

A Crowdsourcing Recommendation that Considers the Influence of Workers

Zhifang Liao¹, Xin Xu¹, Peng Lan¹, Liu Yang¹, Yan Zhang² and Xiaoping Fan^{3,*}

¹School of Computer Science and Engineering, Central South University, Changsha, 410075, China

²Glasgow Caledonian University, Glasgow, G4 0BA, UK

³Hunan University of Finance and Economics, Changsha, 410205, China

*Corresponding Author: Xiaoping Fan. Email: Xpfan@csu.edu.cn

Received: 09 June 2020; Accepted: 04 July 2020

Abstract: In the context of the continuous development of the Internet, crowdsourcing has received continuous attention as a new cooperation model based on the relationship between enterprises, the public and society. Among them, a reasonably designed recommendation algorithm can recommend a batch of suitable workers for crowdsourcing tasks to improve the final task completion quality. Therefore, this paper proposes a crowdsourcing recommendation framework based on workers' influence (CRBI). This crowdsourcing framework completes the entire process design from task distribution, worker recommendation, and result return through processes such as worker behavior analysis, task characteristics construction, and cost optimization. In this paper, a calculation model of workers' influence characteristics based on the ablation method is designed to evaluate the comprehensive performance of workers. At the same time, the CRBI framework combines the traditional open-call task selection mode, builds a new task characteristics model by sensing the influence of the requesting worker and its task performance. In the end, accurate worker recommendation and task cost optimization are carried out by calculating model familiarity. In addition, for recommending workers to submit task answers, this paper also proposes an aggregation algorithm based on weighted influence to ensure the accuracy of task results. This paper conducts simulation experiments on some public datasets of AMT, and the experimental results show that the CRBI framework proposed in this paper has a high comprehensive performance. Moreover, CRBI has better usability, more in line with commercial needs, and can well reflect the wisdom of group intelligence.

Keywords: Crowdsourcing; recommendation framework; workers'; influence; worker recommendation; weighted voting

1 Introduction

Crowd-based cooperative computing is a current research hotspot in cloud computing and big data. Crowdsourcing, an important branch of collaborative intelligent computing, integrates computers of unknown mass on the Internet to accomplish tasks that individual computers alone cannot accomplish [1].



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The concept of crowdsourcing was proposed by Howe in June 2006 [2], and its main participants include task requesters (i.e., requesters) and task completers (i.e., workers), who are linked together through human intelligence tasks. Crowdsourcing is a problem solving strategy that can solve complex problems in various scenarios, such as audio annotation, image classification, and software development. It allows the crowd to complete tasks by accepting tasks and participate in solving complex problems. The task requester completes the work of distributing the crowdsourcing task, and the crowdsourcing worker selects, accepts, and submits the answer to the task. However, if too many tasks are distributed to workers, the workers will become bored with the tasks, and the quality of tasks will be reduced. Therefore, how to choose the right workers from a large crowd has become a key issue for crowdsourcing recommendation.

At present, crowdsourcing recommendations generally target workers' behavior analysis and task characteristic construction. For example, Guo et al. [3] proposed a recommendation mechanism based on interest collaboration that used the professional level and interests of workers as the bases for recommendation. Difallah et al. [4] generated recommendations based on the degree of matching between workers and task descriptions. However, these studies did not consider the recommendation process as a whole and ignored the control of recommendation costs and the treatment of recommendation results during the recommendation process. Furthermore, the ability weight measurement of crowdsourcing workers is a key factor in the rational distribution of crowdsourcing tasks. Therefore, identifying an effective indicator to evaluate the characteristics of workers and using this indicator to recommend workers is a key task in crowdsourcing. Zhong et al. [5] have studied this issue, and most related studies use workers' task accuracy and preferences as indicators. However, if the task accuracy rate of the workers is used as an index for capability evaluation, it is impossible to distinguish between new manual workers and skilled workers with the same accuracy. If the completion accuracy of workers on different tasks is considered, the complexity of tasks is ignored. Therefore, we propose a new indicator of workers 'influence, which takes into account workers' correct rate, error rate, and task complexity. It can not only evaluate the comprehensive characteristics of workers, but also distinguish the behavior differences between workers in more detail. At the same time, for the problem of recommending cold start, this paper proposes a worker recommendation algorithm based on open-call mode to perform worker recommendation and cost optimization. In addition, for the task answers submitted by workers, we propose a weighted voting algorithm based on influence to aggregate workers' answers to ensure the accuracy of task results. Therefore, this paper proposes a crowdsourcing recommendation framework based on the workers' influence (CRBI).

The main contributions of this paper are as follows:

1. Proposing a new evaluation index of worker influence and designing a calculation model of worker influence characteristics based on the ablation method to effectively evaluate the comprehensive performance of workers.
2. Proposing a crowdsourcing task recommendation algorithm based on open-call mode to achieve effective recommendation efficiency and optimize task costs.
3. Proposing a weighted voting algorithm based on worker influence that cooperates with the recommendation algorithm to achieve good performance.

The rest of this paper is organized as follows. Section 2 describes the existing recommendation methods. Section 3 introduces the system model of the CRBI framework. Section 4 introduces the specific process design of the CRBI framework. Section 5 analyzes the experimental results. Finally, Section 6 draws conclusions and discusses directions for future research.

2 Related Work

Crowdsourcing can effectively reduce the cost of innovation, break the boundaries of the source of innovation, fully develop the wisdom of the public, and increase public participation. The research topics in crowdsourcing mainly focuses on the decomposition of crowdsourcing tasks, spatial crowdsourcing, incentive mechanism, workers' behavior analysis, aggregation of crowdsourcing results, and task selection. Moreover, crowdsourcing, as a problem-solving strategy, is also continuously combined with blockchain, Internet of Things [6], cloud computing [7] and social network, which can better solve complex problems in various scenarios. Crowdsourcing is also widely used in teaching. Wang et al. [8] proposed a crowdsourcing-based framework for evaluating teaching quality in classrooms. Moreover, benefiting from the development of public crowdsourcing platforms, such as Amazon Machine Turk (AMT), Crowdfunder (CF), TopCoder, etc., the crowd can accept and solve tasks through the crowdsourcing platform. The research on crowdsourcing has focused on task result aggregation, incentive mechanisms, spatial crowdsourcing, task decomposition, and quality control. Although crowdsourcing has become a promising problem-solving model, there is a lack of overall design research on crowdsourcing processes. Therefore, designing an efficient and available crowdsourcing recommendation framework is important for crowdsourcing systems.

The research on crowdsourcing recommendation is based mainly on traditional recommendation [9], including probability matrix decomposition [10], content-based recommendation [11], collaborative filtering [12], and hybrid recommendation algorithms [13]. Collaborative filtering, which generates recommendations based on the similarity of workers or tasks, is one of the most commonly used algorithms in crowdsourcing recommendation. Safran et al. [14] proposed two top-N recommendation algorithms for crowdsourcing systems and combined them with algorithms such as minimum mean square error and Bayesian personalized ranking for optimization. Generally, workers participating in crowdsourcing have different abilities and interests. Kurup et al. [15] proposed a new task recommendation model for reward-based crowdsourcing applications, derived task-based and reward-based probabilities, and analyzed their effectiveness. In general, crowdsourcing recommendation systems are designed for a single stakeholder; Aldahri et al. [13] proposed a multiobjective recommendation model for the interests of crowdsourcing workers, requesters, and platforms, but the model suffers from the cold start problem. Wang et al. [16] used the negative exponential learning curve model to fit workers' skill upgrades, and eliminated the impact of task difficulty on worker scores through the task difficulty weighting algorithm, and proposed a recommendation framework based on the model. Considering the independent contributions of crowdsourcing developers, Ye et al. [17] proposed a teammate recommendation method for developers based on their expertise and preferences. In the application of software development, Yu et al. [18] considered the dynamic changes in the development capabilities of software crowdsourcing developers, and combined with task types and complexity, proposed a dynamic task assignment algorithm DUTA. Habib et al. [19] considered a developer's preference mode, technical popularity and winning probability in software crowdsourcing development in a competitive mode, and proposed a dynamic task recommendation system that combines task exploration and development.

