is tough for him to improve the trust values between itself and different groups of requesters in a WTV. Thus, CrowdDefense can effectively prevent spam workers from participating in tasks by selecting those workers who have trust values above average levels in WTVs.

---

**Algorithm 2:** The Worker Selection Algorithm

---

**Data:**  $W, CTN$ **Result:**  $SW$ 

```

1 for  $w \in W$  do
2    $\lfloor$  Compute  $WTV_{s_w}$  via Eq. 4.8, Eq. (4.9) and Eq. (4.10);
3 Compute  $avg(DeT_W)$ ,  $avg(NDeT_w)$  and  $avg(OT_W)$ ;
4 for  $w \in W$  do
5    $\lfloor$  if  $DeT_w \geq avg(DeT_W) \& NDeT_w \geq avg(NDeT_w) \& OT_w \geq avg(OT_W)$ 
6      $\lfloor$  then
7        $\lfloor$  Add  $w$  to  $SW$ ;
7 Return  $SW$ ;
```

---

## 4.3 Experiments and Analysis

In the experiments, we evaluate our proposed CrowdDefense, and compare it with CrowdTrust, H2010e [153], and AMT, respectively.

- **CrowdTrust:** A context-aware trust model that considers both task types of tasks and the reward amount of tasks in worker selection, which is presented in detail in Chapter 3.
- **H2010e:** A trust model that considers both long-term trust and short-term trust in worker selection [153].
- **AMT:** An answer overall approval rate based trust model that is applied in Amazon Mechanic Turk [110].

### 4.3.1 Data Preparation

To evaluate our proposed CrowdDefense, we need a dataset containing trust network in crowdsourcing. In a real who-trust-whom dataset *soc-sign-epinions*<sup>1</sup>, a reviewer (like a requester) trusts another participant (like a worker) if the reviewer adopts the participant’s reviews (answers) on a product or a movie (like a crowdsourcing task).

In contrast to other real network datasets, e.g., *email-Enron*<sup>2</sup> and *ego-Twitter*<sup>3</sup>, the *soc-sign-epinions* dataset well fits the structure of our proposed CTN. Thus, we adopt the 841372 edges (trust relations) among 131828 nodes (participants) in *soc-sign-epinions*<sup>1</sup> dataset to construct our proposed CTN.

As there is no available dataset including spam workers with Threat Patterns A, B, and C, we generate spam workers and their accomplices, and connect them to the participants in the *soc-sign-epinions* dataset according to the characteristics of the three threat patterns. Considering spam workers may obtain guises **G1** and **G2** via colluding in shadow tasks belonging to any task type, all workers' answer approval rates are randomly generated in the range between 85% and 95% in different contexts.

#### 4.3.1.1 Experiment 1

In this experiment, we evaluate the effectiveness of CrowdDefense in selecting honest workers when the number of accomplices increases. Targeting Threat Patterns A, B, and C, 60 spam workers are generated while the number of accomplices changes. In particular, the numbers of each type of accomplices are set as 30, 60, and 90, respectively. (see Table 4.1). In total, 9 groups of spam workers are generated (3 threat patterns \* 3 different numbers of colluding participants). In this experiment, each group of spam workers and 240 honest workers apply for the participation in a task.

#### 4.3.1.2 Experiment 2

In this experiment, we evaluate the effectiveness of CrowdDefense in selecting honest workers when the number of spam workers increases. Among the workers who apply for tasks, the number of honest workers is set as 240 while the number of spam workers increases from 60 to 300 with a step of 60 (see Table 4.2).

---

<sup>2</sup><http://www.cs.cmu.edu/enron>

<sup>3</sup><https://snap.stanford.edu/data/egonets-Twitter.html>

**Table 4.1:** The Dishonest Participants in Experiment1

<b>Participants</b> <b>Threat Pattern</b>	# Spam Workers	(# Spam Requesters, # Grey Requesters, # Grey Workers)
Threat Pattern A	60	(30,0,0), (60,0,0), (90,0,0)
Threat Pattern B	60	(30,30,0), (60,60,0), (90,90,0)
Threat Pattern C	60	(30,60,60), (60,60,60), (90,90,90)

**Table 4.2:** The Dishonest Participants in Experiment2

<b>Participant</b> <b>Threat Pattern</b>	# Spam Workers	# Spam Requesters	# Grey Requesters	# Grey Workers
Threat Pattern A	60,120,180,240,300	60	-	-
Threat Pattern B	60,120,180,240,300	60	60	-
Threat Pattern C	60,120,180,240,300	60	60	60

## 4.3.2 Experiment Results

### 4.3.2.1 Experiment 1 (Effectiveness Comparison 1)

This experiment is to investigate if CrowdDefense can effectively select honest workers when spam workers collude with a different number of accomplices.

**Results:** Figs. 4.5-4.7 show the worker selection results of different methods when the numbers of accomplices increase. From the figures, we can observe that (1) In the baselines, the percentages of honest workers and spam workers are close to the default percentages of honest workers and spam workers (i.e., 80% and 20%, respectively); (2) CrowdDefense is always the best model in terms of selecting honest workers in all the cases. Among the workers selected by CrowdDefense, on average, the honest workers account for the highest percentage 95.7% and the spam workers account for the lowest percentage 4.21%. On average, the percentage of honest workers in CrowdDefense is 16.08% higher than that in the second best method, and the percentage of spam workers in CrowdDefense is 76.03% lower than that in the second best method; (3) Among all the workers selected by CrowdDefense, the honest workers always account for at least between 93.3% and 99.2%.

**Analysis:** The experimental results illustrate that (1) Baselines take spam workers as honest workers in a task as they do not consider that spam workers may have guises **G1** (i.e., good reputations) and guises **G2** (i.e., many attack edges); (2) A spam worker

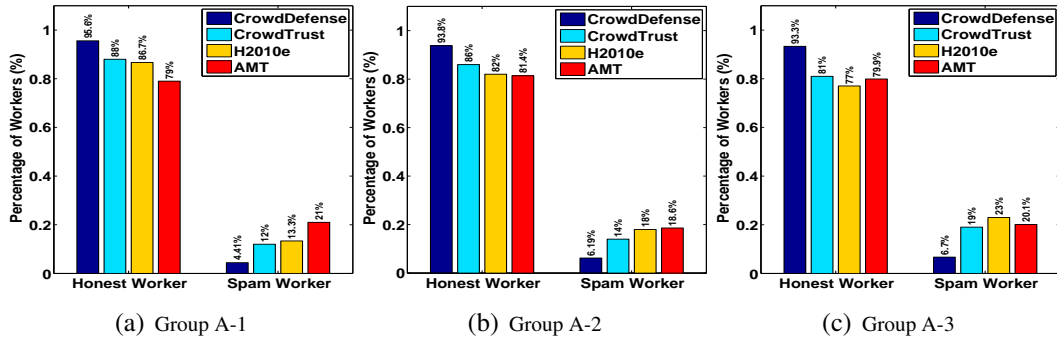


Figure 4.5: The Comparison of Different Methods in Threat Pattern A

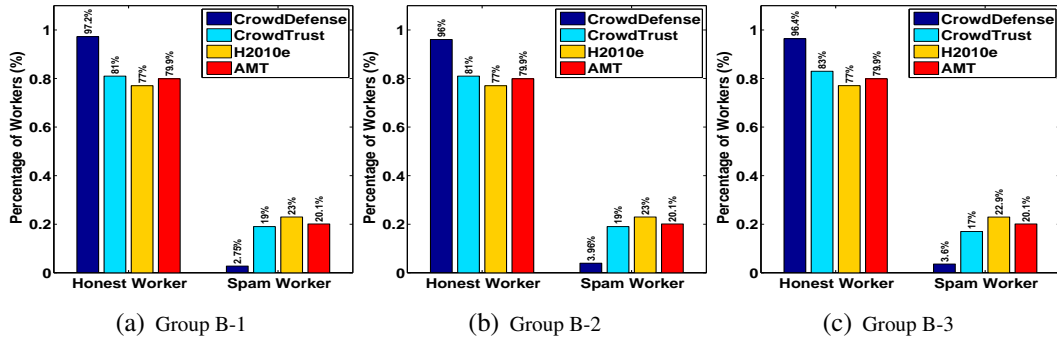


Figure 4.6: The Comparison of Different Methods in Threat Pattern B

connects to different requesters via the paths that contain edges with low direct trust values, which cannot be counterfeited. Thus, the trust values in a spam worker’s WTV are low. CrowdDefense selects those workers with high trust values in their WTVs so as to prevent spam workers from participating in the tasks; (3) When the numbers of a spam worker’s colluding accomplices increase, both the trustworthy paths involving them with high direct trust values and the untrustworthy paths involving them with low direct trust values are generated in a CTN. Thus, the trust values in the spam worker’s WTV do not increase. As such, CrowdDefense maintains its high effectiveness.

#### 4.3.2.2 Experiment 2 (Effectiveness Comparison 2)

This experiment investigates if CrowdDefense is effective in selecting honest workers when the number of spam workers in a CTN increases.

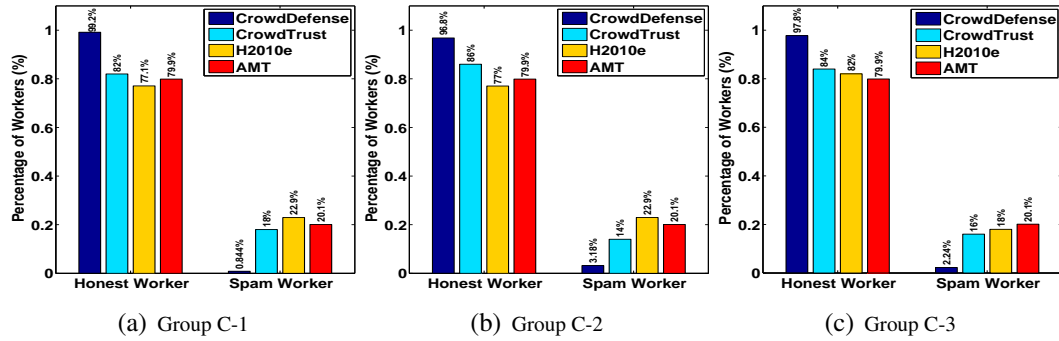


Figure 4.7: The Comparison of Different Methods in Threat Pattern C

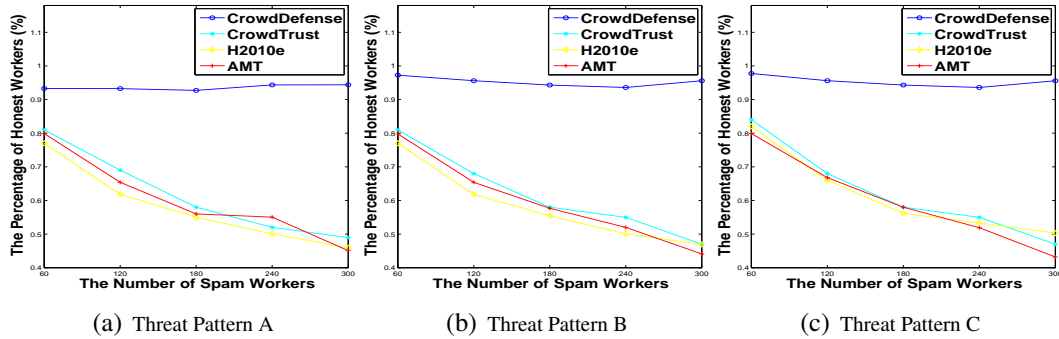


Figure 4.8: The Comparison of Different Methods with Different Numbers of Spam Workers

**Results:** From Figs. 4.8(a)-4.8(c), we can observe that the baselines select less honest workers when the number of spam workers increases from 60 to 300 with a step of 60. Targeting Threat Patterns A, B, and C, among the workers selected by the second best method, the average percentage of honest workers declines from 82.02% to 47.67%. By contrast, CrowdDefense maintains high effectiveness in selecting honest workers. The average percentages of honest workers in the workers selected by CrowdDefense are as high as 93.61%, 95.25%, and 95.37%, respectively.

**Analysis:** The baselines do not consider that spam workers boost reputations with their accomplices, i.e., guise **G1**. Thus, when the number of spam workers increases, baselines take more spam workers as honest workers and select them to participate in tasks. By contrast, CrowdDefense employs a WTV to measure a worker’s trust levels from the views of authenticated requesters, active requesters, and ordinary requesters,

---

respectively. A worker's WTV incorporates all trust traces in the trust paths involving the worker and different requesters. As the paths cannot be counterfeited, no matter how many spam workers exist, none of them can increase the trust values in their WTVs. CrowdDefense only selects the workers whose trust values are all above the average levels, thus it can filter out most of the spam workers.

**Summary:** The baselines fail to accurately indicate the trust levels of spam workers who boost reputations with their accomplices. Thus, when the number of spam workers increases, baselines select more of them to participate in tasks. By contrast, CrowdDefense maintains high effectiveness in selecting honest workers although the numbers of accomplices increase and the number of spam workers increases. Therefore, CrowdDefense significantly outperforms the three state-of-the-art approaches in defending the threats from spam workers by effectively selecting 93.3% ~ 99.2% honest workers.

## 4.4 Conclusion

In this chapter, we have presented a novel trust vector-based spam worker defense model CrowdDefense. In CrowdDefense, we have analyzed three threat patterns and proposed a new trust vector to evaluate a worker's trustworthiness. The results of experiments have demonstrated that CrowdDefense significantly outperforms the context-aware worker selection model CrowdTrust, and two state-of-the-art approaches H2010e and AMT in selecting honest workers when spam workers with guise **G1** and **G2** exist. To sum up, CrowdDefense applies the average trust values of all workers' WTV as the thresholds for defending spam workers with three threat patterns. This model provides a way to select honest workers and prohibit spam workers from participating in tasks. However, it cannot precisely identify who is exactly a spam workers among the prohibited workers. Thus, in the following chapter, we propose a learning-based spam worker identification model that can exactly identify a worker's real identity.



## Chapter 5

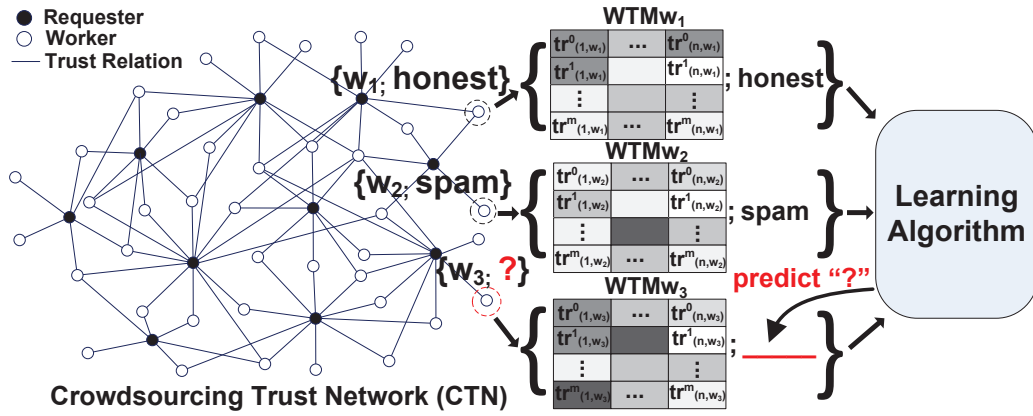
---

# Spam Worker Identification based on Trust Matrix

---

In Chapter 4, our proposed trust vector-based spam worker defense model CrowdDefense has shown its effectiveness in defending against spam workers with counterfeited “good” reputations (i.e., guise **G1**) and trust connections (i.e., guise **G2**) to honest requesters via collusions. In fact, CrowdDefense prevents those high-risk workers from participating in tasks by taking the average trust levels of all workers as the thresholds. In this chapter, we further propose a model that can accurately identify a worker’s identity and thus can effectively prohibit spam workers from participating in tasks. In this model, we first represent each worker by a trust network-based feature set called Worker Trust Matrix (WTM). Based on the WTMs, each worker’s identity is then predicted (see Fig. 5.1 ). To the best of our knowledge, this is the first trust network-based spam worker identification model in crowdsourcing environments.

In particular, we propose a new trust metric called trust trace. A trust trace measures the extent to which a worker is trusted by a requester in a fixed-hop sub-CTN starting from the requester. We then devise a novel worker trust representation called Worker Trust Matrix (WTM). A worker’s WTM contains the trust traces of between the worker and all requesters and thus is a global trust network-based feature set of the worker. We further prove that a WTM cannot be manipulated by any worker and contains usable information for identifying a worker’s identity. Both the un-manipulable property and the usable property of a WTM are critical for effectively identifying s-



**Figure 5.1:** A Worker Trust Matrix (WTM) encapsulates the global trust network-based features of the worker and can be exploited by learning algorithms for further prediction.

pam workers. The results of extensive experiments over four datasets demonstrate the superior effectiveness of our proposed spam worker identification model.

## 5.1 Worker Trust Matrix (WTM) for Spam Worker Identification

### 5.1.1 Spam Worker Identification Problem Formulation

Recall the CTN formulation in Section 4.1.1.4, we can obtain a Crowdsourcing trust network  $CTN(R \cup W, TE \cup DTE)$ , where  $R = \{r_i\}_{i=1}^{|R|}$  denote all the requester nodes,  $W = \{w_j\}_{j=1}^{|W|}$  denote all the worker nodes,  $TE = \{te_k\}_{k=1}^{|TE|}$  denote all the trust edges containing direct trust values, and  $DTE = \{dte_k\}_{k=1}^{|DTE|}$  denote all the distrust edges containing direct trust values. Note that, in a CTN, there is no edge between any two requesters because they cannot transact with each other. Likewise, there is no edge between any two workers. Therefore, one hop in a path in a CTN corresponds to two intermediate nodes consisting of one worker and one requester, which is different from the concept of one hop in social networks.

