Joint modelling of task requirements and worker preferences based on heterogeneous features and multiple interactions for knowledge-intensive crowdsourcing recommendation

Biyu Yang and Xu Wang*

College of Mechanical and Vehicle Engineering,
Chongqing University,
Chongqing, 400044, China
Email: yangby123@outlook.com
Email: wx921@163.com
*Corresponding author

Shuai Zhang and Min Gao

School of Big Data and Software Engineering,
Chongqing University,
Chongqing, 400044, China
Email: zhangshuai@cqu.edu.cnt
Email: gaomin@cqu.edu.cn

Jiejie Tian

School of Data Science,
City University of Hong Kong,
Hong Kong, 999077, China
Email: 18914289890@163.com

Guangzhu Tan

Big Data Department,
Zhubajie Co. Ltd (ZBJ.COM),
Chongqing, 401120, China
Email: 229783860@qq.com

Linda Yang

School of Engineering,
University of Portsmouth,
Portsmouth, PO1 3HE, UK
Email: linda.yang@port.ac.uk

Jiafu Su

International College,
Krirk University,
Bangkok 10220, Thailand
Email: jiafu.su@hotmail.com

Abstract: Automatic worker recommendation has become a key technology in knowledge-intensive crowdsourcing (KIC). However, KIC recommendation encounters the task cold-start problem in nature as only newly posted tasks need to be matched with workers. Current studies fail to accurately model both tasks and workers in the task cold-start scenario, and ignore the problem of task clarity in task requirements understanding and treat task features linearly in worker preferences estimation. Therefore, this paper proposed a heterogeneous features and multiple interactions-based deep neural framework (called HFMIRec) to assist new task completion more smartly in KIC. Specifically, different types of task features can be flexibly incorporated to tackle the cold-start problem. To accurately model both tasks and workers, multiple interactions between tasks and workers are identified and learned by attentive neural networks in the framework. Extensive experiments on a real-world dataset demonstrate the effectiveness of the proposed model.

Keywords: crowdsourcing; task cold-start; worker recommendation; supply-demand matching; recommender system.

Reference to this paper should be made as follows: Yang, B., Wang, X., Zhang, S., Gao, M., Tian, J., Tan, G., Yang, L. and Su, J. (xxxx) 'Joint modelling of task requirements and worker preferences based on heterogeneous features and multiple interactions for knowledge-intensive crowdsourcing recommendation', *Int. J. Bio-Inspired Computation*, Vol. X, No. Y, pp.xxx–xxx.

Biographical notes: Biyu Yang is currently a PhD candidate in Management Science and Engineering at Chongqing University in China. Her current research interests focus on crowdsourcing recommendation.

Xu Wang is a Professor at the College of Mechanical and vehicle Engineering, Chongqing University, China. Her research interests focus on service science and system optimisation.

Shuai Zhang received his BSc from the School of Information and Electronics, Beijing Institute of Technology, China, and his MSs from the School of Software, Chongqing University, China.

Min Gao is a Professor at the School of Big Data and Software Engineering, Chongqing University, China. She also holds a PhD in Computer Science and Application from Chongqing University.

Jiejie Tian received his MSc in Data Science in City University of Hong Kong in 2022. He is currently working as a Research Assistant at City University of Hong Kong.

Guangzhu Tan is a Manager of Big Data Department of Zhubajie Co. Ltd. He is also pursuing his PhD at School of Big Data and Software Engineering, Chong University, China.

Linda Yang is currently a Senior Lecturer of Computer Science at the University of Portsmouth. Her research interests focus on machine learning and their applications.

Jiafu Su received his PhD in School of Mechanical Engineering from Chongqing University, Chongqing, China in 2017. His research interests cover innovation management, knowledge management and supply chain management.

1 Introduction

As an increasingly promising domain in today's knowledge-intensive economy, knowledge-intensive crowdsourcing (KIC) distributes knowledge-intensive tasks via open calls and requires workers to master essential skills and put in significant effort to complete the tasks (Howe, 2006; Gong, 2017; Zhang and Su, 2019; Yang et al., 2021; Zhang et al., 2022c). However, the mismatch between KIC tasks and workers leads to a high failure rate of tasks and the waste of workers' efforts, discouraging seekers and workers from engaging with crowdsourcing platforms in the long term (Geiger and Schader, 2014; Mo et al., 2018; Yin et al., 2020; Yuen et al., 2021). Developing effective recommender systems to conduct mutual matching between KIC tasks and workers and enhance their user experience is of paramount importance (Yang and Dutta, 2021).

However, building an effective and efficient KIC recommendation system is quite difficult as it poses the following major challenges:

1 The task cold-start problem: newly posted KIC tasks have to be matched with suitable workers within a short period of time and expired or completed KIC tasks do not need to recruit workers any more (Dai et al., 2020). This makes the worker recommendation for a new task each time. However, the traditional ID-based collaborative filtering methods such as matrix factorisation (MF) (Lin et al., 2014) and probabilistic matrix factorisation (PMF) (Yuen et al., 2015) are unable to make effective recommendations in task cold-start domain.