3 System Model

Based on traditional crowdsourcing research, this paper proposes a CRBI frame-work that analyzes workers' influence characteristics and combines workers' task performance models and potential task models of new tasks.

As shown in Fig. 1, the CRBI framework is divided into three main steps: Calculation of worker influence, recommendation of workers for new tasks, and aggregation of results generated by recommended workers. The detailed process is as follows.

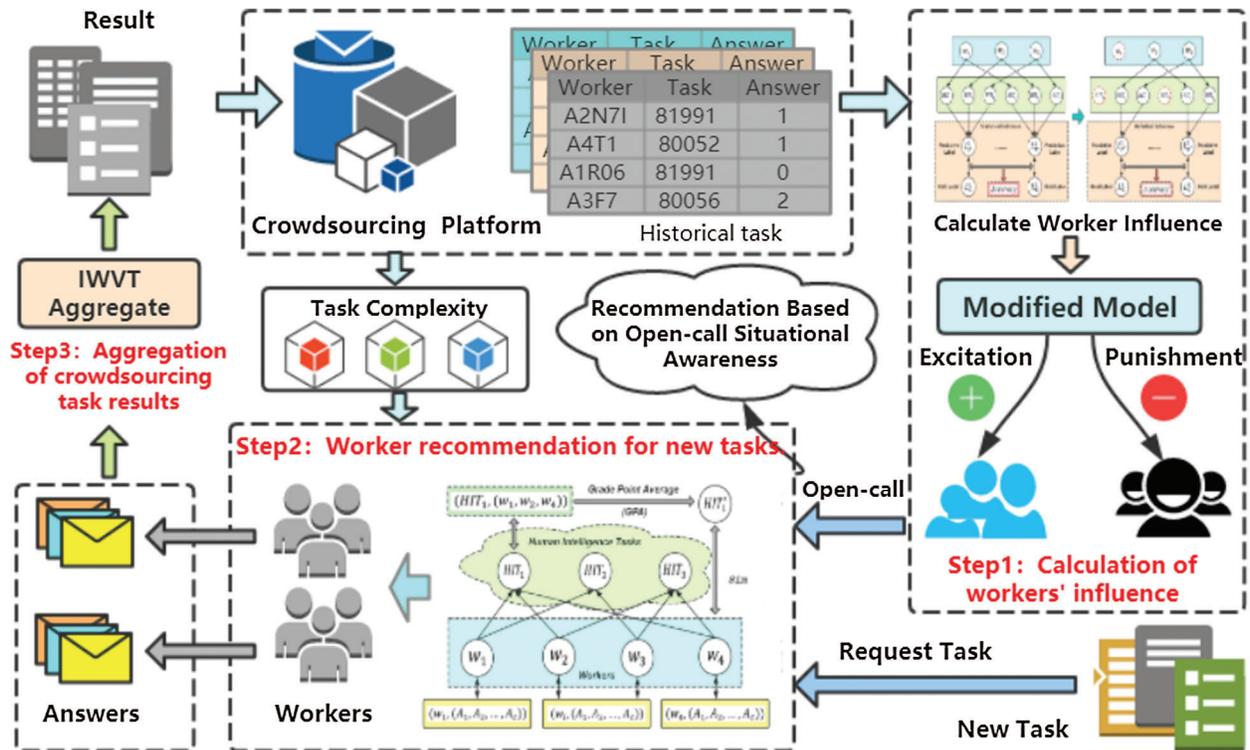


Figure 1: Design flow of the CRBI framework

Step 1: Calculation of workers' influence. First, the historical task information on the crowdsourcing platform is used to construct the initial influence of the workers and the accuracy of the workers in completing tasks. Simultaneously, a weight correction model is used to perform weight training on the initial influence to obtain the corrected worker influence, which is used as the worker weight to evaluate the overall performance of the worker.

Step 2: Worker recommendation for new tasks. Based on the traditional open-call task selection mode, worker influence and task complexity are combined to construct worker performance models for different tasks and to perceive and build characteristic models for new tasks. The recommended set of workers is obtained by calculating the similarity between workers' performance characteristics and new task characteristics. In addition, the task distribution quantity is controlled by fitting the change curve of task entropy to reduce the task cost of the requester, which is an optimization strategy of the CRBI framework.

Step 3: Aggregation of crowdsourcing task results. A weighted majority vote based on workers' influence (IWVT) is applied to the task answers generated by recommended workers to complete the result aggregation. Finally, the task results are returned to the crowdsourcing platform to complete the crowdsourcing process.

4 CRBI Framework

4.1 Definition of Symbols

This paper considers a crowdsourcing problem of n tasks and m workers, where workers choose a label for the task from z categories. The specific symbols used in this paper are defined in [Tab. 1](#).

Table 1: Symbol definitions

Symbol	Definition
n	Number of tasks
m	Number of workers
c	Number of task categories
H	$H = \{h_j \mid j \in \{1, \dots, n\}\}$ is the set of tasks
W	$W = \{w_i \mid i \in \{1, \dots, m\}\}$ is the set of workers
C	$C = \{c_k \mid k \in \{1, \dots, z\}\}$ is the set of categories
G	$G = \{g_j \mid j \in \{1, \dots, n\}\}$ is the truth set of task H
\tilde{Y}	$\tilde{Y} = \{\tilde{y}_{ij} \mid i \in \{1, \dots, m\}, j \in \{1, \dots, n\}\}$ is the set of worker answers. For example, \tilde{y}_{ij} represents the answer label of the j th random variable h_i for the i th worker.

4.2 Calculation of Workers' Influence

In contrast to the traditional workers' ability evaluation index based on the accuracy of task completion, this paper proposes a model of worker influence characteristics based on the ablation method. This model calculates the influence of aggregated workers' task results to obtain a measure of worker influence that serves as a new assessment indicator for the comprehensive ability of workers. The main process is shown in [Fig. 2](#).

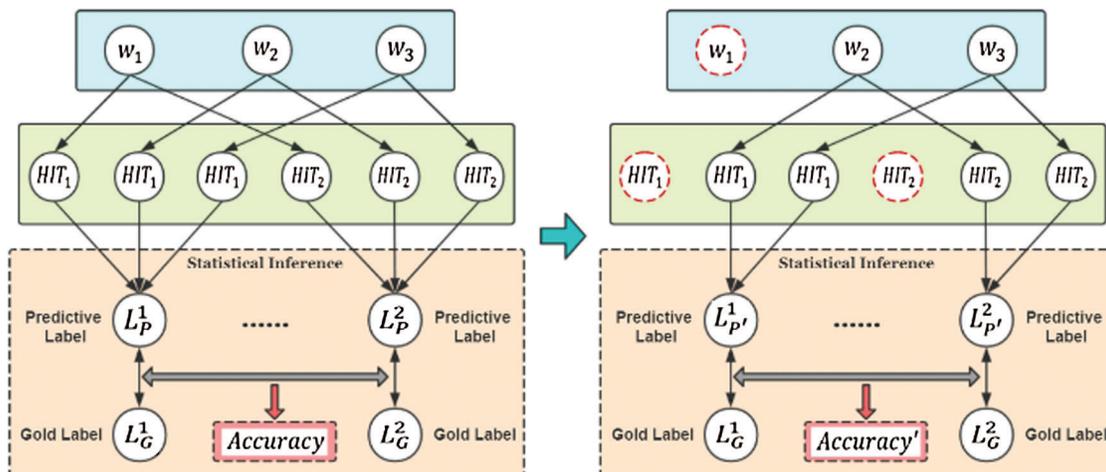


Figure 2: Workers' influence calculation model (The left is the process of the initial aggregation accuracy of the worker group; the right is the process of aggregation after excluding worker w_1 . The difference in accuracy is the degree of worker disturbance to the result aggregation)

Step 1: Calculate the aggregation accuracy of workers in historical tasks. The initial aggregation accuracy of the worker group is obtained as shown in Fig. 2 (left).

$$A = \frac{1}{nm} \sum_{i=0}^m \sum_{j=0}^n I(\tilde{y}_{ij} = g_j) \quad (1)$$

where $I(\cdot)$ is an indicator function; when $\tilde{y}_{ij} = g_j$, the function is 1.