Based on the CTN, we formulate the spam worker identification as below,

- **Input:** (1) A  $CTN(R \cup W, TE \cup DTE)$ ; (2) A small worker set  $U$  where workers' identities have been manually verified; (3) A large worker set  $V$  where workers' identities need to be predicted.
- **Output:** (1) Each worker's trust representation; (2) The predicted identities of the workers in  $V$ .

## 5.1.2 Worker Trust Matrix (WTM)

In this section, we present the components and the properties of a novel worker trust representation called Worker Trust Matrix (WTM) that contains critical information for the identification of a worker's identity. Based on the definition of trustworthy path (Definition 4.2.1) and the definition of untrustworthy path (Definition 4.2.1), we first calculate two types of trust: positive trust and trust penalty. By using the two types of trust, we calculate the atomic trust relations between a requester and a worker in different sub-CTNs. Furthermore, we devise a trust trace-based matrix to represent a worker in the context of trust, which contains un-manipulable and usable properties for effectively spam worker identification.

### 5.1.2.1 Positive Trust Indicator

In a  $k$ -hop sub-CTN starting from requester  $r_i$ , there may exist several  $k$ -hop trustworthy paths that end at worker  $w_j$ . In these trustworthy paths, any two directly connected nodes trust each other. Thus, we apply the sum of all the direct trust values in the trustworthy paths between requester  $r_i$  and worker  $w_j$  as a *positive trust indicator* of the worker  $w_j$  in the  $k$ -hop sub-CTN starting from the requester  $r_i$ , which is calculated by Eq. (5.1):

$$positrust_{(r_i, w_j)}^k = \sum_{tp_l^k \in TP_{(r_i, w_j)}^k} \sum_{dt_u \in DT^{tp_l^k}} dt_u, \quad (5.1)$$

where,  $TP^k(r_i, w_j) = \{tp_l^k\}_{l=1}^{|TP^k(r_i, w_j)|}$  denotes all the  $k$ -hop trustworthy paths between  $r_i$  and  $w_j$ , and  $DT^{tp_l^k} = \{dt_u\}_{u=1}^{|DT^{tp_l^k}|}$  denotes all the direct trust values in  $tp_l^k$ .

Note that, if  $TP^k(r_i, w_j) = \emptyset$ ,  $positrust_{(r_i, w_j)}^k = 0$ .

### 5.1.2.2 Trust Penalty

In a  $k$ -hop sub-CTN starting from requester  $r_i$ , there may exist several  $k$ -hop untrustworthy paths that end at worker  $w_j$ . In these untrustworthy paths, as only the worker  $w_j$  is distrusted, we apply the sum of all the direct trust values in the trust edges and distrust degrees in distrust edges in the paths as a *trust penalty* on the worker  $w_j$  in the  $k$ -hop sub-CTN, which is calculated by Eq. (5.2):

$$penalty_{(r_i, w_j)}^k = \sum_{utp_h^k \in UTP_{(r_i, w_j)}^k} \sum_{dt_v \in DT^{utp_h^k}} dt_v + (1 - dt_e^{utp_h^k}), \quad (5.2)$$

where,  $UTP_{(r_i, w_j)}^k = \{utp_h^k\}_{h=1}^{|UTP_{(r_i, w_j)}^k|}$  denotes all the  $k$ -hop untrustworthy paths between  $r_i$  and  $w_j$ ,  $DT^{utp_h^k} = \{dt_v\}_{v=1}^{|DT^{utp_h^k}|}$  denotes all the direct trust values in the trust edges in  $utp_h^k$ , and  $dt_e^{utp_h^k}$  denotes the direct trust value in the distrust edge in  $utp_h^k$ . Note that, if  $UTP_{(r_i, w_j)}^k = \emptyset$ ,  $penalty_{(r_i, w_j)}^k = 0$ .

### 5.1.2.3 Trust Trace

A trust trace  $tr_{(r_i, w_j)}^k$  aggregates both the positive trust indicator and the trust penalty of a worker  $w_j$  in a  $k$ -hop sub-CTN starting from a requester  $r_i$  to measure the extent to which the worker  $w_j$  is trusted by the requester  $r_i$  in the sub-CTN, which is calculated by Eq. (5.3):

$$tr_{(r_i, w_j)}^k = \frac{positrust_{(r_i, w_j)}^k - penalty_{(r_i, w_j)}^k}{\sum_{DT^{tp_p^k} \in TP_{r_i}^k} \sum_{dt_q \in tp_p^k} dt_q}, \quad (5.3)$$

where,  $TP_{r_i}^k$  denotes all the  $k$ -hop trustworthy paths that start from a requester  $r_i$  and end at any worker. The sum of all direct trust values in  $TP_{r_i}^k$  is applied as the denominator in Eq. (5.3) as it is the total trust information given by  $r_i$  in the  $k$ -hop sub-CTN. Essentially,  $tr_{(r_i, w_j)}^k$  leverages the frequencies by which the worker  $w_j$  appears in the

$k$ -hop trustworthy paths and the  $k$ -hop untrustworthy paths starting from  $r_i$  for measuring the extent to which the worker  $w_j$  is trusted by the requester  $r_i$  in the  $k$ -hop sub-CTN.

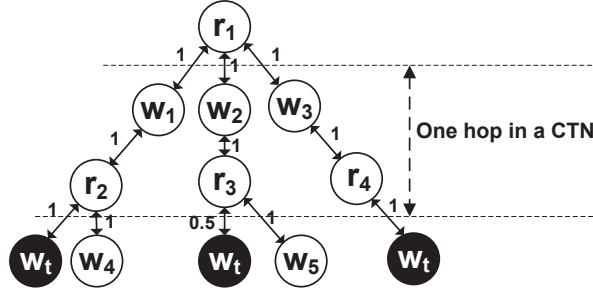


Figure 5.2: An Example of One-hop Sub-CTN

### 5.1.2.4 An Example of Trust Trace Calculation

Based on the 1-hop sub-CTN in Fig. 5.2, we present an example of the calculation of  $tr_{(r_1, w_t)}^1$ . We use  $\varepsilon = 0.9$  to clarify the calculation in this example. First, the total positive trust indicator is calculated based on the direct trust values in two trustworthy paths  $tp_{1(r_1, w_t)}: r_1 \xleftarrow{dt=1} w_1 \xleftarrow{dt=1} r_2 \xleftarrow{dt=1} w_t$  and  $tp_{2(r_1, w_t)}: r_1 \xleftarrow{dt=1} w_3 \xleftarrow{dt=1} r_4 \xleftarrow{dt=1} w_t$ . In particular, the total positive trust indicator is calculated by  $\sum_{i=1}^2 (1 + 1 + 1) = 6$ . Second, the trust penalty is calculated based on the direct trust values in untrustworthy path  $utp_{1(r_1, w_t)}: r_1 \xleftarrow{dt=1} w_2 \xleftarrow{dt=1} r_3 \xleftarrow{dt=0.5} w_t$ , i.e.,  $1 + 1 + (1 - 0.5) = 2.5$ . Thirdly, there are four 1-hop trustworthy paths that start from  $r_s$ , thus the denominator in Eq. (1) is calculated by  $\sum_{i=1}^4 3 = 12$ . Accordingly,  $tr_{(r_1, w_t)}^1 = \frac{6-2.5}{12} = 0.2917$ .

### 5.1.2.5 Worker Trust Matrix (WTM)

Given a worker, we can compute the trust traces between the worker and all the requesters in sub-CTNs with different hops to obtain a global trust feature set for representing the worker, i.e., WTM. For example, in Fig. 5.3, given a worker  $w_t$ , we can first compute the trust traces of  $w_t$  in the sub-CTNs that start from  $r_1$  and end with 0 to  $m$  hops, i.e.,  $TR_{(r_1, w_t)} = \{tr_{(r_1, w_t)}^0, tr_{(r_1, w_t)}^1, \dots, tr_{(r_1, w_t)}^m\}$ . Likewise, the TRs be-

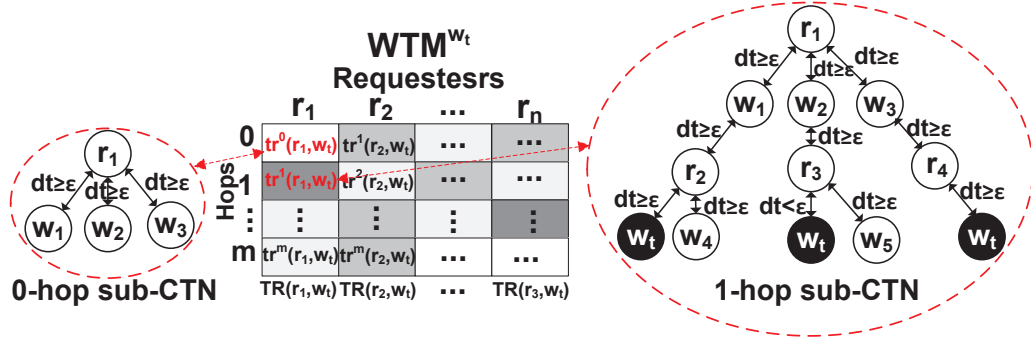


Figure 5.3: An Example of a WTM for a Worker

tween  $w_t$  and other requesters can be computed. Taking each  $TR$  of  $w_t$  as one column of  $WTM_{w_t}$ , we obtain  $WTM_{w_t} = \{TR_{(r_1, w_t)}, TR_{(r_2, w_t)}, \dots, TR_{(r_n, w_t)}\}$ . Below, we prove the un-manipulable property and usable property of a WTM that are critical for identifying a worker's identity.

**Theorem 5.1.** *Let  $d$  denote the minimum number of requesters with whom a spam worker collude in a CTN and  $e$  denote the minimum number of requesters by whom an honest worker is directly trusted. Given an honest worker and a spam worker, let  $pr$  denote the the probability that the spam worker can manipulate its WTM to be the same as that of the honest worker. We can obtain that  $pr \ll \frac{1}{2^{(d+e)}}$ .*

*Proof.* A shadow task is commonly preset to be difficult for general workers to solve, e.g., a jigsaw puzzle task, thus an honest worker who does not know the preset answer can hardly succeed in such a shadow task by the answer submission deadline. As such, the probability  $p_{hw}$  that an honest worker  $hw$  succeeds in a shadow task is below  $\frac{1}{2}$ , i.e.,  $p_{hw} < \frac{1}{2}$ . In a CTN, we can reasonably assume that there are more honest requesters than colluding requesters. Let  $h$  denote the total number of honest requesters and  $c$  denote the total number of colluding requesters in a CTN, we can obtain that  $h > c$ . Given a spam worker  $sw$  who has colluded with  $g$  requesters in shadow tasks and thus possess trust edges to the colluding requesters, the probability that an honest worker also possesses trust edges to  $g$  colluding requesters is  $pr^1 = \frac{\binom{c}{g}}{\binom{h+c}{g}} * p_{hw}^g = \frac{c!(h+c-g)!}{(c-g)!(h+c)!} * p_{hw}^g$ . As  $g \geq d$ ,  $p_{hw} < \frac{1}{2}$ , and  $h > c$ , we can obtain  $pr^1 \ll \frac{1}{2^d}$ . If a spam

worker wants to manipulate its *WTM* to be the same as that of an honest worker, two necessary conditions must be satisfied. First, the spam worker can let the honest worker directly connects with  $g$  colluding requesters via trust edges. Second, the spam worker can simultaneously possess the same number of trust edges that exist between the honest worker and honest requesters. As we have proved above, the probability that the first condition is satisfied is  $pr^1 \ll \frac{1}{2^d}$ . Let  $pr^2$  denote the probability that the second condition is satisfied, likewise, we can obtain that  $pr^2 \ll \frac{1}{2^e}$ . Based on it, we can obtain that  $pr = pr^1 * pr^2 \ll \frac{1}{2^{(d+e)}}$ .  $\square$

As a small number of colluding users may form a clique that can be easily detected [108], the number of colluding requesters  $d$  in a CTN is commonly a big positive integer. Suppose  $d + e = 10$ , we have that  $pr \ll 0.000977$ . In fact, the number of  $d + e$  in a CTN is commonly larger than 10. Therefore, based on the inequality proved in 5.1, i.e.,  $pr \ll \frac{1}{2^{(d+e)}}$ , we can conclude that the value of  $pr$  is almost 0 in a CTN. Due to the extremely low probability that a spam worker can manipulate its *WTM* to be same with that of an honest worker, we suggest that a *WTM* possesses the *un-manipulable* property.

Below, we further prove that a *WTM* has the *usable* property which is critical for identifying spam workers.

**Theorem 5.2.** *Given honest worker set  $HW = \{hw_i\}_{i=1}^{|HW|}$ , grey worker set  $GW = \{gw_j\}_{j=1}^{|GW|}$ , and spam worker set  $SW = \{sw_k\}_{k=1}^{|SW|}$ , for any  $WTM_{hw_i}$ ,  $WTM_{gw_j}$ , and  $WTM_{sw_k}$ , there exists a function  $\phi(\cdot)$  that satisfies  $\phi(WTM_{hw_i}) \neq \phi(WTM_{gw_j})$ ,  $\phi(WTM_{gw_j}) \neq \phi(WTM_{sw_k})$ , and  $\phi(WTM_{hw_i}) \neq \phi(WTM_{sw_k})$ .*

*Proof.* Let  $\mathcal{F}$  denote the distribution function of the trust trace sets *TRs* between a worker  $w_t$  and all the honest requesters  $HR = \{hr_l\}_{l=1}^{|HR|}$ ,  $\mathcal{G}$  denote the distribution function of *TRs* between  $w_t$  and all the grey requesters  $GR = \{gr_p\}_{p=1}^{|GR|}$ , and  $\mathcal{S}$  denote the distribution function of *TRs* between  $w_t$  and all the spam requesters  $SR = \{sr_u\}_{u=1}^{|SR|}$ . As the values of the trust traces in  $WTM_{w_t}$  depend on to whom  $w_t$  links via trust edges, the probabilities  $p_{(hr,w_t)}$ ,  $p_{(gr,w_t)}$ , and  $p_{(sr,w_t)}$  that  $w_t$  has a

trust edge to an honest requester, a grey requester and a spam requester are the latent parameters of  $\mathcal{F}$ ,  $\mathcal{G}$ , and  $\mathcal{S}$ , respectively. Based on it, we obtain that  $WTM_{hw_i} = \{TR_{(hr,hw_i)} \sim \mathcal{F}(p_{(hr,hw_i)}), TR_{(gr,hw_i)} \sim \mathcal{G}(p_{(gr,hw_i)}), TR_{(sr,hw_i)} \sim \mathcal{S}(p_{(sr,hw_i)})\}$ ,  $WTM_{gw_j} = \{TR_{(hr,gw_j)} \sim \mathcal{F}(p_{(hr,gw_j)}), TR_{(gr,gw_j)} \sim \mathcal{G}(p_{(gr,gw_j)}), TR_{(sr,gw_j)} \sim \mathcal{S}(p_{(sr,gw_j)})\}$ , and  $WTM_{sw_k} = \{TR_{(hr,sw_k)} \sim \mathcal{F}(p_{(hr,sw_k)}), TR_{(gr,sw_k)} \sim \mathcal{G}(p_{(gr,sw_k)}), TR_{(sr,sw_k)} \sim \mathcal{S}(p_{(sr,sw_k)})\}$ , respectively. As a worker's identity determines its transaction behaviours, and vice versa, the probabilities that a worker can obtain a trust edge to an *hr*, a *gr* or an *sr* satisfy  $p_{(hr,hw_i)} > p_{(hr,gw_j)} > p_{(hr,sw_k)}$ ,  $p_{(gr,hw_i)} < p_{(gr,gw_j)} < p_{(gr,sw_k)}$ , and  $p_{(sr,hw_i)} < p_{(sr,gw_j)} < p_{(sr,sw_k)}$ . Thus, there must exist a function  $\phi(\cdot)$  that satisfies  $\phi(WTM_{hw_i}) \neq \phi(WTM_{gw_j})$ ,  $\phi(WTM_{gw_j}) \neq \phi(WTM_{sw_k})$ , and  $\phi(WTM_{hw_i}) \neq \phi(WTM_{sw_k})$ .  $\square$

In Theorem 5.2, we have proved the inequality relations between any two workers with different identities. The inequality relations actually demonstrates the existence of the boundaries for classifying workers into the correct identity groups. Thus, we conclude that WTMs contain usable information that is critical for effectively identifying spam workers.

## 5.2 WTM-based Spam Worker Identification Model

A WTM may contain latent spatial features because each column in a WTM consisting of the trust traces derived from sub-CTNs with different hops starting from an identical requester. Considering this, in our model, we modify a convolutional neural network to learn workers' WTMs for classifying workers to the correct identity groups. In particular, our proposed spam worker identification model consists of a random walk-based WTMs estimation algorithm and a modified six-layer convolutional neural network CLnet-6.



## 5.2.1 WTM Estimation Algorithm

### 5.2.1.1 Random Walk Estimation

Ideally,  $WTM_{w_t \in W}$  can be precisely calculated by traversing all the paths in a CTN. However, the complexity of a traversal algorithm exponentially increases with the degree of a node in a CTN. Thus, we devise an improved random walk-based algorithm with a lower complexity to estimate  $WTMs$ . In particular, in each random walk round, we update the trust traces of all the workers in a randomly searched path. The process terminates when the number of searching times exceeds the maximum round number  $rod_{max}$  or the change of  $WTMs$  converges. The algorithm's worst time complexity is  $O(rod_{max} * hop_{max} * de)$ , where  $hop_{max}$  is the maximum number of hops in a searching path, and  $de$  is the maximum degree of a node.