2 Accurate task modelling is an essential but challenging step in KIC recommendation (Fu et al., 2021; Yuen et al., 2021). Traditional statistical text representation methods, such as bag-of-words and topic models, can hardly capture hidden personalised task requirements from unstructured task descriptions (Zhu et al., 2015) as they fail to consider the word orders and the underlying semantics in the texts (Raza and Ding, 2022). Also, service seekers' strategical behaviours and limited professionalism leads to unstable presentation quality of task descriptions, which further influences the clear understanding of task requirements (Erat and Krishnan, 2012; Yang, 2019; Yin et al., 2020; Jiang et al., 2021). In addition, as KIC tasks are represented by a set of heterogeneous features, comprehensively incorporating heterogeneous features in task modelling is necessary to capture important task characteristics (Zhu et al., 2015).

3 Worker preferences estimation: workers usually pay attention to specific task features when selecting preferred KIC tasks and these features may have correlations (Zhang et al., 2022a). Workers' performance history is a useful source to mine workers' heterogeneous preferences (Gong, 2017; Zhang and Su, 2019; Zhang et al., 2020a). However, how to capture the dynamic weights of different task features and their nonlinear correlations based on workers' performance history still challenges the accurate worker preferences estimation in KIC recommendation.

Current crowdsourcing recommendation methods can be classified into content-based, collaborative filtering and hybrid methods. However, they failed to fully consider the above challenges in KIC, leading to poor recommendation accuracy and effectiveness. Recently, deep learning-based recommender systems have flourished in different applications (Liu et al., 2017; Wei et al., 2017; Zhang et al., 2019; Lakshmanan and Anand, 2022). However, there is insufficient research on deep learning-based recommendation technology in the KIC setting.

To comprehensively address the aforementioned challenges and fill these research gaps, we put forward a novel framework named heterogeneous features and multiple interactions-based deep neural network for cold-start task recommendation (HFMIRec) in the context of KIC. The proposed framework allows for the flexible incorporation of heterogeneous features of both tasks and workers, thus handling the task cold-start problem. Besides the commonly used textual task descriptions, arbitrary continuous and categorical features can be easily added to the model. To learn the semantic meaning hidden in the task description and worker profiles, deep learning-based text representation methods can be adopted to process these heterogeneous features. In the proposed model, two types of interactions, namely the task-task interaction indicating a relationship of requirements complementarity between a new task and its similar tasks and the worker-task interaction indicating the preference mapping relationship between a worker and his/her performance histories, are identified and captured by two attentive neural networks, respectively, which can adaptively assign weights to different features. Finally, the two attentive neural networks output the corresponding strengthened vector representations of both new tasks and workers, which are further used to obtain the matching degree between them.

The main contributions of this paper are summarised as follows:

1 We make an in-depth analysis of the cold-start problem in KIC worker recommendation for new task completion, which has not yet been sufficiently studied before.

2 The proposed framework can jointly model the hidden task requirements and worker preferences by flexibly incorporating different types of features and various interactions between tasks or between tasks and workers.

3 Extensive experiments on a real-world dataset and comparisons with the state-of-the-art recommendation approaches illustrate the effectiveness of the proposed approach in recommending potential workers to cold-start tasks in KIC.

The rest of this paper is structured as follows. Related research is described in Section 2. Section 3 introduces the framework of HFMIRec. Section 4 analyses and compares the results of experiments on a real-world dataset, and ablation studies are also conducted. Finally, Section 5 summarises this paper.

2 Literature review

Crowdsourcing recommendation methods can be grouped into three classes, i.e., content-based, collaborative filtering and hybrid methods (Geiger and Schader, 2014; Aldhahri et al., 2015; Ghezzi et al., 2018).

Content-based crowdsourcing recommendation systems rely on task descriptions to estimate the similarity among candidate tasks and workers' preferences. To this end, a set of text modelling methods were adopted to analyse task requirements from task descriptions, such as the bag-of-words approach (Ambati et al., 2011), term frequency – inverse document frequency (TF-IDF) (Fu et al., 2021), and so on. Even though content-based methods are easy to implement and can alleviate the task-cold start problem, they obtain textual similarity solely based on the lexical similarity (Zheng et al., 2017), failing to consider the underlying semantic in task descriptions (Raza and Ding, 2022). Besides, heterogeneous features are difficult to incorporate in content-based recommendation systems, resulting in the deficiency of important task characteristics. Some studies regarded the crowdsourcing recommendation as a classification problem (Ambati et al., 2011; Zhang et al., 2020b; Wang et al., 2021). However, it is hard to train an effective classifier when there is a small number of tasks in workers' historical records (Dai et al., 2020).

Purely collaborative filtering methods attend to recover the user-item interaction matrix based on users' explicit ratings on items. To overcome the lack of explicit feedback in crowdsourcing, Yuen et al. (2012, 2015) proposed to convert the positive interaction behaviours between workers and tasks to ratings. Then a unified PMF model was introduced to capture worker-task preferences, work-category preferences and task-category preferences. Active learning was further adopted to enhance recommendation efficiency in Yuen et al. (2016) and a time-aware task recommendation framework was developed to consider workers' dynamic preferences (Yuen et al., 2021). Lin et al. (2014) assigned different weights to task-worker interactions based on workers' positive and negative implicit feedback. Collaborative filtering methods like MF and PMF only use the IDs of both tasks and workers when making prediction, thus they cannot handle the task or worker cold-start problem.