Step 2: Eliminate the task data of worker w_j and recalculating the result aggregation accuracy again, as shown in Fig. 2 (right).

$$A_i = \frac{\sum_{i=0}^{m'} \sum_{j=0}^n I(\tilde{y}_{ij} = g_j)}{nm - \text{count}(w_i)} \quad (2)$$

where $m' = m - \text{count}(w_i)$ is the number of remaining tasks.

Step 3: Calculate the difference between A and A_i to obtain the degree of disturbance of the aggregation result by worker w_j .

$$\Delta A_i = A - A_i \quad (3)$$

where ΔA_i is the disturbance degree of the worker. If $\Delta A_i < 0$, the answer of w_i has a negative disturbance on the result; thus, the worker is classified as a low-quality worker, where ΔA_i indicates its low-quality degree. If $\Delta A_i > 0$, the answer of w_i has a positive disturbance on the result; thus, the worker is classified as a key worker, where ΔA_i indicates the importance. Furthermore, we map ΔA_i to $[0, 1]$ according to the sigmoid function, and the amplified influence is defined as follows:

$$I_i = \frac{1}{1 + e^{-k\Delta A_i}} \quad (4)$$

where k is a scaling factor. By means of I_i , the degree of disturbance of workers in historical tasks can be evaluated and used as the initial influence of workers.

Step 4: Construct a weight correction model to adjust the initial influence of workers based on the accuracy of workers' task completion.

$$F_i = \frac{1}{2} \alpha [\text{sign}(I_i) + 1] * I_i * A_i + \frac{1}{2} \beta [\text{sign}(I_i) - 1] * I_i * A_i \quad (5)$$

where α , β are correction parameters, α is a reward factor for the weight of key workers, and β is a penalty factor for the weight of low-quality workers to finely distinguish the differences between workers.

Step 5: Calculating the worker's influence R_i as the weight of the worker and using the result to evaluate the comprehensive ability of the worker.

$$R_i = \begin{cases} F_i, & I_i \neq 0 \\ A_i, & I_i = 0 \end{cases} \quad (6)$$

when $I_i = 0$, the influence of worker w_i on the final aggregation result tends to 0; thus, this worker is regarded as a general worker, and the task accuracy rate is used as the weight. When $I_i \neq 0$, worker w_i has a positive or negative influence on the final aggregation result, so the corrected influence F_i is taken as the weight.

4.3 Worker Recommendation Based on Open-Call Mode

In the real world, the crowdsourcing recommendation will have a cold start problem. To address this problem, this paper applies the traditional task selection mode and proposes a worker recommendation algorithm based on the open-call mode. The algorithm process is shown in Fig. 3. First, a worker’s weighted task performance model is constructed based on historical tasks; then, the characteristic model of the task applicant is analyzed to obtain the characteristic model of the new task. Finally, the similarity between the task model and the worker model is calculated to generate recommendations.

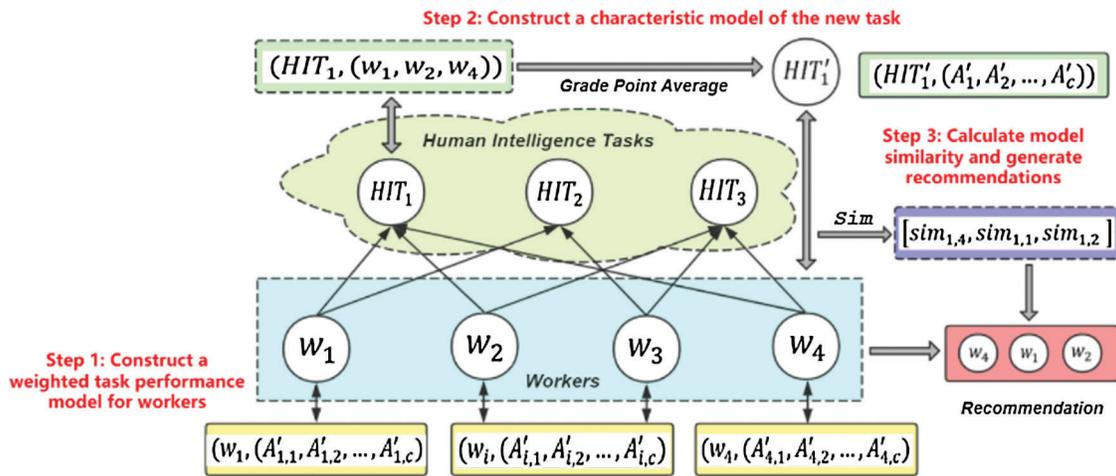


Figure 3: Flow of worker recommendation based on open-call mode

4.3.1 Construction of the Task Performance Model of Workers

Different types of tasks have different difficulties, even tasks of the same category have slight differences. This characteristic is often ignored when building worker models. Therefore, this paper builds a weighted task performance model based on task complexity to distinguish workers’ performance on different types of tasks.

Step 1: Calculate the entropy $H(h_j)$ of the worker answer for task h_j as the complexity of task h_j . That is, the greater the uncertainty in the worker’s answer to the task is, the higher the complexity of the task.

$$H(h_j) = -p_j \log p_j, \quad p_j = p(\tilde{y}_j = g_j | h_j) \quad (7)$$

where p_j is the probability that the worker completed task h_j correctly.

Step 2: Use the task complexity to weight the accuracy $A_{i,k}$ of worker w_i on each type of task to obtain the weighted accuracy $A'_{i,k}$ of the worker.

$$A'_{i,k} = Average(\sum_j^m H(h_j) * A_{i,k} | \tilde{y}_{ij} = g_j, h_j \in c_k) \quad (8)$$

4.3.2 Generation of Recommended Workers

Due to the lack of prior knowledge and descriptive information in crowdsourcing tasks, it is difficult for crowdsourcing platforms to obtain some relevant characteristics of new tasks. However, we can obtain the relevant characteristics of the task by mining the subtle links between the new task and the applicants. In this paper, we combine the traditional open-call task selection model and perceive the potential characteristics of a new task based on the worker’s task performance model and influence.

Step 1: Integrate the workers applying for a new task and build the connection between the new task and the worker ($HIT_1, (w_1, \dots, w_N)$). Then, use the workers' influence R_i to perform a weighted average of their task performance models to obtain a new task characteristic model ($HIT'_1, (A'_1, A'_2, \dots, A'_c)$).

$$A'_k = \frac{1}{AN} \sum_i^{AN} (A'_{i,k} * R_i) \quad (9)$$

where A'_k represents the confidence level of the new task as c_k and AN is the number of recommended workers.