### 5.2.1.2 Dimension Reduction

In any WTM, the number of columns equals the total number of requesters in a CTN, which is too big to achieve effective learning. Thus, in each WTM, we replace a requester node with a requester group to reduce the dimensions. For example,  $WTM_{w_t} = \{TR(r_1, w_t), TR(r_2, w_t), \dots, TR(r_n, w_t)\}$  is transformed to a low-dimensional  $WTM'_{w_t} = \{TR(grp_1, w_t), TR(grp_2, w_t), \dots, TR(grp_{\frac{n}{\alpha}}, w_t)\}$ . Here,  $\alpha$  is a preset value that determines the number of requesters in each group and satisfies  $\frac{n}{\alpha} \in \mathbb{Z}$ . In addition,  $TR(grp_p, w_t)$  is calculated by  $TR(grp_p, w_t) = \frac{\sum_{r_q \in grp_p} TR(r_q, w_t)}{\alpha}$ . As requesters are randomly grouped, the un-manipulable property of  $WTM_{w_t}$  can be inherited by  $WTM'_{w_t}$ . In addition, after the random grouping, we obtain three types of requester groups: *Honest-dominated*, *Grey-dominated*, and *Spam-dominated*, respectively. By replacing the concept of an honest requester in Theorem 2 with the concept of an honest-dominated requester group, and so on, for a grey requester and a spam requester, Theorem 2 still holds on  $WTM'$ s. Thus,  $WTM'_{w_t}$  is also usable for identifying the identity of  $w_t$ .

Algorithm 3 presents the combination of *random walk estimation* and *dimension*

**Algorithm 3:** The WTM Estimation Algorithm

---

**Data:**  $CTN(R \cup W, TE \cup DTE)$ , the max # searching round  $rod_{max}$ , the max # hops in a searching path  $hop_{max}$ , the # of rows in a WTM  $m$ , the # of requesters in a group  $\alpha$ ;

**Result:**  $WTM'_{w_t \in W}$

- 1 **Initiate**  $WTM^p_{w_t \in W} = WTM_{w_t \in W} = \mathcal{O}^{m \times |R|}$ ;
- 2 **for** ( $rod = 1$  to  $rod_{max}$ ) **do**
- 3     Select a random trustworthy path  $rtp$  with  $hop_{max}$ ;
- 4     Slide a window of hops  $m$  over the  $rtp$  to obtain  $RTP = \{rtp'_l\}_{l=1}^{hop_{max}+1-m}$ ;
- 5     **for** ( $l = 1$  to  $(hop_{max} + 1 - m)$ ) **do**
- 6         Update  $TR_{(r_i \in rtp_l, w_t \in W)}$  in  $WTM_{w_t \in W}$
- 7     **if**  $\|WTM_{w_t \in W} - WTM^p_{w_t \in W}\|_F \leq \epsilon$  **or**  $rod == rod_{max}$  **then**
- 8          $WTM'_{w_t \in W} = DimensionReduction(WTM_{w_t \in W})$ ;
- 9         **Return**  $WTM'_{w_t \in W}$ ;

---

*reduction* for extracting low-dimensional  $WTM'$ 's.

## 5.2.2 Learning Algorithm CLnet-6

CLnet-6 is a six-layer neural network that learns  $WTM'_{w_j \in U}$  with known identities for predicting identities of workers from  $V$ . In particular, CLnet-6 consists of one standardized layer  $ST$ , two convolutional layers  $C_1$  and  $C_2$ , two sub-sampling layers  $S_1$  and  $S_2$ , and a fully linked Multi-layer Perceptron  $MP$ . Below, we present the rationale and components of CLnet-6.

### 5.2.2.1 Probabilistic Classifier.

Let  $WE$  denote the weight matrix,  $b$  denote the bias vector,  $WI = \{HW, GW, SW\}$  denote the worker identity set, and  $\otimes$  denote the operations in CLnet-6. Given a  $WTM'_{w_k}$ , the probability that  $w_k \in WI_i$  is defined as a stochastic variable  $P(Y = WI_i | WTM'_{w_k}, WE, b) = \frac{e^{\otimes WTM'_{w_k} * WE_i + b_i}}{e^{\sum_j \otimes WTM'_{w_k} * WE_j + b_j}}$ .

### 5.2.2.2 Loss Function.

Let  $U$  denote a worker set where each worker's identity is known. As there are three types of worker identities (i.e., *honest worker*, *grey worker* and *spam worker*), we define a vector  $ID$  to represent a worker's identity. For example,  $ID_{w_k} = (1, 0, 0)$  represents  $w_k$  belongs to the first type of identity: honest worker. A training sample  $Sa$  consists of a WTM and an identity vector  $ID$ , i.e.,  $Sa^{w_k} = \{WTM_{w_k}, ID^{w_k}\}$ . Given  $WTM'_{w_i \in U}$ , the log-likelihood is calculated as  $L(W, b, WTM'_{w_i \in U}) = \sum_{i=1}^{|U|} \log(P(Y = WI^{(w_i)} | WTM'_{w_i}, WE, b))$ . In CLnet-6, we adopt the negative log-likelihood  $l(W, b, WTM'_{w_i \in U}) = -L(W, b, WTM'_{w_i \in U})$  as the loss. As maximizing the likelihood is equivalent to minimizing the loss, we train CLnet-6 by minimizing  $l$ .

### 5.2.2.3 Training Operations.

The convolution operation is performed between the input layer  $I$  and the first convolutional layer  $C_1$ , and between the first sub-sampling layer  $S_1$  and the second convolutional layer  $C_2$ . The sub-sampling operation is performed between layer  $C_1$  and layer  $S_1$  and between the second convolutional layer  $C_2$  and the second sub-sampling layer  $S_2$ .

(1) *Standardization*: this operation is performed on the  $ST$  layer. Let  $avg(WTM')$  denote the average trust trace values in all  $WTMs$ ,  $min(WTM')$  denote the minimal trust trace value in all  $WTMs$ , and  $max(WTM')$  denote the maximal trust trace value in all  $WTMs$ , we normalize each element in  $WTM'_{w_k \in W}$  as:  $WTM'_{w_k}(i, j) = \frac{WTM'_{w_k}(i, j) + avg(WTM') - min(WTM')}{max(WTM') + avg(WTM') - min(WTM')}$ .

(2) *Convolution*: this operation is performed between  $ST$  layer and  $C_1$  layer, and between  $S_1$  layer and  $C_2$  layer by repeating a filter function across all feature maps. The filter function has a receptive field with a fixed size  $\beta * \gamma$  and the parameters including a weight matrix  $WE_c \in \mathbb{R}^{\beta * \gamma}$  and a bias  $b_c$ . In particular, for each  $\beta * \gamma$  area  $X$  in a feature map, we calculate an output:  $o = \tanh(X * WE_c + b_c)$ .

(3) *Sub-sampling*: this operation is performed between  $C_1$  layer and  $S_1$  layer, and

between  $C_2$  layer and  $S_2$  layer. Given a feature map, the sub-sampling operation extracts the sampling information from the map to reduce the computation in the next operations and also provides robustness of position. We adopt max-pooling as a sub-sampling operation because it is an effective way to reduce the dimensions of intermediate representations [81]. Given a feature map with size  $m' * n'$ , we first partition the map into  $l$  non-overlapping regions with size  $\frac{m'}{l} * \frac{n'}{l}$ . In each region, the maximum value is selected and then mapped to the corresponding feature map in the next layer.

(4) *Parameter Update*: backward propagation is applied to update the parameters used in the full connection layers and the convolutional layers. In particular, we first calculate the gradients  $\frac{\partial l}{\partial WE}$  and  $\frac{\partial l}{\partial b}$ , respectively. Let  $lr$  denote the learning rate, the parameters are updated as follows:  $WE = WE - lr * \frac{\partial l}{\partial WE}$  and  $b = b - lr * \frac{\partial l}{\partial b}$ . By doing the convolution operation and the sub-sampling operation, we complete the forward propagation that transforms an input WTM to a probability vector. In CLnet-6, we apply the Stochastic Gradient Descent (SGD) with mini-batches to update the parameters because it is efficient.

## 5.3 Experiments and Analysis

### 5.3.1 Data Preparation

In crowdsourcing environments, most of the existing studies adopt synthetic datasets, e.g., [153] or datasets consisting of a few real tasks, e.g., [136]. These datasets are not suitable for our experiments because they do not contain the transaction records of all the crowdsourcing participants for constructing a CTN. By contrast, we use a real-world crowdsourcing processing dataset *wiki-RfA*<sup>1</sup> containing complete transaction records in wikipedia to construct a CTN. Based on the CTN, we evaluate the effectiveness of our model in identifying a worker's identity. Moreover, to further evaluate the effectiveness of our model in identifying spam workers who possess dif-

<sup>1</sup><https://snap.stanford.edu/data/wiki-RfA.html>

ferent degrees of guises **G1** and **G2**, we conduct experiments on three semi-synthetic datasets *soc-sign-epinions 1-3*. The *soc-sign-epinions 1-3* are all generated from a real-world dataset *soc-sign-epinions*<sup>2</sup> that contains the required transaction records for constructing a CTN.

(1) *wiki-RfA*: this real-world dataset records the long-term crowdsourcing processes, i.e., administrator elections on wWikipedia. In particular, the votes (answers) are submitted by voters (workers) in the elections for an administrator (a requester). In this dataset, a requester may have launched several times of elections, thus we take the final election result as the ground truth to label workers' identities. Note that, in practice, repeating a crowdsourcing process many times to obtain the ground truth is very costly and thus is infeasible. As such, our model is proposed for predicting workers' identities by only using the labels derived from partial ground truth rather than attempting to costly pursue the global ground truth. In particular, we label the identities of totally 1,880 workers. 1418 workers are labeled as honest workers because their supported requesters are always elected as administrators and their opposed requesters are always denied to be administrators. In addition, 48 workers with definitely contrary behaviours are labeled as spam workers. The remaining 414 workers are then labeled as grey workers. In the experiments, half of the 1800 workers are used for training, and the other half of the workers are used for testing.

(2) *soc-sign-epinions 1-3*: *soc-sign-epinions* contains the complete transaction-based trust relations between reviewers (workers) and review verifiers (requesters), and thus can be used for constructing a CTN. Based on it, we generate three semi-synthetic datasets *soc-sign-epinions 1-3* to contain spam workers with different degrees of guises **G1** and **G2**. In particular, to generate spam workers with different degrees of guises **G1** and **G2**, the percentages of the attack edges in all the edges of a spam worker are set as 10%, 30% and 50%, respectively. Moreover, we randomly generate workers who possess the transaction behaviours of grey workers. In each dataset, the spam workers and the grey workers are commonly set to be less than 10% of all the workers. As the

<sup>2</sup><https://snap.stanford.edu/data/soc-sign-epinions.html>

---

answer of a shadow task is revealed to spam workers and grey workers beforehand, the probabilities that they can succeed in a shadow task are all set as 100%.

### 5.3.2 Comparison Models

To evaluate the effectiveness of our proposed trust network-based spam worker identification model, we compare it with both the state-of-the-art *reputation-based defense models* in crowdsourcing environments (i.e., AMT and H2010e) and the promising *trust network-based defense models* (i.e., SybilDefender and DeepWalk) that can be adapted to crowdsourcing. All of them have been reviewed in Related Work.

- **AMT**: a model that uses the overall answer approval rate to judge if a worker is trustworthy, and is applied in the most popular crowdsourcing platform Amazon Turk [110].
- **H2010e**: a model that leverages a worker’s sequential performance for differentiating between spam workers and honest workers [153].
- **SybilDefender**: a model that uses the average numbers of frequently appearing nodes in the random walks starting from trust seeds, and the standard error of the average numbers for identifying spammers. [142].
- **Deepwalk**: a model that leverages SkipGram to learn latent representations of nodes from a social network for node classification [111].

SybilDefender and Deepwalk are applied in a trust network where each two directly connected nodes trust each other. In a CTN, as a trust edge connects two nodes who directly trust each other. Thus, trust edges in a CTN are adopted to build a trust network for adapting SybilDefender and Deepwalk to crowdsourcing environments.

### 5.3.3 Parameter and Measure Settings

In the training of CLnet-6, the batch size is set as  $0.05\eta$ , where  $\eta$  is the total number of training samples, and the number of feature maps in convolutional layers are set as  $\iota_{c_1} = 20$  and  $\iota_{c_2} = 50$ , respectively. The maximum number of epochs  $\tau$  is set as  $\tau = 150$ ; All the experiments are implemented by using Theano in a Ubuntu 16.04.1 system with 16 GB RAM. To evaluate the effectiveness of each model, we calculate the precision, the recall. Moreover, to evaluate the trade-off between precision and recall, we apply  $F\text{-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Recall} + \text{Precision}}$ .

### 5.3.4 Experimental Results

#### 5.3.4.1 Experiment 1 (Effectiveness in Identifying Worker Identity)

**Results:** (1) *wiki-RfA*: our model is the best one for identifying honest workers. In particular, our model delivers the highest F-measure value 0.9908 which is 21.68% higher than that of the second best model Deepwalk. Interestingly, all the four comparison models deliver satisfactory precision values in identifying honest workers (see Table 5.1). Regarding the performance in identifying grey workers, our model possesses the highest F-measure value 0.974, which improves that of the second best model Deepwalk by 341.12%. Regarding the most important part, i.e., the effectiveness in identifying spam workers, our model is also the best one in terms of both precision and recall. In particular, the precision value and the recall value of our model are 0.9321 and 1, respectively, which are about 4.5 times and 14.5 times higher than the best results delivered by all the four comparison models.

Table 5.1: The Comparison of Different Models in Identifying Workers with Different Identities

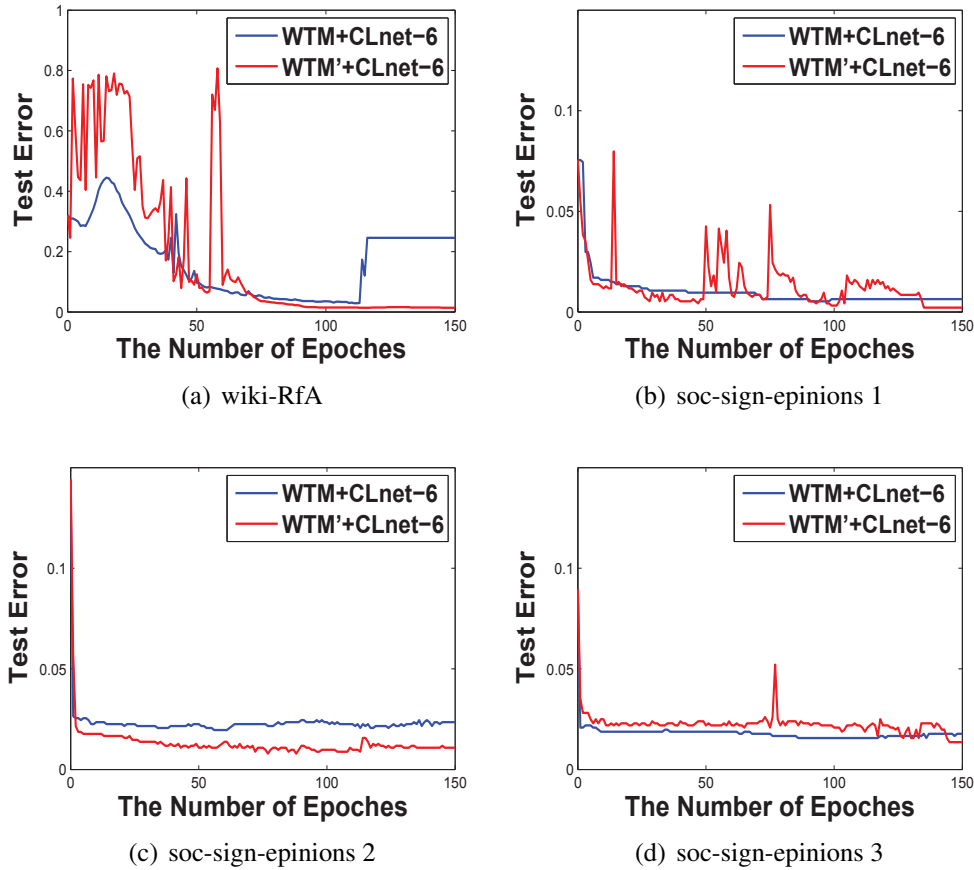
Datasets	Models	Honest Worker			Grey Worker			Spam Worker		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Wiki-RFA	<b>Our model</b>	<b>0.9986</b>	<b>0.9831</b>	<b>0.9908</b>	<b>0.9537</b>	<b>0.9952</b>	<b>0.9740</b>	<b>0.9231</b>	<b>1</b>	<b>0.96</b>
	Deepwalk	0.7812	0.8504	0.8143	0.2535	0.1956	0.2208	0.1667	0.0590	0.0872
	SybilDefender	0.7708	0.1031	0.1819	-	-	-	0.0313	0.8709	0.0604
	H2010e AMT	0.7426 0.7426	1 1	0.8599 0.8599	- -	- -	- -	0 0	0 0	0 0
Epinions1	<b>Our model</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0.9459</b>	<b>1</b>	<b>0.9722</b>	<b>1</b>	<b>0.9444</b>	<b>0.9714</b>
	Deepwalk	1	0.9970	0.9985	0.4615	0.3529	0.4000	0.4211	0.5714	0.4849
	SybilDefender	0.9153	0.0620	0.1161	-	-	-	0.0370	0.9167	0.0711
	H2010e AMT	0.9245 0.9245	1 1	0.9608 0.9608	- -	- -	- -	0 0	0 0	0 0
Epinions2	<b>Our model</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0.9079</b>	<b>0.9857</b>	<b>0.9452</b>	<b>0.9859</b>	<b>0.9091</b>	<b>0.9459</b>
	Deepwalk	1	0.9907	0.9953	0.68	0.5	0.5763	0.6170	0.8286	0.7073
	SybilDefender	0.8704	0.0537	0.1012	-	-	-	0.0753	0.9481	0.1395
	H2010e AMT	0.8559 0.8559	1 1	0.9223 0.9223	- -	- -	- -	0 0	0 0	0 0
Epinions3	<b>Our model</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0.6923</b>	<b>0.9643</b>	<b>0.8060</b>	<b>0.9787</b>	<b>0.7931</b>	<b>0.8762</b>
	Deepwalk	1	0.9935	0.9968	0.1667	0.067	0.0952	0.6216	0.8846	0.7302
	SybilDefender	0.9048	0.0648	0.1209	-	-	-	0.0598	0.9310	0.1124
	H2010e AMT	0.9104 0.9104	1 1	0.9531 0.9531	- -	- -	- -	0 0	0 0	0 0



(2) *soc-sign-epinions 1-3*: When the spam workers possess different degrees of guises **G1** and **G2**, i.e., the percentage of the attack edges in all the edges of a spam worker increases from 10% to 50% with a step of 20%, our model maintains the highest F-measure values in identifying all types of workers. Though the second best model Deepwalk delivers satisfactory F-measure values in identifying honest workers, the F-measure values of our proposed model are still higher than those of Deepwalk (see Table 5.1). Regarding the performance in identifying grey workers in all the three datasets, our model possesses the highest F-measure values.

On average, the F-measure value of our model in identifying grey workers is 0.9078, which improves the best result delivered by all the four comparison models by 154.14%. Regarding the performance in identifying spam workers in the three datasets, the F-measure values of our model are as high as 0.9714, 0.9459, and 0.8762, respectively, which are 100.33%, 33.73%, and 19.99% higher than those of the second best model Deepwalk.

**Analysis:** (1) In *wiki-RfA* dataset, all the four comparison models deliver poor performance in identifying spam workers demonstrates that spam workers in real-world crowdsourcing environments possess guises **G1** and **G2** and thus can bypass the general defense models. (2) With the percentage of the attack edges in all the edges of a spam worker increasing from 10% to 50% with a step of 20%, the collusion between requesters and spam workers becomes more frequently. Nevertheless, our proposed model still maintains the highest precision rate between 0.978 and 1 in identifying spam workers. This is because our proposed model leverages the trust network-based features for identifying spam workers, which is not influenced by the number of attack edges owned by a spam worker. (3) The superior performance of our model on both *wiki-RfA* and *soc-sign-epinions 1-3* datasets results from the fact that WTM is un-manipulable and contains the critical trust network-based features for identifying workers' identities.



**Figure 5.4:** The Comparison of Test Errors of  $WTM$ -based CLnet-6 and  $WTM'$ -based CLnet-6

#### 5.3.4.2 Experiment 2 (Effectiveness of Low-Dimensional $WTM$ ).

**Results:** Figs. 5.4(a)-5.4(d) depict the test errors achieved by CLnet-6 algorithm that respectively takes the original  $WTMs$  and the low-dimensional  $WTM'$ s as inputs. We can observe that the low-dimensional  $WTM'$ s help CLnet-6 achieve the lowest test errors in a small number of epochs. In particular, the minimal errors achieved by  $WTM'$ -based CLnet-6 in 150 epochs are 0.0021, 0.0078, and 0.0135, which are 43.98%, 60%, and 13.33% lower than the minimum test errors delivered by  $WTM$ -based CLnet-6, respectively.

**Analysis:** The experimental results demonstrate that our proposed dimension reduc-

---

tion method for WTMs not only can preserve the un-manipulable property and the usable property of a WTM but also can help CLnet-6 achieve higher effectiveness in identifying a worker's identity.

## **5.4 Conclusion**

In this chapter, we have proposed an un-manipulable Worker Trust Matrix (WTM) for realizing the effective spam worker identification. Moreover, we have incorporated a random walk-based WTM estimation algorithm with a convolutional neural network-based algorithm CLnet-6 to predict a worker's identity. The experiments have demonstrated the superiority of our model in identifying spam workers. CLnet-6 can be directly applied to a practical crowdsourcing platform because the transaction data are always available to the operator of a crowdsourcing platform.

## Chapter 6

---

# Trust-aware Worker Recommendation

---

In Chapter 3, we have introduced a context-aware trust-based worker selection model. Essentially, this model helps a requester effectively obtain a global view of a worker’s performance in different types of tasks with varying reward amounts. Based on the results of trust evaluation, our proposed worker selection model can pick out those *reliable* and *capable* workers. However, the context-aware trust evaluation is still vulnerable to the spam workers who masquerade themselves as “good” workers via colluding with its accomplices in particular contexts. Thus, in Chapter 4 and Chapter 5, we have proposed two spam worker defense models where the *truthfulness* of trust is investigated for defending against spam workers. With the effective context-aware trust evaluation-based worker selection and the spam worker defense models, the *reliable*, *truthful*, and *capable* workers can be found out. However, due to the ‘first come first served’ basis of crowdsourcing, some of the most capable workers may miss the opportunities to participate in tasks. Thus, in this chapter, we propose a trust-aware worker recommendation model that help recommend the most capable workers to participate in a task, which can be further combined with our proposed trust evaluation model and spam worker defense models for selecting the most trustworthy workers in a task.

In the literature, several worker-oriented recommendation models have been proposed. However, a worker who is interested in a task may not be the suitable one who is most likely to provide a correct answer in the task. By contrast, from the perspective of a requesters, our model recommends workers to a task published by a requester by

investigating the implicit interests of the requester based on the workers' past transaction records. The advantage is that such a requester-oriented recommendation model takes the worker's objective performance as the basis, which is not influenced by the worker's subjective willingness to obtain more recommendation opportunities. To obtain high-quality recommendations, we have to consider four challenging problems:

**C1 (Homogeneous Workers):** Homogeneous workers have the equal opportunity to get recommendations as they all have good reputations. However, they actually have different performance in a task published by different requesters. This is because two different requesters may have their specific task requirements that reflect the different preferences on the capacity of workers.

**C2 (Dishonest Behaviours):** Workers may boost good reputations in easy tasks and also overstate their skills in their registration in order to obtain more opportunities to be recommended.

**C3 (Data Sparsity):** On a crowdsourcing platform, each element in a requester-worker matrix is a statistic transaction record between a requester and a worker. As a requester may transact with a tiny fraction of all the workers, the requester-worker matrix is very sparse.

**C4 (Cold Start):** As there is nearly no available transaction record of the freshly registered requesters and workers, it is complicated to generate recommendations for the freshly registered requesters but also to get the freshly registered workers to be recommended.

Thus, we first propose two metrics to evaluate the similarities between two requesters who have transacted with common workers. In particular, one similarity metric indicates if two requesters similarly trust a worker, while the other one indicates if the two requesters similarly distrust a worker. Based on the two types of trust-aware similarities of requesters, we further propose an implicit metric to measure the trust-aware similarity between two requesters who have not transacted with common work-

ers. Given a target requester, by discovering the trustworthy requesters who are similar to the requester, we can obtain more valuable suggestions for generating recommendations. In particular, the trust-aware similarity metrics help solve the challenges **C1**, **C2**, and **C3** well. Targeting the challenge **C4**, we propose two strategies for recommending workers for a task published by a freshly registered requester and for letting freshly registered workers have the opportunities to be recommended, respectively. The experiments conducted in a simulated crowdsourcing platform demonstrate the superiority of our proposed recommendation model CrowdRec in terms of both accuracy and coverage.

## 6.1 Trust-based Similarity Metrics

In a social network, recommendations are generated by asking the target user’s friends or his/her friends’ friends. However, on crowdsourcing platforms, there is no social connection available between two requesters. Two requesters are connected via the workers with whom they have transacted. We refer to the requesters who have transacted with common workers as *explicitly connected requesters*, and refer to the requesters who have not transacted with common workers as *implicitly connected requesters*. Accordingly, we define the direct similarity metrics for measuring the degree of similarity between two explicitly connected requesters, and the indirect similarity metrics for measuring the degree of similarity between two implicitly connected requesters.

### 6.1.1 Explicit Similarity Metrics

#### 6.1.1.1 The Basic Explicit Similarity Metrics

**Definition 6.1. Explicit Trust Similarity**  $sim_{(r_i, r_j)}^t$  indicates the probability that a worker may have a same satisfied performance in a task published by either requester  $r_i$  or requester  $r_j$ .

**Definition 6.2. Explicit DisTrust Similarity**  $sim_{(r_i, r_j)}^d$  indicates the probability that a worker may have a same unsatisfied performance in a task published by either requester  $r_i$  or requester  $r_j$ .

Based on the direct trust defined in Definition 4.1, the explicit trust-based similarity  $sim_{(r_i, r_j)}^t$  between requester  $r_i$  and requester  $r_j$  is modelled by Eq. (6.1) based on Pearson correlation coefficient. Likewise, the explicit distrust-based similarity  $sim_{(r_i, r_j)}^d$  is modelled by Eq. (6.2):

$$sim_{(r_i, r_j)}^t = \frac{\sum_{w_k \in W^t} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})(dt_{(r_j, w_k)} - \overline{dt}_{r_j})}{\sqrt{\sum_{w_k \in W^t} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})^2} \sqrt{\sum_{w_k \in W^t} (dt_{(r_j, w_k)} - \overline{dt}_{r_j})^2}}, \quad (6.1)$$

where  $W^t$  is the set of common workers who have been trusted by both the requester  $r_i$  and the requester  $r_j$ ;  $dt_{r,w}$  is the direct trust between a requester  $r$  and a worker  $w$ ;  $\overline{dt}_r$  is the average direct trust between requester  $r$  and the workers who have transacted with the requester  $r$ .

$$sim_{(r_i, r_j)}^d = \frac{\sum_{w_k \in W^d} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})(dt_{(r_j, w_k)} - \overline{dt}_{r_j})}{\sqrt{\sum_{w_k \in W^d} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})^2} \sqrt{\sum_{w_k \in W^d} (dt_{(r_j, w_k)} - \overline{dt}_{r_j})^2}}, \quad (6.2)$$

$W^d$  is the set of common workers who have been distrusted by both the requester  $r_i$  and the requester  $r_j$ . The two types of similarities between two requesters is calculated according to the direct trust values between them and their commonly trusted workers and distrusted workers, respectively. This is because the meaning of trusting a common worker is different from the meaning of distrusting a common worker. An untrustworthy worker may be distrusted by most requesters, however, a trustworthy worker may only be trusted by a group of requesters who publishes similar tasks. Thus, given two requesters, the  $sim^t$  indicates their similarity derived from the common workers they

trust and the  $sim^d$  indicates their similarity from the common workers they distrust. In practice, given a target requester, only the requesters who have a positive value of similarity are considered in generating recommendations. Therefore, in our proposed trust model, we use positive  $sim^t$  and  $sim^d$  to infer the trust between two requesters.

### 6.1.1.2 The Revised Explicit Similarity Metrics

Suppose there are two requesters who have transacted with only one common worker, either the  $sim^t$  or the  $sim^d$  between the two requesters may be a high value if the common worker similarly performs when transacting with them. Thus, we revise the similarity between two requesters by considering the proportion of common workers. Specifically, in the calculation of  $sim^t$  between requester  $r_i$  and requester  $r_j$ , we consider the proportion of the commonly trusted workers  $pw^t$ . In the calculation of  $sim^d$  between requester  $r_i$  and requester  $r_j$ , we consider the proportion of commonly distrusted workers  $pw^d$ . In Eq. (3), we present how to calculate the  $pw^t$  and the  $pw^d$ :

$$PW_{(r_i,r_j)} = \{pw^t_{(r_i,r_j)}, pw^d_{(r_i,r_j)}\} = \left\{ \frac{n^t_{(r_i,r_j)}}{m^t_{(r_i,r_j)}}, \frac{n^d_{(r_i,r_j)}}{m^d_{(r_i,r_j)}} \right\}, \quad (6.3)$$

where  $n^t_{(r_i,r_j)}$  denotes the number of common workers who are trusted by both the requester  $r_i$  and the requester  $r_j$ ;  $m^t_{(r_i,r_j)}$  denotes the total number of workers who are trusted by any of the two given requesters;  $n^d_{(r_i,r_j)}$  denotes the number of common workers who are distrusted by both  $r_i$  and  $r_j$ ;  $m^d_{(r_i,r_j)}$  denotes the total number of workers who are distrusted by any of the two given requesters.

As either  $pw^t$  or  $pw^d$  may be a small value that is very close to 0, both  $pw^t$  and  $pw^d$  cannot be directly used as the weights of  $sim^t$  and  $sim^d$ , respectively. Therefore, we map  $pw^t$  and  $pw^d$  into the range  $[0, 1]$  by using sigmoid function:

$$pw'_{(r_i,r_j)} = \frac{1}{1 + e^{-\tau pw_{(r_i,r_j)}}}, \quad (6.4)$$

where  $\tau$  is a variant to magnify the  $pw$  in order to decrease the gap among the values



of  $pw$ . Given a target requester, the value of  $\tau$  is set to the reciprocal of the average value of  $pw$  between the target requester and other requesters.

$$\tau = \frac{1}{\overline{pw}}, \quad (6.5)$$

According to the similarities calculated in Eq. (6.1) and (6.2), the revised similarities  $sim^t$  and  $sim^d$  is calculated as follows,

$$sim_{(r_i, r_j)}^t = \frac{pw_{(r_i, r_j)}^t \sum_{w_k \in W^t} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})(dt_{(r_j, w_k)} - \overline{dt}_{r_j})}{\sqrt{\sum_{w_k \in W^t} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})^2} \sqrt{\sum_{w_k \in W^t} (dt_{(r_j, w_k)} - \overline{dt}_{r_j})^2}}, \quad (6.6)$$

$$sim_{(r_i, r_j)}^d = \frac{pw_{(r_i, r_j)}^d \sum_{w_k \in W^d} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})(dt_{(r_j, w_k)} - \overline{dt}_{r_j})}{\sqrt{\sum_{w_k \in W^d} (dt_{(r_i, w_k)} - \overline{dt}_{r_i})^2} \sqrt{\sum_{w_k \in W^d} (dt_{(r_j, w_k)} - \overline{dt}_{r_j})^2}}. \quad (6.7)$$

## 6.1.2 Implicit Similarity Metrics

Given a target requester, as there may be few explicitly connected requesters who can provide opinions on a target worker's performance, we have to consider implicitly connected requesters' opinions. Therefore, we propose a novel type of implicit similarity to measure the trust-aware similarities between both two explicitly connected requesters and two implicitly connected requesters.

**Definition 6.3. Implicit Trust Similarity** between two requesters is the variant frequency that reflects how often  $r_j$  appears in the  $r_i$ -centered subnetwork consisting of explicitly connected requesters with trust similarity.

**Definition 6.4. Implicit DisTrust Similarity** between two requesters is the variant frequency that reflects how often  $r_j$  appears in the  $r_i$ -centered subnetwork consisting of explicitly connected requesters with distrust similarity.

We model the implicit trust similarity  $imsim_{(r_i, r_j)}^t$  and implicit distrust similarity  $imsim_{(r_i, r_j)}^d$  from requester  $r_i$  to requester  $r_j$  as follows:

$$imsim_{(r_i, r_j)}^t = \frac{\sum_{k=1}^{\zeta} \prod_{l=2}^{\eta} sim_{r_{(l-1)}, r_l^k}^{t'}}{n}, \quad (6.8)$$

$$imsim_{(r_i, r_j)}^d = \frac{\sum_{k=1}^{\zeta} \prod_{l=2}^{\eta} sim_{r_{(l-1)}, r_l^k}^{d'}}{n}, \quad (6.9)$$

where  $\zeta$  is the total number of  $r_i$ -centered paths consisting of explicitly connected requesters;  $r_l^k$  is the  $l^{th}$  requester in the number  $k$  path;  $\eta$  is the total number of requesters in the number  $k$  path;  $sim^{t'}$  and  $sim^{d'}$  are the revised similarity metrics defined in Eq. (6.6) and Eq. (6.7), respectively. Given a path starting from the source requester  $r_i$  to a requester  $r_j$ , the implicit similarity between  $r_j$  and  $r_i$  in this path is determined by the length of the path and the values of similarity between  $r_i$  and  $r_j$ . In such a path, the farther  $r_j$  is from  $r_i$ , the lower similarity of  $r_j$  should be. For example, in a path  $path_1 : r_1 \rightarrow r_2 \rightarrow r_3$ , the implicit similarity between  $r_1$  and  $r_3$  is weaker than the implicit similarity between  $r_1$  and  $r_2$  from  $r_1$ 's point of view. Thus, we use the product of all  $sim'$  values in the trust path between  $r_i$  and  $r_j$  to model the  $r_j$ 's trustworthiness from  $r_i$ 's point of view. For example, in path  $tp_1$ ,  $sim'_{(r_1, r_3)} = sim'_{(r_1, r_2)} * sim'_{(r_2, r_3)}$ . By exploring all  $r_i$ -centered paths, we can calculate the implicit trust-based similarity and implicit distrust-based similarity between  $r_i$  and  $r_j$  according to Eq. (6.8) and Eq. (6.9), respectively.

## 6.2 A Trust-aware Recommendation Method in Crowdsourcing

Given a task published by a target requester  $r^*$ , the recommendation method needs to correctly predict an unknown worker's performance in this task when transacting with the target requester  $r^*$ . In this section, we first present how to exact the trust similarity and distrust similarity networks. Then, we introduce the trust-aware recommendation method in detail.