Step 2: Calculate the similarity of the characteristic model ($HIT'_1, (A'_1, A'_2, \dots, A'_c)$) of the new task and the task performance characteristic model ($w_i, (A'_{i,1}, A'_{i,2}, \dots, A'_{i,c})$) of each worker to measure how well workers match the new task.

$$sim(h', w_i) = \frac{\sum_{k=1}^z ((w_i \rightarrow A_{i,k}) * (h' \rightarrow A'_k))}{\sqrt{\sum_{k=1}^z (w_i \rightarrow A_{i,k})^2} * \sqrt{\sum_{k=1}^z (h' \rightarrow A'_k)}} \quad (10)$$

Map the similarity vector of the worker group [$sim_{1,4}, sim_{1,1}, sim_{1,2}$] to the worker list to obtain the final recommendation list according to the number of workers set by the requester.

Step 3: Use an exponential curve to fit the trend of task entropy $E(x)$ to select the appropriate task recommendation number.

$$E(x) = \alpha * \exp(-\beta x) + \lambda \quad (11)$$

Let $E(x) = 0$ to obtain the minimum number of tasks for which the accuracy of the task converges, and use this value to control the number of task distributions. We attempt to obtain the best accuracy with a small number of tasks to improve the overall performance of the CRBI framework.

4.3.3 Aggregation of Task Results

After recommending suitable workers for new tasks, the crowdsourcing platform must also collect workers' task result tags. Aggregation rules must be used to infer the true results of the task from the results of the task with noise. The quality of the task results depends not only on the capabilities of the workers but also on the aggregation rules. Therefore, good aggregation rules are important.

In general, aggregation rules include algorithms based on majority voting and statistical methods. However, statistics-based aggregation entails considerable computational cost, while majority-based aggregation assigns different quality workers the same weight and produces inefficient results. Therefore, we propose a weighted voting aggregation algorithm based on workers' influence (WIVT) that uses worker influence as a weight for weighted voting to infer the trusted label of the task. The formula is as follows:

$$\hat{y}_j = \arg \max_{k \in \{1, \dots, z\}} \delta_{jk}, \quad icI_M \quad (12)$$

where \hat{y}_j is the label category of task h_j inferred by the algorithm and δ_{jk} represents the probability that task h_j belongs to category c_k . The formula is as follows:

$$\delta_{jk} = \frac{\sum_{i=0}^m R_i I(\tilde{y}_{ij} = c_k)}{\sum_{k=0}^z \sum_{i=0}^m R_i I(\tilde{y}_{ij} = c_k)} \quad (13)$$

In contrast to the traditional weighted majority voting algorithm, the WIVT algorithm takes into account workers' task completion accuracy in addition to workers' task amount, task complexity and other characteristics. The characteristics of workers covered by WIVT are more comprehensive. Moreover, the weight calculation process, weight optimization is performed to maximize the accuracy of the

aggregation. The WIVT algorithm can effectively alleviate problems such as voting ties and malicious voting while ensuring the accuracy of the aggregation.

5 Experiment

5.1 Dataset

The dataset involved in the experiment is shown in [Tab. 2](#). Each dataset is collected from the AMT platform, and the main attributes are worker ID, task ID, worker answer, and golden label (that is, the correct label provided by an expert).

Table 2: Dataset

Dataset	Judgments	Workers	Tasks	Labels
MG	2945	44	700	10
SP-2013	27746	203	5000	2
SP-2015	10000	143	500	2
ZC-India	11205	25	2040	2
ZC -USA	12190	74	2040	2

The MG dataset requires workers to listen to a 30-second song sample and classify it into one of 10 music types. The SP-2013 and SP-2015 datasets require workers to classify the polarity of sentences in movie reviews as positive or negative. The ZC-India and ZC-USA datasets contain links between entity names extracted from news articles and URIs describing the entities, requiring each worker to distinguish whether the URI is related or unrelated to the entity.

The experimental program is written mainly in Python. The execution environment is a Windows 10 operating system, and the CPU is an Intel i7 quad-core processor.

In the experiment, accuracy and RMSE are used as indicators to evaluate the recommendation results. Accuracy is used to measure the accuracy of recommended workers in completing a task, and RMSE is used to evaluate the accuracy with respect to the average cumulative residual. Avg-Precision, Avg-Recall, and F1-Measure are used as indicators to evaluate the task aggregation effectiveness of the recommended workers: Avg-Precision indicates the average accuracy of task result aggregation; Avg-Recall indicates the probability of a worker completing a certain type of task correctly; and F1-Measure is a metric that comprehensively considers accuracy and recall.

5.2 Experiment and Results Analysis

To illustrate the scientificity and effectiveness of the CRBI framework, the experiment aims to solve the following four problems:

Q1: How are workers' influence indicators structured? What is the effect?

Q2: How effective is the CRBI recommendation?

Q3: How is cost optimization implemented in the CRBI?

Q4: How to integrate the task answers submitted by recommended workers?

5.2.1 Construction and Evaluation of Workers' Influence

In this experiment, we first identified the existing problems by assessing the accuracy of the workers, then calculated the disturbance and selected the parameters to obtain workers' influence, and finally evaluated and analyzed the workers' influence indicators. We use 50% of the dataset as known historical data to construct and evaluate the worker influence characteristics. We obtain the distribution map of the completion accuracy of the workers on historical tasks via statistical analysis of the historical tasks of the workers (Fig. 4).

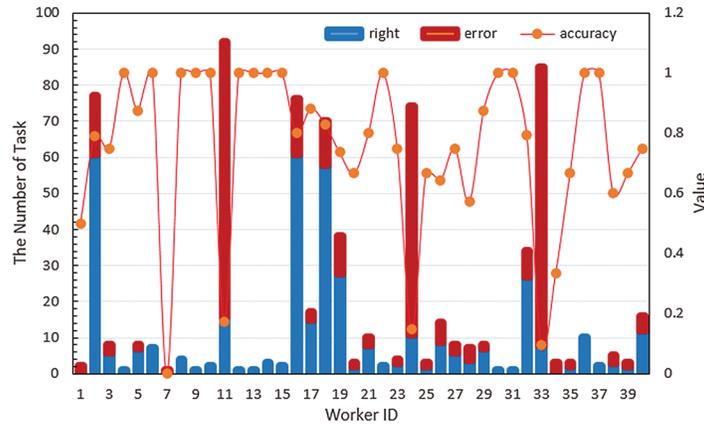


Figure 4: Characteristic distribution of workers in the MG dataset. (the line chart is the change in the accuracy of each worker; the histogram is the total number of workers' historical tasks, where the number of incorrect tasks is shown in red and the number of correct is shown in blue)

Fig. 4 shows that using the accuracy of workers as their weight leads to an excessively broad index, making it difficult for the platform to distinguish the behavioral differences between workers with the same accuracy. For example, in the Fig. 4, the worker with ID 36 completed 12 tasks correctly, while the worker with ID 4 completed 1 task correctly: the accuracy of both workers is 100%. Furthermore, the worker with ID 16 is correct in 61 of 76 tasks completed, while the worker with ID 32 is correct in 27 of 34 tasks, and both have an accuracy of approximately 80%. Clearly, the different workloads of the workers lead to differences in the accuracy assessment; therefore, accuracy is not an effective indicator of workers' ability. Moreover, workers' accuracy can be higher or lower depending on the difficulty of the task type (Fig. 5).