### 6.2.1 A Trust-based Similarity Network Extraction Algorithm

Given a target requester, we first need to find out the requesters who are similar to the target requester and then can use their transaction records as the basis to generate worker recommendations. As such, the *implicit trust similarity* and the *implicit distrust similarity* between the target requester and all the other requesters should be calculated beforehand. In general, traversing all  $r^*$ -centered trust paths is an intuitive solution, however, the complexity of traverse exponentially grows with the number of requesters who explicitly connects with a requester. Thus, we propose a random walk based searching algorithm to extract a sub-network in a lower complexity, which consists of the most similar requesters to the target requester and the  $imsim^t$  and  $imsim^d$  values between the similar requesters and the target requester can be estimated. In our algorithm,  $r$  denotes a general requester,  $r^*$  denotes a target requester for whose tasks we generate worker recommendations,  $w$  denotes a general worker, and  $hop_{max}$  is the maximum number of hops in one time of searching.

#### 6.2.1.1 Description of Algorithm

Below we present the Trust-based Similarity Network Extraction Algorithm in detail.

**Step 1:** Initiate all the implicit trust similarity and the implicit distrust similarity between target requester  $r_t$  and other requesters in  $R$  as  $imsim^t_{(r_t, r \in R)} = 0$  and  $imsim^d_{(r_t, r \in R)} = 0$ , the number of searched paths starting from target requester  $r_t$  as  $nsp_{r_t} = 0$ , the current number of searching hops  $hop_c = 0$ , respectively. Set the target requester  $r_t$  as the start node  $r^*$ .

**Step 2:** Fill the set  $TW_{r^*}$  with the workers who are directly trusted by  $r^*$  in the past transactions, i.e.,  $dt_{(r^*, w \in TW_{r^*})} \geq \varepsilon$ . Fill the set  $DW_{r^*}$  with the workers who are directly distrusted by  $r^*$ , i.e.,  $dt_{(r^*, w \in DW_{r^*})} < \varepsilon$ .

**Step 3:** Randomly select number  $\mu$  workers from the set  $TW_{r^*}$ , then store them in the set  $RW = \{rw_1, rw_2, \dots, rw_\mu\}$ . Update the value of  $hop_c$  by the value of  $hop_c + 1$ ; Record the requesters who directly trust any worker of  $RW$  in  $UR$ .

**Step 4:** Randomly select number  $h$  requesters from the  $UR$ , and then add each selected requester  $cr$  into a candidate set  $CR = \{cr_1, cr_2, \dots, cr_h\}$ .

**Step 5:** Calculate the  $sim^t$  and  $sim^d$  between  $r^*$  and each  $cr \in CR$  according to Eq. (6.6) and Eq. (6.7), record the top  $\kappa$  requesters who have high trust similarity  $sim^t$  to requester  $r^*$  in  $PR$ . Then, replace  $r^*$  with the requester  $pr \in PR$  who possesses the highest distrust similarity  $sim^d$ .

**Step 6:** Check whether  $pr$  exists in the current searching path. If the result is true, go to **Step 8**; Otherwise, go to **Step 7**;

**Step 7:** Update the value of  $imsim^t_{(r_t, r^*)}$  and  $imsim^d_{(r_t, r^*)}$  according to Eq. (6.8) and Eq. (6.9).

**Step 8:** Check whether the current searching exceeds the preset maximum number of searching hops  $hop_{max}$ . If  $hop_c == hop_{max}$ , update  $nsp_{r_t} = nsp_{r_t} + 1$ , and then go to **Step 9**; Otherwise, update  $hop_c = hop_c + 1$ , and then go to **Step 10**.

**Step 9:** Check whether the searching has satisfied any of the termination conditions, i.e., the number of random walks exceeds the maximum times or the values of  $imsim$  converges. If any condition is true, output values of  $imsim$  and terminate; Otherwise, go to **Step 10**.

**Step 10:** Refill the set  $TW$  with the workers who are directly trusted by the requester  $r^*$  and refill the set  $DW$  with the workers who are not directly trusted by the requester  $r^*$ . Then go to **Step 3**.

In the algorithm, selecting requesters in **Step 5** is the most time-consuming process because it needs to sort the requesters according to the  $sim^t$  and  $sim^d$ . The average time complexity of the algorithm is  $mO(n \log n)$ , where  $m$  is the maximum number of iterations, and  $n$  is the average number of requesters with whom a requester explicitly connects. We adopt  $sim^t$  and  $sim^d$  as the heuristic factors when selecting a requester as the next  $r^*$ , which makes it more likely to connect two similar requesters in a path. Essentially, this action helps quickly obtain a set of similar requesters with the calculated  $imsim$  values from whom the transaction records for generating recommendations are obtained.

### 6.2.1.2 Termination of Algorithm

TSNE is terminated when the pre-set maximum searching times is exceeded or the estimated implicit similarity values of the most influential requesters' converge. The purpose of extracting trust-based similarity sub-network is to find the requesters who are trustworthy and are more similar to the requester  $r_t$ . Therefore, those requesters whose  $imsim^t$  and  $imsim^d$  values are above the average levels as the most influential requesters.

$$\sigma^2 = \frac{\sum_{r \in S} (imsim_{(r^*, r)}^t - \overline{imsim}_{(r^*, r)}^t)^2 + \sum_{r \in S} (imsim_{(r^*, r)}^d - \overline{imsim}_{(r^*, r)}^d)^2}{|S|} \quad (6.10)$$

where  $S$  is a set of requesters whose  $imsim^t$  and  $imsim^d$  are above the average levels in a searching iteration.  $\overline{imsim}_{(r^*, r)}$  is the average value of  $imsim$  between a target requester  $r^*$  and all requesters in  $S$ , and  $|S|$  is the number of requesters in  $S$ . When the  $|\sigma_{c+1}^2 - \sigma_c^2| \leq \epsilon$ , the algorithm terminates.

## 6.2.2 Strategies for the Cold Start Problem

Below we present the strategies for generating recommendations for the freshly registered workers and the freshly registered requesters, i.e. the cold start challenges (C4).

**Strategy 1:** For a freshly registered requester  $r_{new}$  who has no transaction records that can be used to generate recommendations, we provide a series of completed tasks for  $r_{new}$ .  $r_{new}$  can select any of the tasks as the ones that are believed by themselves are mostly similar to the tasks will be published by them. Then the transaction records in these projects are temporarily taken as the “existing” transaction records of the  $r_{new}$ , which enables  $r_{new}$  to get the first set of recommended workers for completing tasks.

**Strategy 2:** For a freshly registered worker  $w_{new}$  who has no transaction records that can be used to generate recommendations, we randomly select a series of tasks that have been completed by other workers to test the worker's performance. After the worker completes these testing tasks, if its reputation level reaches the minimum

requirement for taking a task, the testing transaction records of this worker are added into the transaction-based network for generating recommendations.

### 6.2.3 Trust-aware Worker Recommendation

By incorporating the novel similarity metrics, the trust-based similarity sub-network, and the new strategies, we propose a recommendation model to predict a target worker's performance in a published task. First, we calculate the similarities of each pair of explicitly connected requesters. Then, we adopt the trust-based similarity network extraction algorithm to discover the similar requesters whose implicit trust similarity and implicit distrust similarity values are calculated. Furthermore, we divide requesters into two groups according to whether they trust or distrust the target worker. Combining the target worker's performance in past transactions with requesters in different groups, we predict the target worker's performance in a new task published by a requester. Here, a worker's performance represents his/her answer approval rate in the past transactions in a task published by different requesters. For new participants, we apply our proposed strategies to generate initial transaction records for them. The prediction of a worker's performance in a task published by a requester is modelled as follows:

$$p_{(r^*, tt, w^*)}^t = \bar{p}_{(r^*, tt)}^t + \frac{\sum_{r_i \in R^t} (p_{(r_i, tt, w^*)}^t - \bar{p}_{(r_i, tt)}^t) \text{imsim}_{(r^*, r_i)}^t}{\sum_{r_i \in R^t} \text{imsim}_{(r^*, r_i)}^t}, \quad (6.11)$$

$$p_{(r^*, tt, w^*)}^d = \bar{p}_{(r^*, tt)}^d + \frac{\sum_{r_i \in R^d} (p_{(r_i, tt, w^*)}^d - \bar{p}_{(r_i, tt)}^d) \text{imsim}_{(r^*, r_i)}^d}{\sum_{r_i \in R^d} \text{imsim}_{(r^*, r_i)}^d}, \quad (6.12)$$

where  $\bar{p}_{(r^*, tt)}^t$  and  $\bar{p}_{(r_i, tt)}^t$  are the average performance of the workers who are directly trusted by the target requester  $r^*$  and the trustworthy requester  $r_i$  in the task type  $tt$ , respectively;  $\bar{p}_{(r^*, tt)}^d$  and  $\bar{p}_{(r_i, tt)}^d$  are the average performance of the distrusted workers who have transacted with the target requester  $r^*$  and the trustworthy requester  $r_i$  in the task type  $tt$ , respectively;  $p_{(r_i, tt, w^*)}^t$  is the target worker  $w^*$ 's performance in transacting with the requester  $r_i$  in the tasks belonging to the task type  $tt$  if the worker  $w^*$  is trusted

by the requester  $r_i$ ;  $p_{(r_i, tt, w^*)}^d$  is the the target worker  $w^*$ 's performance in transacting with the requester  $r_i$  in the tasks belonging to the task type  $tt$  if the worker  $w^*$  is distrusted by the requester  $r_i$ ;  $R^t$  is a sub-set of the selected similar and important requesters, in which all requesters directly trust the target worker  $w^*$ , i.e.  $dt \geq \varepsilon$ ;  $R^d$  is another sub-set of the selected similar and important requesters, in which all requesters directly distrust the target worker  $w^*$ , i.e.  $dt < \varepsilon$ .

Based on the Eq. (6.11) and Eq. (6.12), we predict the target worker's performance  $p_{(r^*, tt, w^*)}^t$  and  $p_{(r^*, tt, w^*)}^d$  in the task type  $tt$  when transacting with the target requester. To determine the final prediction, we first compare the average value of  $imsim^t$  used in calculating  $p_{(r^*, tt, w^*)}^t$  and the average value of  $imsim^d$  used in calculating  $p_{(r^*, tt, w^*)}^d$ . The  $p$  with a high average value of  $imsim$  is adopted. If the average value of  $imsim^t$  is close to that of  $imsim^d$ , we adopt the predicted performance  $p$  obtained from the group in which there are more requesters. Moreover, if the numbers of requesters in the two groups are equal, we adopt the  $p$  in the group where the standard deviation  $sd$  of predicted performance is lower.

## 6.3 Experiments and Analysis

In the experiments, we evaluate our proposed CrowdRec, and compare it with the conventional user-based CF and three state-of-the-art trust-based recommendation methods, which are described below.

- **User-based CF:** An user-based collaborative filtering recommendation model.
- **TrustWalker:** A trust-aware and item-based recommendation model [65].
- **TSR:** A trust-aware and item-based collaborative filtering recommendation model [34].
- **MTMRRRA:** A multidimensional trust model-based robust recommendation model [66].

## 6.3.1 Experiment Settings

### 6.3.1.1 Experiment Environment

There is no available dataset that including all the worker accounts, requester accounts, and historical transactions records of different types of tasks for evaluating a recommendation method [156]. Thus, based on the statistical data of real crowdsourcing platforms released in NAACL 2010<sup>1</sup>, we simulate the crowdsourcing processes of participants in a crowdsourcing platform for generating transaction data that are further used for evaluating our recommendation model. In particular, according to the statistical data, a requester’s preferences in publishing tasks nearly follow a normal distribution, and a worker’s performance in different types of tasks also nearly follows a normal distribution. Thus, we initiate 15 different types of tasks, 100 types of requesters’ preferences in publishing different types of tasks, and 800 types of homogeneous workers’ performance in different types of tasks. On the simulated crowdsourcing platform, in each iteration, the randomly selected requesters publish different types of tasks, and then the randomly selected workers take and complete the published tasks. Based on the simulation rules, we set four experiments to evaluate the effectiveness of our model in solving the four targeted problems: homogeneous workers, dishonest behaviours, data sparsity and cold start problems. In each experiment, different participants are simulated to transact with each other (see Table 6.1), and 1000 iterations are executed to generate transaction records.

**Table 6.1:** The Number of Different Participants in a Simulated Crowdsourcing Environment

Number Participant Type \ Group No.	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Requester	100	100	100	100
New Requester	-	-	-	10
Homogeneous honest workers	800	800	1600	1600
Homogeneous Dishonest Worker	-	100,200,300,400,500	200	200
Homogeneous New Worker	-	-	-	200

<sup>1</sup><https://sites.google.com/site/amtworkshop2010/data-1>



### 6.3.1.2 Evaluation Method and Metrics

As the leave-one-out method is applied by majority of recommendation models as the evaluation method [65, 34], we use the leave-one-out method to withhold a worker's performance in the past transactions with a requester in a task and try to predict the withheld performance. Regarding the evaluation metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) have been widely used to evaluate the accuracy of recommendation methods in the literature [65, 34]. Thus, in our experiments, we apply them to evaluate the accuracy. For each of MAE and RMSE, a lower value represents a higher accuracy. Below, we present the calculation of MAE and RMSE, respectively.

$$MAE = \frac{\sum pre_{(r_i, tt, w_j)} |\hat{p}_{(r_i, tt, w_j)} - p_{(r_i, tt, w_j)}|}{\sum w_{(r_i, tt, w_j)}}, \quad (6.13)$$

$$RMSE = \sqrt{\frac{\sum pre_{(r_i, tt, w_j)} (\hat{p}_{(r_i, tt, w_j)} - p_{(r_i, tt, w_j)})^2}{\sum pre_{(r_i, tt, w_j)}}}, \quad (6.14)$$

where  $pre_{(r_i, tt, w_j)} = 1$  if the  $w_j$ 's performance in the type  $tt$  of tasks published by requester  $r_i$  is predictable, and 0 otherwise. To evaluate the coverage of recommendation models, we follow the definition used in [133]. The coverage is calculated by performing  $n$  leave-one-out experiments.

$$Coverage = \frac{\sum_{i=1}^n pre_{(r_i, tt, w_j)}}{n}, \quad (6.15)$$

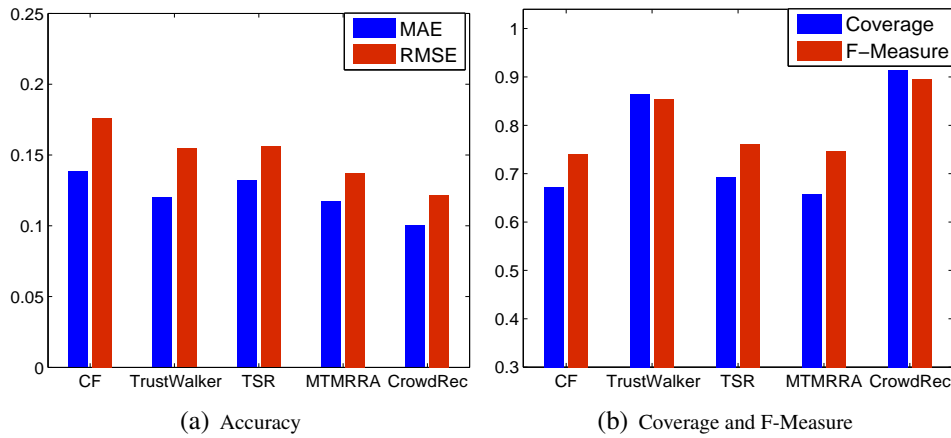
As accuracy and coverage must be considered together, we use the *FMeasure* to evaluate the trade-off between accuracy and coverage.

$$FMeasure = \frac{2 * (1 - RMSE) * Coverage}{1 - RMSE + Coverage}. \quad (6.16)$$

A higher F-Measure represents a better trade-off between accuracy and coverage.

**Table 6.2:** Effectiveness of Different Recommendation Models Under Homogeneous Worker Problem

Method \ Evaluating Metric	MAE	RMSE	Coverage (%)	F-Measure
CF	0.138	0.176	67.21	0.740
TrustWalker	0.121	0.155	86.34	0.854
TSR	0.132	0.156	69.22	0.761
MTMRRRA	0.118	0.137	65.71	0.746
<b>CrowdRec</b>	<b>0.101</b>	<b>0.122</b>	<b>91.34</b>	<b>0.896</b>

**Figure 6.1:** The Effectiveness Comparison of Different Recommendation Models on Homogeneous Workers

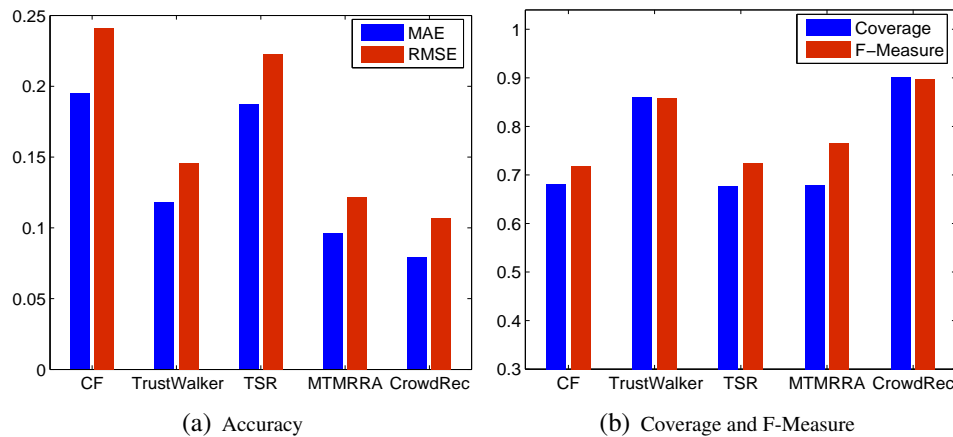
## 6.3.2 Experiment Results

### 6.3.2.1 Experiment 1 (for Homogeneous Workers Problem)

Table 6.2 shows the results of different recommendation models when 800 homogeneous workers are added into the simulated crowdsourcing processes. Fig. 6.1(a) and Fig. 6.1(b) show the comparison of different methods when homogeneous workers exist. From Fig. 6.1(a) and Fig. 6.1(b), we can observe that CrowdRec is the best method in terms of accuracy with the lowest MAE 0.101 and the lowest RMSE 0.122. Moreover, regarding coverage and F-Measure, CrowdRec has the highest values 91.34% and 0.896, respectively. CrowdRec outperforms all the compared methods when facing homogeneous workers because CrowdRec differentiates a worker's performance

**Table 6.3:** Effectiveness of Different Recommendation Models Under Homogeneous Worker and Dishonest Behaviour Problems

Method \ Evaluating Metric	MAE	RMSE	Coverage (%)	F-Measure
CF	0.195	0.241	68.11	0.718
TrustWalker	0.118	0.146	86.14	0.858
TSR	0.188	0.223	67.60	0.723
MTMRRRA	0.097	0.122	67.80	0.765
<b>CrowdRec</b>	<b>0.079</b>	<b>0.107</b>	<b>90.03</b>	<b>0.897</b>



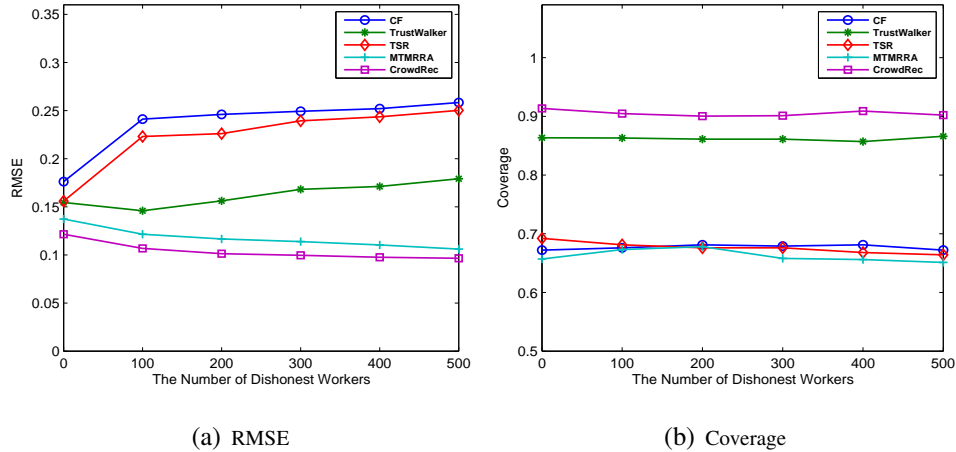
**Figure 6.2:** The Effectiveness Comparison of Different Recommendation Models on Homogeneous Worker and Dishonest Behaviour Problems

in different types of tasks published by different requesters.

### 6.3.2.2 Experiment 2 (for Homogeneous Workers and Dishonest Behaviours Problems)

Table 6.3 shows the results of different recommendation models when 800 homogeneous workers and 100 dishonest workers are simulated to join the crowdsourcing processes. Comparing to the results in Table 6.2, we can observe CrowdRec improves 12.3% and 21.8% in MAE and RMSE, respectively. This is because CrowdRec can obtain the opinions from those requesters who distrust the target worker, which increases the accuracy of CrowdRec in predicting a dishonest worker’s performance. From Fig.

6.2(a) and Fig. 6.2(b), we can observe that CrowdRec is the most effective method with the lowest MAE 0.079, the lowest RMSE 0.107, the highest coverage 90.03% and the highest F-Measure 0.897 when homogeneous workers and dishonest workers co-exist.



**Figure 6.3:** The Effectiveness Comparison of Different Models on Different Number of Dishonest Workers

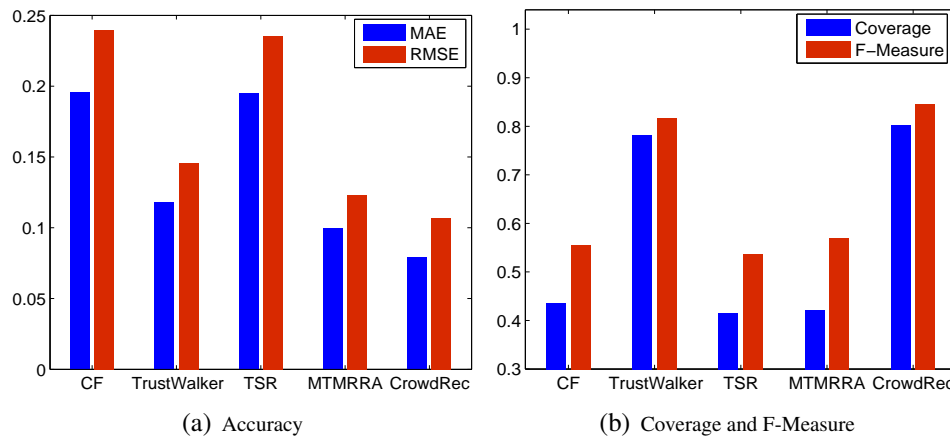
As CrowdRec is a trust-aware similarity-based recommendation method, its robustness is verified by changing the number of dishonest workers in the simulated crowdsourcing processes (see Table 6.1). Fig. 6.3(a) and Fig. 6.3(b) plot the comparison of different models when the number of dishonest workers increases. On average, CrowdRec maintains the lowest RMSE 0.1 which is 11.5% better than that of the second best method MTMRRRA. In the meantime, CrowdRec maintains the highest coverage 90.3%. CrowdRec outperforms all the other four models because it can obtain more opinions from the requesters in the trust-based similarity sub-network.

### 6.3.2.3 Experiment 3 (for Homogeneous Workers, Dishonest Behaviours, and Data Sparsity Problems)

The results of different models when facing the data sparsity problem are listed in Table 6.4. In experiment 3, totally 1800 homogeneous workers are added into the simulated crowdsourcing processes, which makes the transaction data be sparse. Fig.

**Table 6.4:** Effectiveness of Different Recommendation Models Under Homogeneous Worker, Dishonest Behaviour, and Data Sparsity Problems

Method \ Evaluating Metric	MAE	RMSE	Coverage (%)	F-Measure
CF	0.196	0.240	43.57	0.554
TrustWalker	0.118	0.146	78.14	0.816
TSR	0.195	0.235	41.36	0.537
MTMRRRA	0.099	0.123	42.11	0.569
<b>CrowdRec</b>	<b>0.079</b>	<b>0.106</b>	<b>80.21</b>	<b>0.845</b>



**Figure 6.4:** The Effectiveness Comparison of Different Models on Homogeneous Worker, Dishonest Behaviour and Data Sparsity Problems

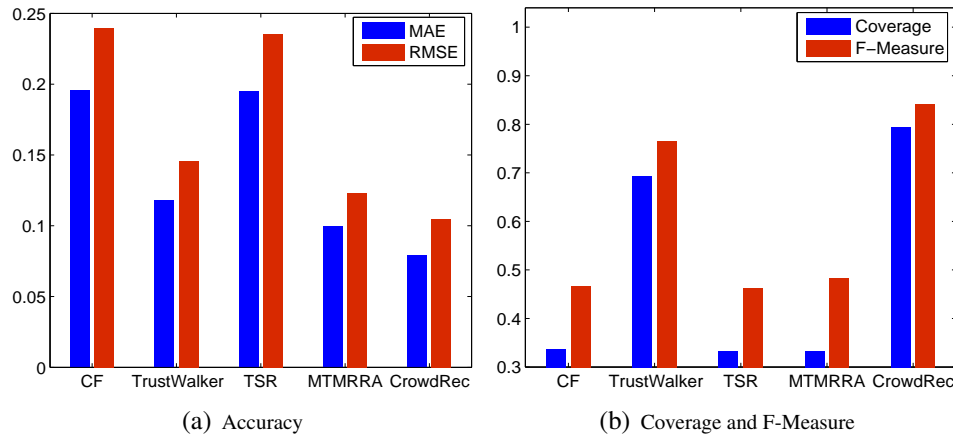
6.4(a) and Fig. 6.4(b) plot the comparison of different methods. We can observe that CrowdRec is the best model with the lowest MAE 0.079, the lowest RMSE 0.106, the highest coverage 80.21% and the highest F-Measure 0.845. This is because CrowdRec can obtain more useful information in a sparse dataset by searching more requesters who are implicitly similar to a target requester.

### 6.3.2.4 Experiment 4 (for Homogeneous Workers, Dishonest Behaviours, Data Sparsity, and Cold Start Problems)

Table 6.5 shows the results of different recommendation models when Homogeneous workers, dishonest behaviours, data sparsity and cold start problems co-exist. Com-

**Table 6.5:** Effectiveness of Different Recommendation Models under Homogeneous Worker, Dishonest Behaviour, Data Sparsity and Cold Start Problems

Method	Evaluating Metric			
	MAE	RMSE	Coverage (%)	F-Measure
CF	0.196	0.240	33.52	0.465
TrustWalker	0.118	0.146	69.21	0.764
TSR	0.195	0.235	33.11	0.462
MTMRRRA	0.099	0.123	33.18	0.481
<b>CrowdRec</b>	<b>0.078</b>	<b>0.104</b>	<b>79.26</b>	<b>0.841</b>

**Figure 6.5:** The Comparison of Different Methods on Homogeneous Worker, Dishonest Behaviour, Data Sparsity, and Cold Start Problems

paring the results in Table 6.5 with those in Table 6.4, we can observe that the values of coverage in CF and the three state-of-art methods obviously decrease. By contrast, the value of coverage in CrowdRec only slightly decreases from 80.21% to 79.26%. This is because CrowdRec adopts two strategies to obtain initial transaction data for freshly registered participants, which can generate recommendations for these freshly registered participants.

Fig. 6.5(a) and Fig. 6.5(b) compare the results of different recommendation methods when homogeneous workers, dishonest behaviours, data sparsity and cold start problems co-exist. In the term of accuracy, the MAE of CrowdRec is the lowest, which is 60.2%, 33.8%, 60.0% and 21.2% better than those in CF, TrustWalker, TSR

and MTMRRRA, respectively. Moreover, the RMSE of CrowdRec is the lowest, which is 56.7%, 28.8%, 55.7% and 15.5% better than those in CF, TrustWalker, TSR and MTMRRRA, respectively. In the meantime, CrowdRec maintains the highest coverage 79.26%, which is 45.74%, 10.05%, 46.15% and 46.08% higher than those in CF, TrustWalker, TSR and MTMRRRA, respectively. Regarding F-Measure, CrowdRec has the highest value 0.841, which is 10.07% better than that in the second best method TrustWalker.

To sum up, CrowdRec significantly outperforms the conventional CF recommendation method and the three state-of-the-art trust-based recommendation methods in both accuracy and coverage when homogeneous workers, dishonest behaviours, data sparsity and cold start problems co-exist.

## 6.4 Conclusion

In this chapter, we have presented a novel trust-aware worker recommendation method CrowdRec. To our best knowledge, in the literature, this is the first model that focuses on recommending workers to a requester and take all the four challenging problems: homogeneous worker, dishonest behaviour, data sparsity and cold start problems into account. In CrowdRec, targeting the homogeneous worker problem, we have proposed the new similarity metrics to obtain more information for improving the accuracy of predicting a worker's performance in a task published by a requester. Moreover, we propose a novel trust sub-network extraction approach to tackling dishonest behaviours and data sparsity problems, and new strategies for cold start problem. The results of experiments have demonstrated that CrowdRec significantly outperforms the conventional user-based CF recommendation model and three state-of-the-art trust-based recommendation models TrustWalker, TSR and MTMRRRA in terms of both accuracy and coverage.

# Chapter 7

---

## Conclusions

---

The explosive development of information techniques makes humans and electronic equipment be tightly connected than ever before. In such a new circumstance, crowdsourcing is widely applied in solving various tasks due to its superior cost-effectiveness. In the meantime, a trust crisis arises in crowdsourcing environments. In particular, the untrustworthy workers take various actions to benefit themselves or sabotage others' crowdsourcing processes. To guarantee a published task can be successfully solved, an effective trustworthy worker selection is urgently demanded. However, the complex meaning of trust in crowdsourcing environments, the new characteristics of crowdsourcing, and the various behaviour patterns of workers make trustworthy worker selection be extremely tough.

In this thesis, we focus on the three sub-challenges of helping crowdsourcing requesters to select trustworthy workers. The first challenge is to evaluate a worker's trust in different contexts so as to select reliable and capable workers under a specific context. In our solution for this challenge, we have taken the *reliable* aspect and the *capable* aspect of trust into account. The second challenge is to defend against those untrustworthy workers who look like "trustworthy" but actually collude to counterfeit "trustworthy" guises. In this sub-challenge, we have proposed our solution that mainly leverages the *truthful* aspect of trust to defend against spam workers. Thirdly, we present how to discover the most preferred workers of a requester by further investigating the *capable* aspect of trust. In particular, targeting the three sub-challenges, we have proposed our solutions.



- We have proposed a context-aware trust-based worker selection model called CrowdTrust that is based on the trust evaluation of a worker in the context of task type and in the context of task reward amount. In particular, our proposed trust evaluation method can effectively differentiate trustworthy workers and untrustworthy workers when both of them have high overall answer approval rates. In CrowdTrust, to avoid the subject bias of weights for aggregating multiple trust values for selecting workers, we have proposed a multiple objectives optimization algorithm to find a worker combination whose trust scores in the two contexts cannot be dominated simultaneously. Our experimental results have shown that CrowdTrust is more effective than the average answer approval rate-based worker selection in differentiating trustworthy workers from untrustworthy workers.
- Targeting the spam workers who masquerade themselves as “trustworthy” workers by applying three typical threat patterns, we have proposed a trust-aware defense model based on a global evaluation on the *truthfulness* of a worker’s trust level. The global view of a worker’s trust level is represented by our proposed worker trust vector. A worker’s trust vector records the inferred trust relations between the worker and different types of requesters in a crowdsourcing trust network. As spam workers with untruthful transaction records always succeed in the transactions with their accomplices, they cannot obtain good trust scores in their trust vectors and thus are prevented from participating in tasks. The experiments have shown that our proposed trust vector-based spam worker defense model effectively outperforms three state-of-the-art models in both selecting trustworthy workers and filtering out spam workers with fake “trustworthy” guises obtained from the colluding transactions.
- Furthermore, we have proposed a novel spam worker identification model that incorporates our devised worker trust matrix and learning algorithm to proactively predict a worker’s identity. A worker trust matrix consists of the trust

indicators that measure the extents to which the worker is trusted by different requesters in different crowdsourcing trust sub-networks. In fact, a worker's trust matrix records the worker's global trust features derived from the worker's past transaction behaviours. We have also proved that a worker's trust matrix contains the un-manipulable property and the usable property that are critical for effectively identifying a worker's identity. We have conducted extensive experiments in a real dataset and three real scenarios-based synthetic datasets, the experimental results have demonstrated the superiority of our model in identifying spam workers.

- Focusing on the *capable* aspect of trust, we have also proposed a trust-aware worker recommendation model. This model first helps a requester to discover potential trustworthy allies by inferring the similarity between two requesters who have no commonly transacted workers. By taking the suggestions of the allies as recommendation basis, the trustworthy workers are then recommended to a requester. The main novelty is that both the trust similarity and the distrust similarity are considered in generating recommendations. Considering the two types of similarity together can help increase the accuracy of predicting a worker's performance. In addition, our model has also contained the effective strategies for solving the data sparsity and the cold start problems in crowdsourcing environments. The experimental results have demonstrated our proposed recommendation model significantly outperforms the conventional CF recommendation model and three state-of-the-art trust-based recommendation models in terms of both accuracy and coverage.

In this thesis, we have mainly discussed the trustworthy worker selection issue in crowdsourcing environments. Besides, there are still some opening trust issues in crowdsourcing environments, e.g., the accountability of crowdsourcing services and the trustworthy maintenance of transaction data. For example, it is very difficult to resolve a dispute between a requester and a worker because each of them may have its

own version of logs or arguments. In a current crowdsourcing site, crowdsourcing operators actually control the functions of accounting for evidence, auditing and dispute arbitration. Thus, the result of a dispute is commonly determined by a crowdsourcing operator who is in charge of the dispute. In addition, the trustworthiness of the transaction data is mostly determined by the operator, leading the corruptions to be possible. In fact, when a dispute arises, an operator may tend to help one of the two involved crowdsourcing participants and thus impairs the benefits of the other one. As such, it is critical to guarantee the fairness of the crowdsourcing operators and the truthfulness of the transaction data for arbitration. In our paper submitted to TSC 2017, we have discussed the feasibility of applying block chain techniques to build a trustworthy consensus mechanism for achieving the accountability of the crowdsourcing services.