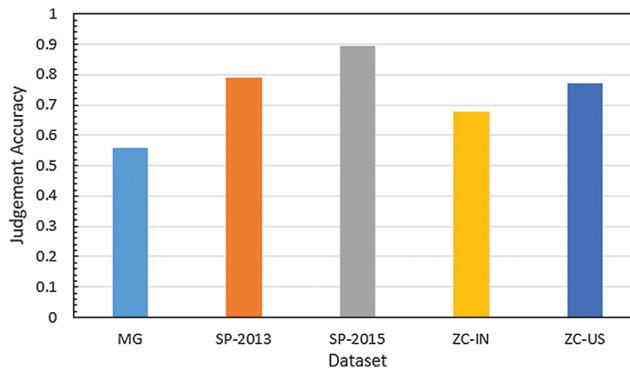


Figure 5: Average accuracy of each dataset

Therefore, in response to the above problems, we propose an indicator of worker influence to evaluate the comprehensive ability of workers. First, the disturbance of workers to the task aggregation result is calculated, as shown in Fig. 6, where the vertical axis is the degree of worker disturbance. When the value is positive, the worker has a promoting effect on the task; when the value is negative, the worker's answer has a negative effect on the task. When the value is 0, the worker does not play a key role in the task aggregation, that is, an ordinary worker. However, ordinary workers are the most common, which satisfies with the normal distribution of the population.

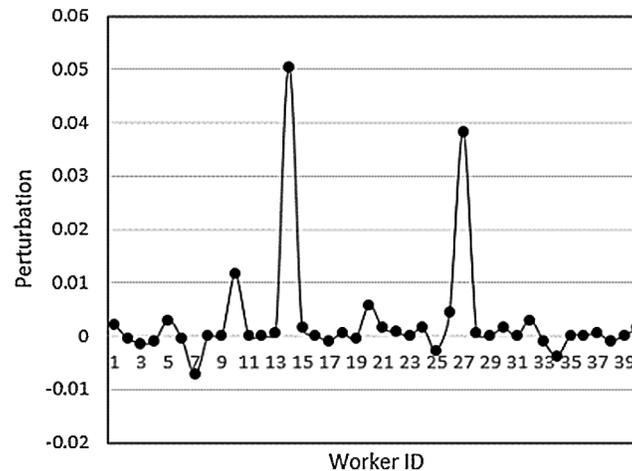


Figure 6: Distribution of worker disturbances

After the worker's disturbance value is calculated, the reward factor α and the penalty factor β are selected to perform weight correction. According to previous experiments and mathematical statistical analysis, we set the selection interval of the worker's reward factor α to $[1, 3]$ and the selection interval of the penalty factor β to $[0, 2.5]$. Then, the values in the interval are combined in pairs, and the aggregation accuracy of each combination is calculated to select the optimal parameter.

Fig. 7 shows the aggregation accuracy under different reward factors α and penalty factors β . The horizontal axis is the worker's reward factor α , the vertical axis is the worker's penalty factor β , and the value in the matrix is the aggregation accuracy of the task results. Then, these values are normalized. In Fig. 7 (left), the coordinates corresponding to a value of 1 are the optimal values of parameters α and β . As shown in Fig. 7 (right), if there are multiple coordinates with a value of 1, the historical data are randomly divided and aggregated multiple times, and parameters are selected from the intersection set of the matrices.

Fig. 8 shows that the indicator of worker influence objectively reflects the difference in ability among workers. For example, even though the worker with ID 36 has completed only 12 tasks, he receives a high weight because of the 100% accuracy. The worker with ID 16 has the same accuracy as the worker with ID 32, but the worker with ID 16 has a higher weight because of the greater number of completed tasks. Therefore, this indicator can distinguish between "new" and "experienced" workers, and there are few cases where the weights are the same, which aids in distinguishing among workers.

In addition, some workers have the same accuracy and are considered to have equal chances of being recommended. Moreover, dishonest workers may improve their accuracy through cheating. As shown in Fig. 9, when the accuracy rates of the workers are all 1.0, the accuracy rate index cannot be used to effectively distinguish the excellence degree of the workers. However, the influence index can be used to distinguish between homogeneous workers and reduce the weight of some low-quality workers.

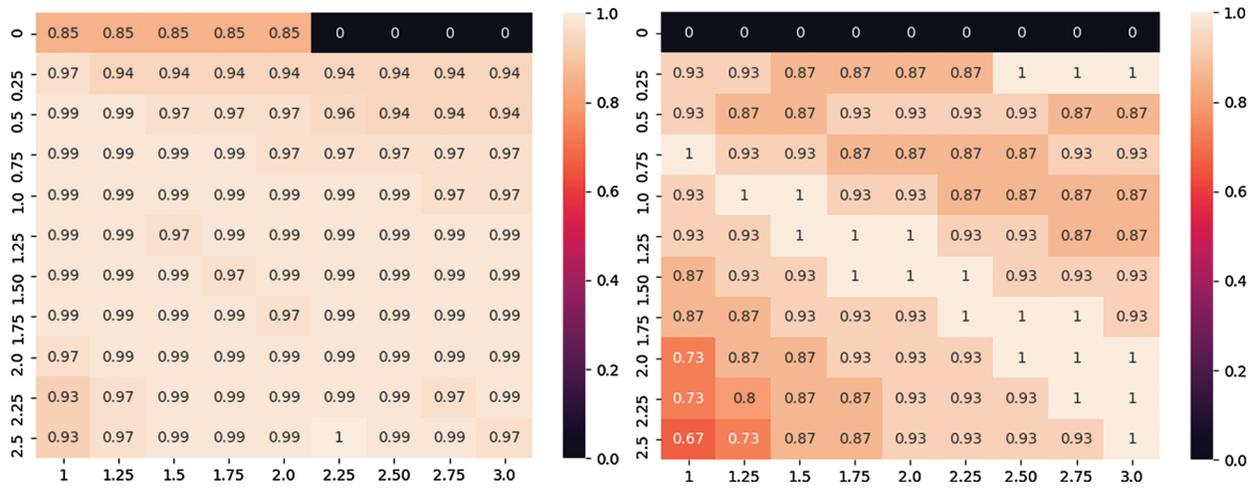


Figure 7: Aggregation accuracy under different parameters (the accuracy on the MS dataset on the left and the accuracy on the SP-2015 dataset on the right)

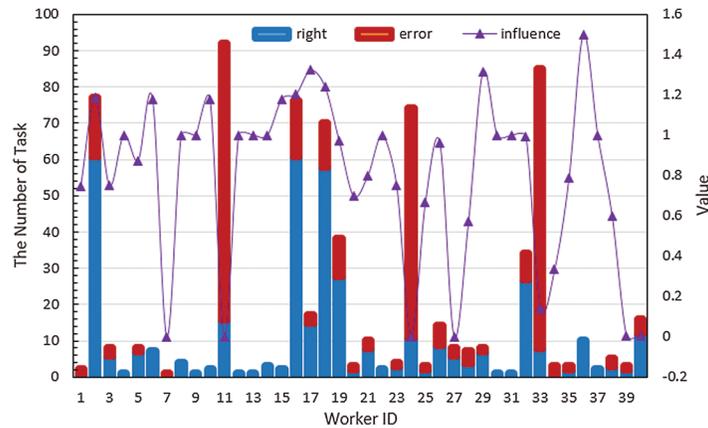


Figure 8: Distribution of workers' influence

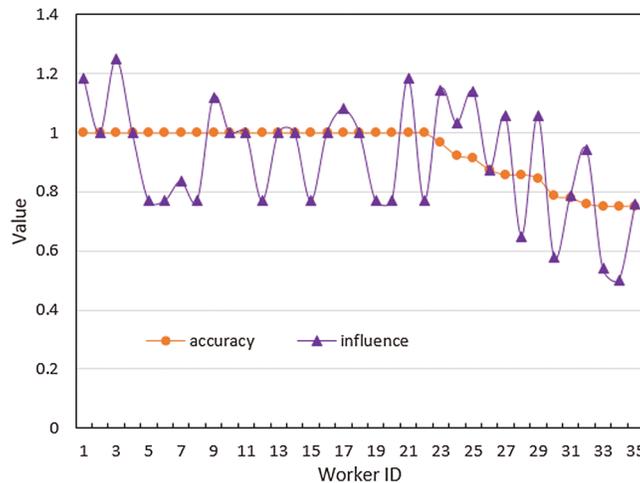


Figure 9: Changes in worker accuracy and influence on the ZC-USA dataset

5.2.2 Performance Evaluation of Recommendation Algorithms

To evaluate the recommendation performance of the CRBI framework, this experiment compares our recommendation algorithm with the following four recommendation methods: RBAcc, recommendation based on workers' task accuracy; RBCf, recommendation based on collaborative filtering; RBInf, recommendation based on workers' influence; Random, no recommendation is used (workers are randomly selected).