# Appendix A

## The Notations in the Thesis

**Table A.1:** The Notations in Chapter 3

<b>Notations</b>	<b>Explanations</b>
$ha$	The number of the approval answers
$hs$	The number of the submitted answers
$hr$	The answer approval rate
$w_i$	The $i$ th worker
$TaTrust$	The task type aware trust of a worker
$RaTrust$	The task reward amount aware trust of a worker
$TC_j$	The $j$ th trust cube
$m$	The number of trust cubes containing records
$k$	The number of the same coordinates
$tin f_{(k,i)}$	The influence factor of the records in trust cube $i$ with $k$ same coordinates
$g(ha, ha_{(k,i),k})$	The independent variable of $tin f_{(k,i)}$
$re^*$	The reward amount of an upcoming task
$re_i$	The reward amount of the task $i$
$d$	The relative distance between two tasks with different reward amounts
$d'$	The distance $d$ based similarity
$n_{d'}$	The number of the tasks in the task group with similarity $d'$
$rin f_{(d',i)}$	The influence factor of the records in task group $i$ with similarity $d'$
$z_i(\widetilde{ha}, \widetilde{ha}_{(d',i)}, d')$	The independent variable of $rin f_{(d',i)}$
$wn$	The number of the workers required by a requester
$aw$	The number of the available workers

**Table A.2:** The Notations in Chapter 3 (continued)

<b>Notations</b>	<b>Explanations</b>
$f_{(X)}$	The Multi-objective optimization function
$X$	The feasible solution
$\overline{TaT_{X_j}}$	The average value of <i>TaTrust</i> of solution $X_j$
$\overline{RaT_{X_j}}$	The average value of <i>RaTrust</i> of solution $X_j$
$WS = \{W_i\}_{i=1}^N$	The initial worker combination set with number $N$ worker combinations
$SumT_{W_i}$	The sum of <i>TaTrust</i> and <i>RaTrust</i> of worker combination $W_i$
$fit_{W_i}$	The fitness value (non-domination level) of worker combination $W_i$
$den_{W_i}$	The density-estimation of the worker combination $W_i$
$WS^{off}$	The offspring worker combination set
$\sigma$	The mutation probability
$\zeta$	The crossover probability
$iter$	The number of iterations
$iter_{max}$	The maximum number of iterations
$M$	The number of optimal objectives

**Table A.3:** The Notations in Chapter 4

<b>Notations</b>	<b>Explanations</b>
$dt_{(r_i, w_j)}$	The direct trust between requester $r_i$ and worker $w_j$
$n_{apv(r_i, w_j)}$	the number of the approved answers submitted by worker $w_j$ in the tasks of requester $r_i$
$n_{sub(r_i, w_j)}$	The total number of the answers submitted by worker $w_j$ in the tasks of requester $r_i$
$\varepsilon$	The threshold of the direct trust relation
$R$	The set of all the requester nodes
$W$	The set of all the worker nodes
$te$	A trust edge
$TE$	The set of all trust edges
$dte$	A distrust edge
$DTE$	The set of all distrust edges

**Table A.4:** The Notations in Chapter 4(continued)

<b>Notations</b>	<b>Explanations</b>
$tp_{(r_i, w_j)}^k$	The $k$ -hop trustworthy path that starts from requester $r_i$ and ends at worker $w_j$
$utp_{(r_i, r_e, w_j)}^k$	The $k$ -hop untrustworthy path that starts from requester $r_i$ and ends at worker $w_j$ who is directly distrusted by requester $r_e$
$tph_{p_{(r_i, w_j)}}$	The aggregation of the direct trust values called trust pheromone in path $p_{(r_i, w_j)}$
$len_{p_{(r_i, w_j)}}$	The number of the edges in path $p_{(r_i, w_j)}$
$TP_{(r_i, w_j)}$	The set of the trustworthy paths that start from requester $r_i$ and end at worker $w_j$
$UTP_{(r_i, w_j)}$	The set of the untrustworthy paths that start from requester $r_i$ and end at worker $w_j$
$SOT_{(r_i, w_j)}$	The strength of trust between requester $r_i$ and worker $w_j$
$n$	The average number of edges owned by a crowdsourcing participant
$hop_{max}$	The maximum number of the hops in a searching path
$stph_{(r_i, w_j)}$	The sum of trust pheromone from $r_i$ to $w_j$
$stph_{r_i}$	The sum of trust pheromone from $r_i$ to all workers
$\sigma^2$	The variant of the estimated value of the strength of trust in two times of searchings
$\epsilon$	The threshold of $\sigma^2$
$DeT_{w_i}$	The deterministic trust value of worker $w_i$
$R_{aut}$	The set of the authenticated requesters
$NDeT_{w_i}$	The non-deterministic trust value of worker $w_i$
$R_{act}$	The set of the active requesters
$DeT_{w_i}$	The deterministic trust value of worker $w_i$
$OT_{w_i}$	The ordinary trust value of worker $w_i$
$R_{ord}$	The set of the randomly selected ordinary requesters

**Table A.5:** The Notations in Chapter 5

<b>Notations</b>	<b>Explanations</b>
$U$	A set of workers whose identities are known
$V$	A set of workers whose identities are unknown
$positrust_{(r_i, w_j)}^k$	The positive trust indicator of worker $w_j$ in the $k$ -hop sub-CTN starting from requester $r_i$
$TP^k(r_i, w_j)$	All the $k$ -hop trustworthy paths between $r_i$ and $w_j$
$DT^{tp_i^k}$	All the direct trust values in the trustworthy path $tp_i^k$
$penalty_{(r_i, w_j)}^k$	The trust penalty on worker $w_j$ in a $k$ -hop sub-CTN
$UTP_{(r_i, w_j)}^k$	All the $k$ -hop untrustworthy paths between $r_i$ and $w_j$
$DT^{utp_h^k}$	All the direct trust values in untrustworthy path $utp_h^k$
$penalty_{(r_i, w_j)}^k$	The trust penalty on worker $w_j$ in a $k$ -hop sub-CTN
$dt_e^{utp_h^k}$	The direct trust value in the only one distrust edge in $utp_h^k$
$tr_{(r_i, w_j)}^k$	The trust trace of worker $w_j$ in a $k$ -hop sub-CTN starting from requester $r_i$
$TP_{r_i}^k$	All the $k$ -hop trustworthy paths that start from requester $r_i$ and end at any worker
$WTM_{w_t}$	The worker trust matrix of worker $w_k$
$pr$	The probability that a spam worker manipulate its $WTM$ to be same as that of an honest worker
$d$	The minimum number of requesters who collude with a worker
$e$	The minimum number of requesters by whom an honest worker is directly trusted
$h$	The number of honest requesters in a crowdsourcing trust network
$c$	The number of colluding requesters in a crowdsourcing trust network
$g$	The number of requesters who collude with a spam worker
$p_w$	The probability that a worker can succeed in a shadow task
$\mathcal{F}$	The distribution function of the trust trace sets between a worker and all honest requesters
$\mathcal{G}$	The distribution function of the trust trace sets between a worker and all grey requesters
$\mathcal{S}$	The distribution function of the trust trace sets between a worker and all spam requesters

**Table A.6:** The Notations in Chapter 5 (continued)

Notations	Explanations
$rod_{max}$	The maximum round number of random walks
$hop_{max}$	The maximum number of hops in a searching path
$de$	The maximum degree of a node
$ST$	A standardized layer in the learning algorithm CLnet-6
$C$	A convolutional layer in the learning algorithm CLnet-6
$S$	A sub-sampling layer in the learning algorithm CLnet-6
$MP$	A multi-layer perceptron in the learning algorithm CLnet-6
$WE$	A weight matrix in the learning algorithm CLnet-6
$\otimes$	The operators of the learning algorithm CLnet-6
$b$	A bias vector in the learning algorithm CLnet-6
$WI = \{HW, GW, SW\}$	The worker identity set including honest worker identity $HW$ , grey worker identity $GW$ , and spam worker identity $SW$
$ID_{w_k}$	The vector for indicating the identity of the worker $w_k$
$\eta$	The total number of training samples
$\iota_c$	The number of feature maps in a convolutional layer
$\tau$	The maximum number of epochs of training

**Table A.7:** The Notations in Chapter 6

Notations	Explanations
$sim_{(r_i, r_j)}^t$	The explicit trust similarity between requester $r_i$ and requester $r_j$
$sim_{(r_i, r_j)}^d$	The explicit distrust similarity between requester $r_i$ and requester $r_j$
$W^t$	The set of common workers who have been trusted by both requester $r_i$ and requester $r_j$
$\overline{dt}_r$	The average direct trust value between requester $r$ and the workers who have transacted with the requester $r$
$W^d$	The set of common workers who have been distrusted by both requester $r_i$ and requester $r_j$
$pw^t$	The proportion of the workers who are commonly trusted by two requesters
$pw^d$	The proportion of the workers who are commonly distrusted by two requesters



**Table A.8:** The Notations in Chapter 6 (continued)

<b>Notations</b>	<b>Explanations</b>
$n_{(r_i, r_j)}^t$	The number of workers who are trusted by both requester $r_i$ and requester $r_j$
$m_{(r_i, r_j)}^t$	The number of workers who are trusted by any of requester $r_i$ and requester $r_j$
$n_{(r_i, r_j)}^d$	The number of workers who are distrusted by both requester $r_i$ and requester $r_j$
$m_{(r_i, r_j)}^d$	The number of workers who are distrusted by any of requester $r_i$ and requester $r_j$
$\tau$	A variant to decrease the gap among the $pw$ between different pairs of requesters
$imsim_{(r_i, r_j)}^t$	The implicit trust similarity between requester $r_i$ and requester $r_j$
$imsim_{(r_i, r_j)}^d$	The implicit distrust similarity between requester $r_i$ and requester $r_j$
$\zeta$	The total number of $r_i$ -centered paths
$r_l^k$	The $l^{th}$ requester in a path
$\eta$	The total number of requesters in a path
$m$	The maximum number of iterations
$n$	The average number of requesters with whom a requester explicitly connects
$S$	The set of requesters whose $imsim^t$ and $imsim^d$ are above the average levels in a searching iteration
$\overline{imsim}_{(r^*, r)}$	The average value of $imsim$ between a target requester and all the other requesters
$\bar{p}_{(r, tt)}^t$	The average performance of the workers who are directly trusted by requester $r$ in task type $tt$
$\bar{p}_{(r, tt)}^d$	The average performance of the workers who are directly distrusted by requester $r$ in task type $tt$
$p_{(r_i, tt, w^*)}^t$	The performance of worker $w^*$ in transacting with requester $r_i$ who directly trusts $w^*$ in the tasks belonging to type $tt$
$p_{(r_i, tt, w^*)}^d$	The performance of worker $w^*$ in transacting with requester $r_i$ who directly distrusts $w^*$ in the tasks belonging to type $tt$
$pre_{(r_i, tt, w_j)}$	The parameter that indicates if $w_j$ 's performance in the type $tt$ of tasks published by requester $r_i$ is predictable

# Appendix B

---

## The Acronyms in the Thesis

---

**Table B.1:** The Acronyms in All the Sections

Sections	Explanations	Acronyms
Chapter 1&3&4&7	Context-aware Trust Evaluation based Crowdsourcing Worker Selection	CrowdTrust
Chapter 1&4&5	Crowdsourcing Trust Network	CTN
Chapter 1&4&5	Worker Trust Vector	WTV
Chapter 1&3	Task Type-aware Trust	TaTrust
Chapter 1&3	Reward Amount-aware Trust	RaTrust
Chapter 1&4	Trust Vector-based Crowdsourcing Spam Worker Defense	CrowdDe- fense
Chapter 1&4	Strength of Trust	SOT
Chapter 1&4	Trust Sub-Network Extraction algorithm	TSE
Chapter 1&5	Worker Trust Matrix	WTM
Chapter 1&6	Trust-aware Crowdsourcing Worker Recommendation	CrowdRec
Chapter 2&6	Collaborative Filtering	CF
Chapter 3	Answer Approval Rate based Random Worker Selection	ARS
Chapter 4	Deterministic Trust	DeT
Chapter 4	Non-Deterministic Trust	NDeT
Chapter 4	Ordinary Trust	OT
Chapter 5	Six-layer Neural Network	CLnet-6
Chapter 6	Stochastic Gradient Descent	SGD
Chapter 6	Mean Absolute Error	MAE
Chapter 6	Root Mean Square Error	RMSE

---

# Bibliography

---

- [1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *33rd Annual Hawaii International Conference on System Sciences (HICSS-33)*, 4-7 January, 2000, Maui, Hawaii, USA, 2000.
- [2] A. Abrizah, D. Nicholas, A. Noorhidawati, Y. I. A. M. Khalid, and F. Badawi. Not so different after all: Malaysian researchers' cross-discipline view of quality and trustworthiness in citation practices. *Learned Publishing*, 29(3):165–172, 2016.
- [3] P. S. Adler. Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization science*, 12(2):215–234, 2001.
- [4] G. A. Akerlof. The market for "lemons": Quality uncertainty and the market mechanism. *The quarterly journal of economics*, pages 488–500, 1970.
- [5] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S. Beheshti, E. Bertino, and N. Foo. Reputation management in crowdsourcing systems. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2012, Pittsburgh, PA, USA, October 14-17, 2012*, pages 664–671, 2012.
- [6] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz. Location-based crowdsourcing: extending crowdsourcing to the real world. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction 2010, Reykjavik, Iceland, October 16-20, 2010*, pages 13–22, 2010.
- [7] V. Ambati, S. Vogel, and J. G. Carbonell. Towards task recommendation in

- 
- micro-task markets. In *Human Computation, Papers from the 2011 AAAI Workshop, San Francisco, California, USA, August 8, 2011*.
- [8] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *J. Web Sem.*, 5(2):58–71, 2007.
- [9] P. S. Aulakh, M. Kotabe, and A. Sahay. Trust and performance in cross-border marketing partnerships: A behavioral approach. *Journal of international business studies*, 27(5):1005–1032, 1996.
- [10] S. Ba and P. A. Pavlou. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS quarterly*, pages 243–268, 2002.
- [11] P. Beatty, I. Reay, S. Dick, and J. Miller. Consumer trust in e-commerce web sites: A meta-study. *ACM Computing Surveys (CSUR)*, 43(3):14, 2011.
- [12] E. Berscheid and H. T. Reis. *Attraction and close relationships*. 1998.
- [13] K. Blomqvist. The many faces of trust. *Scandinavian journal of management*, 13(3):271–286, 1997.
- [14] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.
- [15] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.
- [16] P. Bonatti, C. Duma, D. Olmedilla, and N. Shahmehri. An integration of reputation-based and policy-based trust management. *networks*, 2(14):10, 2007.
- [17] D. C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75–90, 2008.

- 
- [18] S. Brin and L. Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833, 2012.
- [19] R. D. Burke. Hybrid web recommender systems. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 377–408, 2007.
- [20] A. Caballero, J. A. B. Blaya, and A. F. Gómez-Skarmeta. On the behaviour of the TRSIM model for trust and reputation. In *Multiagent System Technologies, 5th German Conference, MATES 2007, Leipzig, Germany, September 24-26, 2007, Proceedings*, pages 182–193, 2007.
- [21] C. Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using amazon’s mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 286–295, 2009.
- [22] C. Callison-Burch and M. Dredze. Creating speech and language data with amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 1–12. Association for Computational Linguistics, 2010.
- [23] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012, San Jose, CA, USA, April 25-27, 2012*, pages 197–210, 2012.
- [24] K. Chen, C. Chang, C. Wu, Y. Chang, and C. Lei. Quadrant of euphoria: a crowdsourcing platform for qoe assessment. *IEEE Network*, 24(2):28–35, 2010.
- [25] W. Chen, R. Khoury, and S. Fong. Web 2.0 recommendation service by multi-collaborative filtering trust network algorithm. *Information Systems Frontiers*, 15(4):533–551, 2013.

- 
- [26] B. Christianson and W. S. Harbison. Why isn't trust transitive? In *International workshop on security protocols*, pages 171–176. Springer, 1996.
- [27] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2009, San Diego, California, USA, 8th February - 11th February, 2009*.
- [28] E. Davami and G. Sukthankar. Evaluating trust-based fusion models for participatory sensing applications. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 1377–1378, 2014.
- [29] E. Davami and G. Sukthankar. Improving the performance of mobile phone crowdsourcing applications. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 145–153. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- [30] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.
- [31] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2):182–197, 2002.
- [32] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 469–478, 2012.
- [33] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Large-scale linked data integration using probabilistic reasoning and crowdsourcing. *VLDB J.*, 22(5):665–687, 2013.

- 
- [34] S. Deng, L. Huang, J. Wu, and Z. Wu. Trust-based personalized service recommendation: A network perspective. *J. Comput. Sci. Technol.*, 29(1):69–80, 2014.
- [35] M. Deutsch. Cooperation and trust: Some theoretical notes. 1962.
- [36] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2-4):97–166, 2001.
- [37] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux. Pick-a-crowd: tell me what you like, and i’ll tell you what to do. In *22nd International World Wide Web Conference, WWW ’13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 367–374, 2013.
- [38] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, 2011.
- [39] C. Eickhoff and A. de Vries. How crowdsourcable is your task. In *Proceedings of the workshop on crowdsourcing for search and data mining (CSDM) at the fourth ACM international conference on web search and data mining (WSDM)*, pages 11–14, 2011.
- [40] E. Estellés-Arolas and F. González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200, 2012.
- [41] S. T. Fiske. *Social beings: Core motives in social psychology*. John Wiley & Sons, 2009.
- [42] O. Folorunso and O. A. Mustapha. A fuzzy expert system to trust-based access control in crowdsourcing environments. *Applied Computing and Informatics*, 11(2):116–129, 2015.

- 
- [43] S. Fritz, I. McCallum, C. Schill, C. Perger, R. Grillmayer, F. Achard, F. Kraxner, and M. Obersteiner. Geo-wiki.org: The use of crowdsourcing to improve global land cover. *Remote Sensing*, 1(3):345–354, 2009.
- [44] D. Gao, Y. Tong, J. She, T. Song, L. Chen, and K. Xu. Top-k team recommendation and its variants in spatial crowdsourcing. *Data Science and Engineering*, 2(2):136–150, 2017.
- [45] D. Geiger and M. Schader. Personalized task recommendation in crowdsourcing information systems - current state of the art. *Decision Support Systems*, 65:3–16, 2014.
- [46] J. Golbeck and J. A. Hendler. Inferring binary trust relationships in web-based social networks. *ACM Trans. Internet Techn.*, 6(4):497–529, 2006.
- [47] J. Golbeck, B. Parsia, and J. A. Hendler. Trust networks on the semantic web. In *Cooperative Information Agents VII, 7th International Workshop, CIA 2003, Helsinki, Finland, August 27-29, 2003, Proceedings*, pages 238–249, 2003.
- [48] M. F. Goodchild and J. A. Glennon. Crowdsourcing geographic information for disaster response: a research frontier. *Int. J. Digital Earth*, 3(3):231–241, 2010.
- [49] E. Gray, J. Seigneur, Y. Chen, and C. D. Jensen. Trust propagation in small worlds. In *Trust Management, First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28-30, 2002, Proceedings*, pages 239–254, 2003.
- [50] S. Greengard. Following the crowd. *Commun. ACM*, 54(2):20–22, 2011.
- [51] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 403–412, 2004.
- [52] J. P. Guilford. The nature of human intelligence. 1967.