As shown in Fig. 10, under different numbers of recommended workers, CRBI performs better than the Random, RBAcc, RBCF, and RBFf algorithms by approximately 29%, 15%, 10%, and 11%, respectively, and the curve of the algorithm is more stable. Moreover, CRBI achieves better accuracy with a small number of workers.

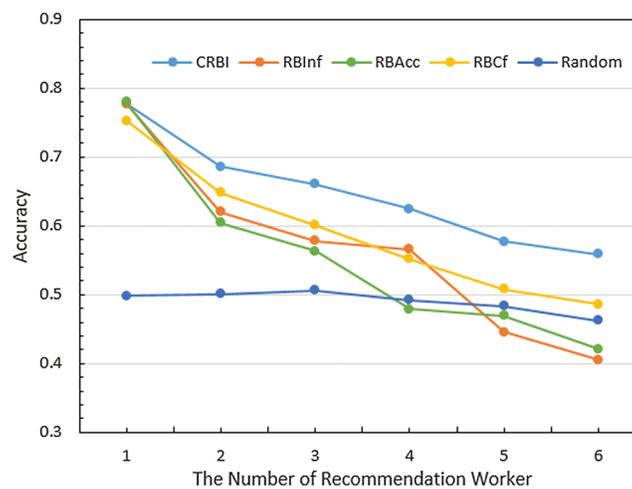


Figure 10: The accuracy of the recommended workers in completing the task with different numbers of recommended workers

However, since we use existing data for the recommendation simulation, as the number of recommenders increases, the number of available workers gradually decreases, and the set of recommended workers will tend to be the same. Therefore, the recommendation accuracy will gradually decrease. However, even in this case, our algorithm maintains an accuracy improvement of 10%~29% compared to the other algorithms.

In addition, to effectively evaluate the performance advantage of the recommendation algorithm in CRBI, we choose $N = 3$ (that is, the number of recommended workers for each task is 3). The task completion accuracy and the RMSE of the recommended worker are shown in Fig. 11.

As shown in Fig. 11 (left), compared to the Random algorithm without recommendation, our algorithm achieves an accuracy improvement of approximately 29%~39% by considering the workers' influence and task performance characteristics. Compared to the other recommendation algorithms, CRBI achieves different degrees of accuracy improvement. Moreover, Fig. 11 (right) shows that as the number of tasks increases, the accuracy and RMSE of the compared methods tend to converge. By contrast, the accuracy of our recommendation algorithm steadily improves throughout the entire process, and the RMSE is always low, indicating good performance.

Tab. 3 presents the recommendation results when the number of tasks is 20, 40, and 60. The accuracy of the recommendation algorithm based on the open-call mode in the CRBI framework is the highest and the MSE and RMSE are the lowest, indicating that our recommendation algorithm has good performance. In

addition, the RBInf algorithm performs better than the Random algorithm, indicating that the index of worker influence we constructed is beneficial in evaluation worker performance.

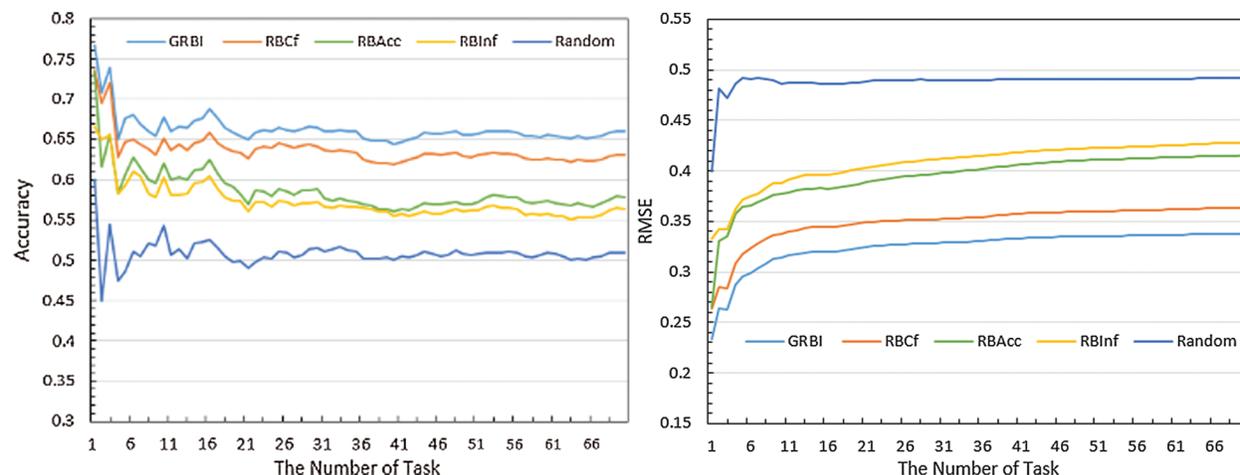


Figure 11: Accuracy and RMSE of the recommendation algorithm

Table 3: Evaluation of various recommendation algorithms under different task numbers

Task number	Evaluation	CRBI	RBCF	RBAcc	RBInf	Random
M = 20	Accuracy	0.6542	0.6333	0.5833	0.5733	0.5000
	MSE	0.3232	0.3479	0.3870	0.4017	0.4876
	RMSE	0.1044	0.1211	0.1498	0.1614	0.2377
M = 40	Accuracy	0.6442	0.6192	0.5608	0.5550	0.5008
	MSE	0.3329	0.3570	0.4056	0.4177	0.4906
	RMSE	0.1108	0.1275	0.1645	0.1745	0.2406
M = 60	Accuracy	0.6552	0.6272	0.5744	0.5583	0.5100
	MSE	0.3365	0.3616	0.4132	0.4251	0.4911
	RMSE	0.1133	0.1301	0.1707	0.1807	0.2412

In summary, the recommendation algorithm in the CRBI framework can greatly improve the accuracy of task completion, and it performs better than recommendation based on traditional ideas and characteristics.

5.2.3 Cost Optimization in Recommendation

In the general task distribution or recommendation process, the same number of workers will be recommended to complete each type of task. However, in the actual crowdsourcing process, if the number of tasks is insufficient, the accuracy of task completion will be low, whereas if too many tasks are distributed, the cost will increase cost and the accuracy of the tasks will be difficult to improve. Therefore, choosing an appropriate number of distributions for different tasks is key to optimizing the cost optimization.

In this experiment, we use three methods to fit the relationship between the number of tasks and task entropy to select the optimal task distribution volume for the recommendation algorithm to optimize the cost.

Fig. 12 shows that the exponential approximation matches the trend of task entropy, so we select the exponential curve. Due to the difficulty of different types of tasks and the number of tasks required to reach convergence, we use exponential curves to fit other types of tasks (Fig. 13).

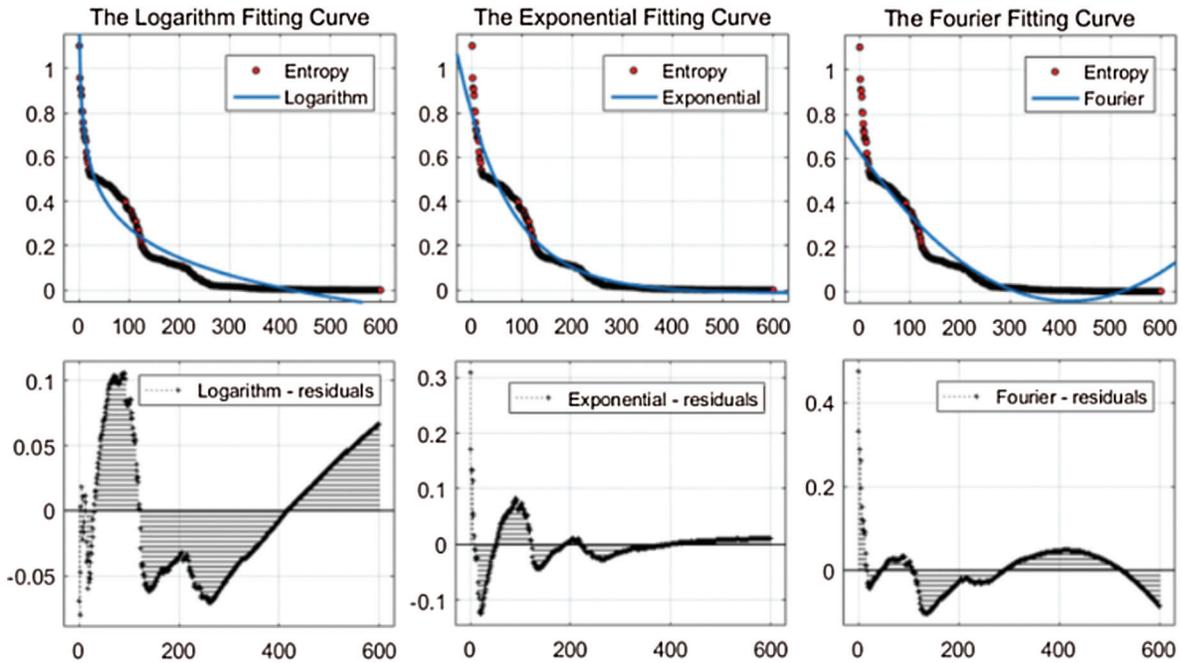


Figure 12: Fit between task number and task entropy

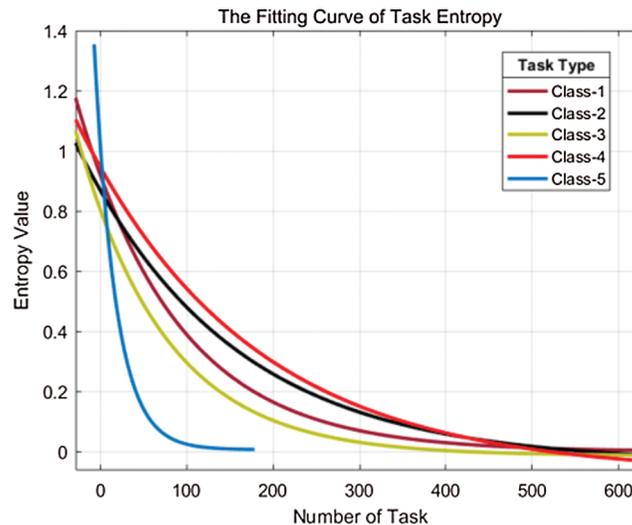


Figure 13: Entropy fitting curves

As shown in Fig. 13, as the number of tasks increases, the task results tend to stabilize. When the curve approaches 0, the tasks tends to converge, and the value of the abscissa at this time is the minimum number of tasks required for this type of task. When the task entropy converges, the accuracy of the task results cannot be improved even if the task continues to be distributed. These experiments illustrated that if the task

distribution is stopped when the task entropy converges, the cost can be reduced by 20~30%. Therefore, the task number selection method based on task entropy in the CRBI framework can effectively optimize the cost of crowdsourcing platforms.

5.2.4 Aggregation of the Task Results of Recommended Workers

Crowdsourcing can produce efficient output but also substantial noise, which leads to unreliable quality of the results. Therefore, the true value must be inferred from unreliable results in a short time. For the task answers submitted by workers, we use the WIVT in the CRBI framework to aggregate the results and compare them with the results of four other aggregation algorithms: Majority Vote (MV), Vote Distribution (VD) [20], Dawid & Skene (DS), and Bayesian (BCC) [21].

As shown in Fig. 14, the variation curve of the WIVT algorithm is always at a relatively high position, and the curve growth is stable, indicating that WIVT has the best stability. Furthermore, Tab. 4 indicates that at 50% of the data volume, the WIVT algorithm achieves a 9% performance improvement over the majority voting algorithm and a 6%, 4%, and 16% performance improvement over other aggregation algorithms, respectively. At 100% data volume, the WIVT algorithm achieves a performance improvement of approximately 10%, 8%, 1%, and 3% compared to the voting algorithm and the other aggregation algorithms.

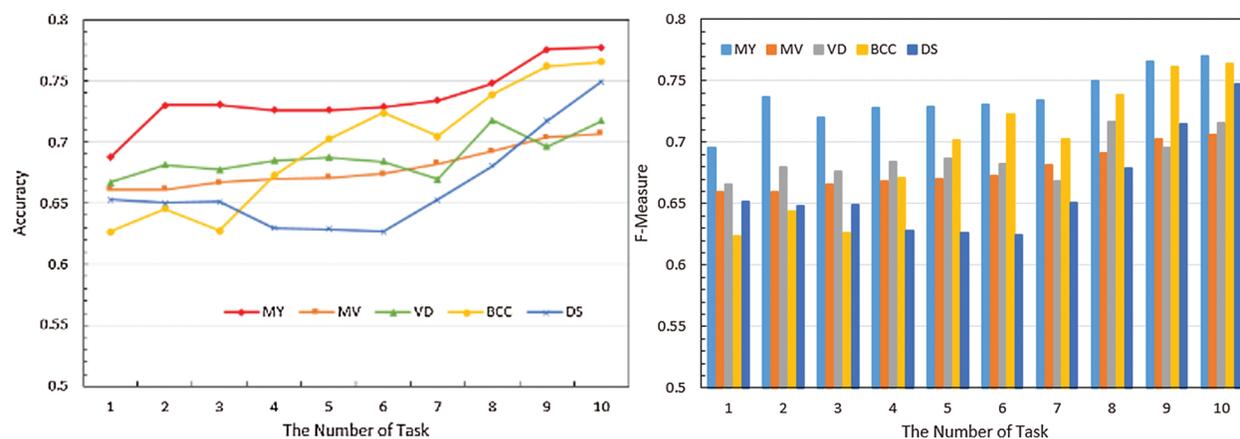


Figure 14: Aggregated results of various algorithms (accuracy line chart on the left and histogram of F1-measures on the right)

Table 4: Aggregation results of various algorithms

Algorithm	Recall		Precision		F1-measure		AVG-Time
	50%	100%	50%	100%	50%	100%	
MV	0.6686	0.7043	0.6709	0.7069	0.6697	0.7056	0.018
VD	0.6857	0.7143	0.6877	0.7069	0.6867	0.7157	0.015
DS	0.7000	0.7629	0.7025	0.7172	0.7012	0.7641	2.002
BCC	0.6243	0.7457	0.6288	0.7654	0.6265	0.7475	2.268
WIVT	0.7323	0.7633	0.7261	0.7772	0.7292	0.7702	0.042

The MV and VD algorithms assume that the task finishers are high-quality workers and do not distinguish between the voting weights of workers, which leads to inefficient aggregation results. The other two statistics-based methods, despite having good performance, have a substantial time cost. Because the workers are filtered by the recommendation algorithm, the recommended workers not only have better comprehensive ability but also a good matching degree with the new task, producing a good aggregation effect. In addition, the WIVT algorithm has a considerable advantage with respect to time: a 50 to 100 times improvement in time compared to the statistical method. In addition to ensuring time efficiency, the WIVT algorithm produces high-quality task results, which can maximize the application performance of the CRBI framework.