- 
- [53] A. Hagedorn and A. Pinkwart. The financing process of equity-based crowdfunding: An empirical analysis. In *Crowdfunding in Europe*, pages 71–85. Springer, 2016.
- [54] R. Hardin. *Trust and trustworthiness*. Russell Sage Foundation, 2002.
- [55] M. Hirth, T. Hoßfeld, and P. Tran-Gia. Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In *Proceedings of the Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2011, Seoul, Korea, June 30-July 02, 2011*, pages 316–321, 2011.
- [56] M. Hoogendoorn, S. W. Jaffry, and J. Treur. Exploration and exploitation in adaptive trust-based decision making in dynamic environments. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2010, Toronto, Canada, August 31 - September 3, 2010*, pages 256–260, 2010.
- [57] J. Howe. ‘crowdsourcing: A definition’, crowdsourcing: Tracking the rise of the amateur. [http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing\\_a.html](http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html).
- [58] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.
- [59] J. Huang, F. Nie, H. Huang, Y. Lei, and C. H. Q. Ding. Social trust prediction using rank-k matrix recovery. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2647–2653, 2013.
- [60] J. E. Hupcey, J. Penrod, J. M. Morse, and C. Mitcham. An exploration and advancement of the concept of trust. *Journal of advanced nursing*, 36(2):282–293, 2001.

- 
- [61] P. G. Ipeirotis. Be a top mechanical turk worker: You need \$5 and 5 minutes. *Blog: Behind Enemy Lines*, 2010.
- [62] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.
- [63] R. Ismail and A. Jøsang. The beta reputation system. In *15th Bled eConference: eReality: Constructing the eEconomy, June 17-19, 2002*, page 41, 2002.
- [64] S. Jagathula, L. Subramanian, and A. Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2492–2500, 2014.
- [65] M. Jamali and M. Ester. *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 397–406, 2009.
- [66] D. Jia, F. Zhang, and S. Liu. A robust collaborative filtering recommendation algorithm based on multidimensional trust model. *JSW*, 8(1):11–18, 2013.
- [67] S. Jones and P. Morris. Trust-ec: requirements for trust and confidence in e-commerce. *European Commission, Joint Research Centre*, pages 81–87, 1999.
- [68] A. Jøsang. Artificial reasoning with subjective logic. In *Proceedings of the second Australian workshop on commonsense reasoning*, volume 48, page 34. Perth:[sn], 1997.
- [69] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.

- 
- [70] A. Jøsang, E. Gray, and M. Kinatader. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.
- [71] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.
- [72] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.
- [73] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 1953–1961, 2011.
- [74] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008*, pages 453–456, 2008.
- [75] S. Konomi, W. Ohno, T. Sasao, and K. Shoji. A context-aware approach to microtasking in a public transport environment. In *Communications and Electronics (ICCE), 2014 IEEE Fifth International Conference on*, pages 498–503. IEEE, 2014.
- [76] K. Konrad, G. Fuchs, and J. Barthel. Trust and electronic commerce - more than a technical problem. In *The Eighteenth Symposium on Reliable Distributed Systems, SRDS 1999, Lausanne, Switzerland, October 19-22, 1999, Proceedings*, pages 360–365, 1999.
- [77] R. M. Kramer. Trust and distrust in organizations: Emerging perspectives, enduring questions. *Annual review of psychology*, 50(1):569–598, 1999.

- 
- [78] S. C. Kuek, C. Paradi-Guilford, T. Fayomi, S. Imaizumi, P. Ipeirotis, P. Pina, and M. Singh. The global opportunity in online outsourcing. 2015.
- [79] A. P. Kulkarni, M. Can, and B. Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *CSCW '12 Computer Supported Cooperative Work, Seattle, WA, USA, February 11-15, 2012*, pages 1003–1012, 2012.
- [80] M. Lease and E. Yilmaz. Crowdsourcing for information retrieval: introduction to the special issue. *Information retrieval*, 16(2):91–100, 2013.
- [81] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [82] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg. Signed networks in social media. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*, pages 1361–1370, 2010.
- [83] R. J. Lewicki and B. B. Bunker. Trust in relationships. *Administrative Science Quarterly*, 5:583–601, 1995.
- [84] L. Li and Y. Wang. A trust vector approach to service-oriented applications. In *2008 IEEE International Conference on Web Services (ICWS 2008), September 23-26, 2008, Beijing, China*, pages 270–277, 2008.
- [85] L. Li and Y. Wang. Subjective trust inference in composite services. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, pages 1377–1384, 2010.
- [86] G. Liu, Y. Wang, and M. A. Orgun. Social context-aware trust network discovery in complex contextual social networks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012.

- 
- [87] G. Liu, Y. Wang, M. A. Orgun, and E. Lim. A heuristic algorithm for trust-oriented service provider selection in complex social networks. In *2010 IEEE International Conference on Services Computing, SCC 2010, Miami, Florida, USA, July 5-10, 2010*, pages 130–137, 2010.
- [88] G. Liu, Y. Wang, M. A. Orgun, and E. Lim. Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks. *IEEE Trans. Services Computing*, 6(2):152–167, 2013.
- [89] X. Liu, H. He, and J. S. Baras. Crowdsourcing with multi-dimensional trust. In *18th International Conference on Information Fusion, FUSION 2015, Washington, DC, USA, July 6-9, 2015*, pages 574–581, 2015.
- [90] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. CDAS: A crowdsourcing data analytics system. *PVLDB*, 5(10):1040–1051, 2012.
- [91] I. Lotosh, T. Milo, and S. Novgorodov. Crowdplanr: Planning made easy with crowd. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 1344–1347. IEEE, 2013.
- [92] N. Luhmann. Trust and power. 1982.
- [93] R. Mansell and B. S. Collins. *Trust and crime in information societies*. Edward Elgar, 2005.
- [94] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman. Developer recommendation for crowdsourced software development tasks. In *2015 IEEE Symposium on Service-Oriented System Engineering, SOSE 2015, San Francisco Bay, CA, USA, March 30 - April 3, 2015*, pages 347–356, 2015.
- [95] S. P. Marsh. Formalising trust as a computational concept. *PhD Thesis*.

- 
- [96] M. G. Martinez. Inspiring crowdsourcing communities to create novel solutions: Competition design and the mediating role of trust. *Technological Forecasting and Social Change*, 117:296–304, 2017.
- [97] A. J. Mashhadi and L. Capra. Quality control for real-time ubiquitous crowdsourcing. In *Proceedings of the 2nd international workshop on Ubiquitous crowdsourcing*, pages 5–8. ACM, 2011.
- [98] W. Mason and S. Suri. Conducting behavioral research on amazons mechanical turk. *Behavior research methods*, 44(1):1–23, 2012.
- [99] R. McCann, W. Shen, and A. Doan. Matching schemas in online communities: A web 2.0 approach. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México*, pages 110–119, 2008.
- [100] D. H. McKnight and N. L. Chervany. *The meanings of trust*. 1996.
- [101] S. Mirri, C. Prandi, and P. Salomoni. A context-aware system for personalized and accessible pedestrian paths. In *International Conference on High Performance Computing & Simulation, HPCS 2014, Bologna, Italy, 21-25 July, 2014*, pages 833–840, 2014.
- [102] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *ACM DL*, pages 195–204, 2000.
- [103] L. Mui. *Computational models of trust and reputation: Agents, evolutionary games, and social networks*. PhD thesis, Massachusetts Institute of Technology, 2002.
- [104] V. Muñoz, J. Murillo, B. López, and D. Busquets. Strategies for exploiting trust models in competitive multi-agent systems. In *MATES*, pages 79–90. Springer, 2009.

- 
- [105] V. Naroditskiy, I. Rahwan, M. Cebrian, and N. R. Jennings. Verification in referral-based crowdsourcing. *PloS one*, 7(10):e45924, 2012.
- [106] W. Nejdl, D. Olmedilla, and M. Winslett. Peertrust: Automated trust negotiation for peers on the semantic web. In *Workshop on Secure Data Management*, pages 118–132. Springer, 2004.
- [107] B. C. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications magazine*, 32(9):33–38, 1994.
- [108] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 201–210, 2007.
- [109] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on amazon mechanical turk. 2010.
- [110] E. Peer, J. Vosgerau, and A. Acquisti. Reputation as a sufficient condition for data quality on amazon mechanical turk. *Behavior research methods*, 46(4):1023–1031, 2014.
- [111] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710, 2014.
- [112] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*, pages 385–388, 2009.
- [113] D. Povey. Developing electronic trust policies using a risk management model. In *Secure Networking - CQRE (Secure) '99, International Exhibition and*

- 
- Congress Düsseldorf, Germany, November 30 - December 2, 1999, Proceedings*, pages 1–16, 1999.
- [114] L. Qu, Y. Wang, M. A. Orgun, L. Liu, H. Liu, and A. Bouguettaya. Cccloud: Context-aware and credible cloud service selection based on subjective assessment and objective assessment. *IEEE Trans. Services Computing*, 8(3):369–383, 2015.
- [115] V. C. Raykar and S. Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13:491–518, 2012.
- [116] M. T. Ribeiro, S. Singh, and C. Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.
- [117] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.
- [118] J. B. Rotter. Generalized expectancies for interpersonal trust. *American psychologist*, 26(5):443, 1971.
- [119] J. M. Rzeszutarski and A. Kittur. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 16-19, 2011*, pages 13–22, 2011.
- [120] J. Samek and F. Zboril. Hierarchical model of trust in contexts. In *Networked Digital Technologies - Second International Conference, NDT 2010, Prague, Czech Republic, July 7-9, 2010. Proceedings, Part II*, pages 356–365, 2010.



- 
- [121] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5(1/2):115–153, 2001.
- [122] S. Schnitzer, C. Rensing, S. Schmidt, K. Borchert, M. Hirth, and P. Tran-Gia. Demands on task recommendation in crowdsourcing platforms-the workers perspective. In *ACM RecSys 2015 CrowdRec Workshop, Vienna, 2015*.
- [123] A. B. Seligman. *The problem of trust*. Princeton University Press, 2000.
- [124] W. Sherchan, S. Nepal, and C. Paris. A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):47, 2013.
- [125] H. Simula. The rise and fall of crowdsourcing? In *46th Hawaii International Conference on System Sciences, HICSS 2013, Wailea, HI, USA, January 7-10, 2013*, pages 2783–2791, 2013.
- [126] N. Stefanovitch, A. Alshamsi, M. Cebrian, and I. Rahwan. Error and attack tolerance of collective problem solving: The darpa shredder challenge. *EPJ Data Science*, 3(1):13, 2014.
- [127] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [128] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou. Crowdplanner: A crowd-based route recommendation system. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1144–1155. IEEE, 2014.
- [129] P. Sztompka. *Trust: A sociological theory*. Cambridge University Press, 1999.
- [130] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 990–998, 2008.

- 
- [131] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen. Online mobile micro-task allocation in spatial crowdsourcing. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 49–60, 2016.
- [132] M. Venanzi, A. Rogers, and N. R. Jennings. Trust-based fusion of untrustworthy information in crowdsourcing applications. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, pages 829–836, 2013.
- [133] P. Victor, C. Cornelis, and M. De Cock. *Trust networks for recommender systems*, volume 4. Springer Science & Business Media, 2011.
- [134] L. von Ahn and L. Dabbish. ESP: labeling images with a computer game. In *Knowledge Collection from Volunteer Contributors, Papers from the 2005 AAAI Spring Symposium, Technical Report SS-05-03, Stanford, California, USA, March 21-23, 2005*, pages 91–98, 2005.
- [135] C. Vondrick, D. J. Patterson, and D. Ramanan. Efficiently scaling up crowd-sourced video annotation - A set of best practices for high quality, economical video labeling. *International Journal of Computer Vision*, 101(1):184–204, 2013.
- [136] J. B. P. Vuurens and A. P. de Vries. Obtaining high-quality relevance judgments using crowdsourcing. *IEEE Internet Computing*, 16(5):20–27, 2012.
- [137] K. Wang, X. Qi, L. Shu, D.-j. Deng, and J. J. Rodrigues. Toward trustworthy crowdsourcing in the social internet of things. *IEEE Wireless Communications*, 23(5):30–36, 2016.
- [138] Y. Wang, L. Li, and G. Liu. Social context-aware trust inference for trust enhancement in social network based recommendations on service providers. *World Wide Web*, 18(1):159–184, 2015.

- 
- [139] Y. Wang and E. Lim. The evaluation of situational transaction trust in e-service environments. In *2008 IEEE International Conference on e-Business Engineering, ICEBE 2008, Xi'an, China, October 22-24, 2008*, pages 265–272, 2008.
- [140] Y. Wang, K. Lin, D. S. Wong, and V. Varadharajan. Trust management towards service-oriented applications. *Service Oriented Computing and Applications*, 3(2):129–146, 2009.
- [141] Y. Wang and V. Varadharajan. Role-based recommendation and trust evaluation. In *9th IEEE International Conference on E-Commerce Technology (CEC 2007) / 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (EEE 2007), 23-26 July 2007, National Center of Sciences, Tokyo, Japan*, pages 278–288, 2007.
- [142] W. Wei, F. Xu, C. C. Tan, and Q. Li. Sybildefender: A defense mechanism for sybil attacks in large social networks. *IEEE Trans. Parallel Distrib. Syst.*, 24(12):2492–2502, 2013.
- [143] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 2035–2043, 2009.
- [144] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu. Negotiating trust in the web. *IEEE Internet Computing*, 6(6):30–37, 2002.
- [145] S. E. Woo, M. Keith, and M. A. Thornton. Amazon mechanical turk for industrial and organizational psychology: Advantages, challenges, and practical recommendations. *Industrial and Organizational Psychology*, 8(2):171–179, 2015.

- 
- [146] J. Yang and A. Bozzon. On the improvement of quality and reliability of trust cues in micro-task crowdsourcing (position paper). *arXiv preprint arXiv:1702.03385*, 2017.
- [147] X. Yang, Y. Guo, and Y. Liu. Bayesian-inference-based recommendation in online social networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):642–651, 2013.
- [148] B. Ye and Y. Wang. Crowdrec: Trust-aware worker recommendation in crowdsourcing environments. In *IEEE International Conference on Web Services, ICWS 2016, San Francisco, CA, USA, June 27 - July 2, 2016*, pages 1–8, 2016.
- [149] B. Ye, Y. Wang, and L. Liu. Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 121–128, 2015.
- [150] B. Ye, Y. Wang, and L. Liu. Crowddefense: A trust vector-based threat defense model in crowdsourcing environments. In *2017 IEEE International Conference on Web Services, ICWS 2017, Honolulu, HI, USA, June 25-30, 2017*, pages 245–252, 2017.
- [151] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA*, pages 3–17, 2008.
- [152] H. Yu, C. Miao, Z. Shen, and C. Leung. Quality and budget aware task allocation for spatial crowdsourcing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1689–1690. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

- 
- [153] H. Yu, Z. Shen, C. Miao, and B. An. Challenges and opportunities for trust management in crowdsourcing. In *2012 IEEE/WIC/ACM International Conferences on Intelligent Agent Technology, IAT 2012, Macau, China, December 4-7, 2012*, pages 486–493, 2012.
- [154] H. Yu, Z. Shen, C. Miao, and B. An. A reputation-aware decision-making approach for improving the efficiency of crowdsourcing systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1315–1316. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [155] M. Yuen, I. King, and K. Leung. A survey of crowdsourcing systems. In *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, pages 766–773, 2011.
- [156] M. Yuen, I. King, and K. Leung. Taskrec: A task recommendation framework in crowdsourcing systems. *Neural Processing Letters*, 41(2):223–238, 2015.
- [157] H. Zhang, Y. Wang, and X. Zhang. Transaction similarity-based contextual trust evaluation in e-commerce and e-service environments. In *IEEE International Conference on Web Services, ICWS 2011, Washington, DC, USA, July 4-9, 2011*, pages 500–507, 2011.
- [158] H. Zhang, Y. Wang, and X. Zhang. Efficient contextual transaction trust computation in e-commerce environments. In *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom, June 25-27, 2012*, pages 318–325, 2012.
- [159] H. Zhang, Y. Wang, and X. Zhang. A trust vector approach to transaction context-aware trust evaluation in e-commerce and e-service environments. In

- 
- 2012 Fifth IEEE International Conference on Service-Oriented Computing and Applications (SOCA), Taipei, Taiwan, December 17-19, 2012*, pages 1–8, 2012.
- [160] H. Zhang, Y. Wang, X. Zhang, and E. Lim. Reputationpro: The efficient approaches to contextual transaction trust computation in e-commerce environments. *TWEB*, 9(1):2:1–2:49, 2015.
- [161] Z. Zhang and K. Wang. A trust model for multimedia social networks. *Social Netw. Analys. Mining*, 3(4):969–979, 2013.
- [162] Y. Zhao and Q. Zhu. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers*, 16(3):417–434, 2014.
- [163] C. Zhu, H. Nicanfar, V. C. Leung, and L. T. Yang. An authenticated trust and reputation calculation and management system for cloud and sensor networks integration. *IEEE Transactions on Information Forensics and Security*, 10(1):118–131, 2015.
- [164] C. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.
- [165] L. G. Zucker. Production of trust: Institutional sources of economic structure, 1840–1920. *Research in organizational behavior*, 1986.