6 Conclusion

Crowdsourcing is a problem-solving mechanism with broad application prospects that can apply crowd intelligence to large-scale and complex tasks. Crowdsourcing recommendation is an important means to improve the quality of task completion. However, some traditional recommendation methods have certain shortcomings and ignore the control of task costs and the processing of task results generated by recommended workers. Therefore, we propose a crowdsourcing recommendation framework based on worker influence to systematically implement the entire crowdsourcing process. Among them, the worker's influence takes into account the worker's correct rate, error rate, and task complexity, which can more accurately assess the worker's characteristic performance. Moreover, the weighted voting based on workers' influence can also ensure the accuracy and safety of task aggregation. The CRBI framework proposed in this paper is used for the entire crowdsourcing recommendation process. From a user perspective, CRBI implements the entire process from task distribution to task answer acquisition; from a platform perspective, CRBI implements various processes, including worker behavior analysis, characteristic mining, worker recommendation, and task result aggregation. However, because this paper uses the classification tasks of the AMT platform for experiments, further experiments are needed for the suitability of other types of tasks. Moreover, the protection of workers' privacy and incentive mechanisms in the recommendation also need to be further improved.

In the future, we will consider using technologies such as big data to perform parallel calculations on massive crowdsourced data to improve the efficiency of the model. And use blockchain technology to optimize the crowdsourcing voting algorithm to ensure the security of the crowdsourcing voting process. In addition, we plan to conduct research on other types of crowdsourcing tasks, such as crowdsourcing software development, to design a more applicable recommendation framework.

Funding Statement: The works that are described in this paper are supported by Ministry of Science and Technology: Key Research and Development Project (2018YFB003800), Hunan Provincial Key Laboratory of Finance & Economics Big Data Science and Technology (Hunan University of Finance and Economics) 2017TP1025 and HNSF 2018JJ2535.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] P. Kucherbaev, F. Daniel, S. Tranquillini and M. Marchese, "Crowdsourcing processes: A survey of approaches and Opportunities," *IEEE Internet Computing*, vol. 20, no. 2, pp. 50–56, 2015.
- [2] J. Howe, "The rise of crowdsourcing," *Wired Magazine*, vol. 14, no. 6, pp. 1–4, 2006.
- [3] Z. Guo, C. Tang, W. Niu, Y. Fu, T. Wu *et al.*, "Fine-grained recommendation mechanism to curb astroturfing in crowdsourcing systems," *IEEE Access*, vol. 5, pp. 15529–15541, 2017.

- [4] D. E. Difallah, G. Demartini and P. Cudré-Mauroux, “Pick-a-crowd: Tell me what you like, and I will tell you what to do,” in *Proc. of the 22nd Int. Conf. on World Wide Web*, New York, NY, USA, pp. 367–374, 2013.
- [5] Q. Zhong, Y. Zhang, C. Li and Y. Li, “Task recommendation method based on workers’ interest and competency for crowdsourcing,” *System Engineering Theory and Practice*, vol. 37, pp. 3270–3280, 2017.
- [6] S. Lazarova-Molnar and N. Mohamed, “Collaborative data analytics for smart buildings: Opportunities and models,” *Cluster Computing*, vol. 22, no. 1, pp. 1065–1077, 2019.
- [7] M. Hosseini, C. M. Angelopoulos, W. K. Chai and S. Kunding, “Crowdcloud: A crowdsourced system for cloud infrastructure,” *Cluster Computing*, vol. 22, no. 2, pp. 455–470, 2019.
- [8] T. Wang, T. Wu, A. H. Ashrafzadeh and J. He, “Crowdsourcing-based framework for teaching quality evaluation and feedback using linguistic 2-tuple,” *Computers, Materials & Continua*, vol. 57, no. 1, pp. 81–96, 2018.
- [9] H. Hu, Y. Zheng, Z. Bao, G. Li, J. Feng *et al.*, “Crowdsourced poi labelling: Location-aware result inference and task assignment,” in *IEEE 32nd Int. Conf. on Data Engineering*, Helsinki, Finland, pp. 61–72, 2016.
- [10] M. C. Yuen, I. King and K. S. Leung, “TaskRec: A task recommendation framework in crowdsourcing systems,” *Neural Processing Letters*, vol. 41, no. 2, pp. 223–238, 2015.
- [11] K. Mao, Y. Yang, Q. Wang, Y. Jia and M. Harman, “Developer recommendation for crowdsourced software development tasks,” in *IEEE Sym. on Service-Oriented System Engineering*, San Francisco Bay, CA, USA, pp. 347–356, 2015.
- [12] M. Nilashi, O. Ibrahim and K. Bagherifard, “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques,” *Expert Systems with Applications*, vol. 92, pp. 507–520, 2018.
- [13] E. Aldahari, V. Shandilya and S. Shiva, “Crowdsourcing multi-objective recommendation system,” in *Companion Proc. of the Web Conf.* Republic and Canton of Geneva Switzerland, pp. 1371–1379, 2018.
- [14] M. Safran and D. Che, “Efficient learning-based recommendation algorithms for top-N tasks and top-N workers in large-scale crowdsourcing systems,” *ACM Transactions on Information Systems*, vol. 37, no. 1, pp. 1–46, 2018.
- [15] A. R. Kurup and G. P. Sajeev, “Task recommendation in reward-based crowdsourcing systems,” in *Int. Conf. on Advances in Computing, Communications and Informatics*, Udupi, India, pp. 1511–1518, 2017.
- [16] Z. Wang, H. Sun, Y. Fu and L. Ye, “Recommending crowdsourced software developers in consideration of skill improvement,” in *32nd IEEE/ACM Int. Conf. on Automated Software Engineering*, Urbana, IL, USA, pp. 717–722, 2017.
- [17] L. Ye, H. Sun, X. Wang and J. Wang, “Personalized teammate recommendation for crowdsourced software developers,” in *33rd IEEE/ACM Int. Conf. on Automated Software Engineering*, Montpellier, France, pp. 808–813, 2018.
- [18] D. Yu, Y. Wang and Z. Zhou, “Software crowdsourcing task allocation algorithm based on dynamic utility,” *IEEE Access*, vol. 7, pp. 33094–33106, 2019.
- [19] A. Habib, S. Hussain, A. A. Khan, M. K. Sohail, M. Ilahi *et al.*, “Knowledge based quality analysis of crowdsourced software development platforms,” *Computational and Mathematical Organization Theory*, vol. 25, no. 3, pp. 122–131, 2019.
- [20] W. Wang and Z. H. Zhou, “Crowdsourcing label quality: A theoretical analysis,” *Science China Information Sciences*, vol. 58, no. 11, pp. 1–12, 2015.
- [21] H. C. Kim and Z. Ghahramani, “Bayesian classifier combination,” in *Artificial Intelligence and Statistics*, La Palma, Canary Islands, pp. 619–627, 2012.