Hybrid methods combine the advantages of content-based and collaborative filtering methods. Usually, content features are used to identify similar workers or tasks in hybrid methods, thus alleviating the cold-start and data sparsity problems. Therefore, traditional text modelling techniques commonly used in content-based methods, such as IT-IDF and topic-based models, were also adopted in hybrid methods (Wu et al., 2011; Yang et al., 2017).

Recently, neural network-based models have been adopted in crowdsourcing recommendations. Targeting the task cold-start problem, Dai et al. (2020) incorporated the task content features into a BPR-based recommender system to learn a latent representation of the new task. Worker ID was used to learn worker representations and worker profile was ignored. Yu et al. (2022) combined factorisation machine (FM), MF and multiple layer perception (MLP) to solve both task and developer cold-start problem. However, the modelling of cold-start tasks and developers is independent of scores. Therefore, there is no guarantee that the learned implicit features can be beneficial to the score prediction (Zheng et al., 2017).

Our work is essentially a neural network-based recommendation method. What differentiates our proposed recommender system from prior methods is that it can flexibly incorporate heterogeneous data of both tasks and workers and capture multifarious nonlinear relationships between tasks or between tasks and workers to jointly model the unstructured and personalised task requirements and estimate worker preferences in the task cold-start scenario. Moreover, the vectorisation of heterogeneous features and adaptive weight capturing parts in our framework is broadly defined and fine-tuned as desired depending on the specific application context.

3 Proposed model

3.1 Model framework

As shown in Figure 1, the proposed HFMIRec model consists of three main modules: a heterogeneous feature preprocessing (*FeaPre*) module, a task requirements modelling (*TaskMod*) module, and a worker preferences estimation (*WorkerEst*) module. The *FeaPre* module is a prerequisite for the other two modules, and the *TaskMod* and *WorkerEst* modules work in parallel.

The *FeaPre* module aims to learn the feature representation of t and w from their own heterogeneous features, respectively. The module first employs different methods to process the heterogeneous features of t and w to extract their respective feature vectors. These feature vectors of t and w are then concatenated and fed into the corresponding multiple layer perceptron (MLP) to obtain the feature representation of t and w, respectively. It is noted that the implementation of heterogeneous task features can handle the task cold-start problem.

The *TaskMod* module will further learn a more informative fused representation of t based on the task-task interactions. A prerequisite of the module is to identify complementary tasks that are most similar to t in terms of feature similarity, which will be detailed in Section 3.3. The most relevant information is extracted from the complementary tasks by applying a self-attention mechanism and then is fused into the feature representation of t to obtain the fused representation of t.

Figure 1 The framework of HFMIRec (see online version for colours)

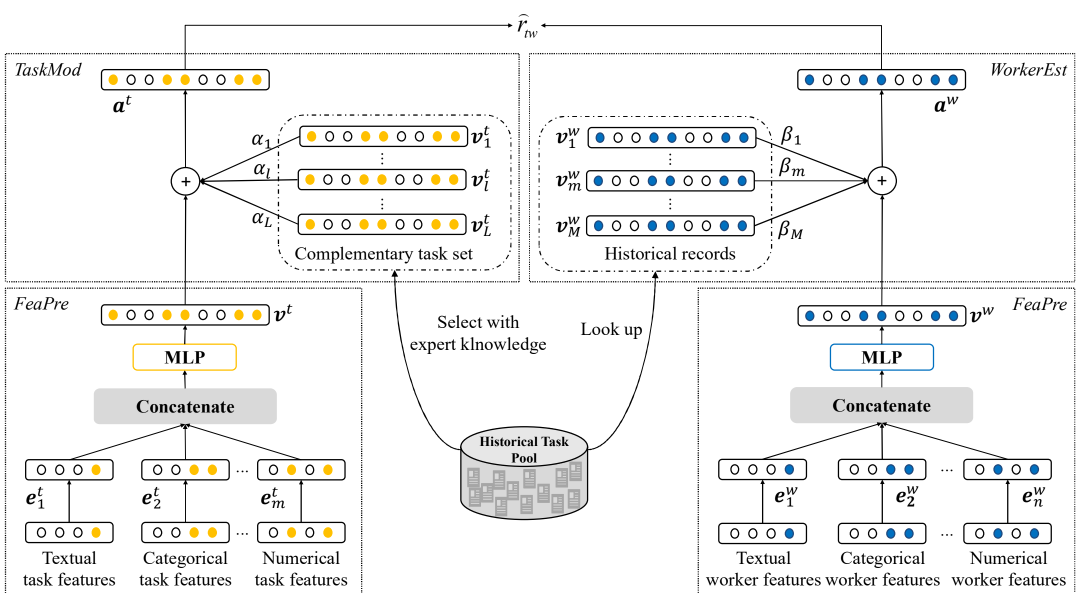

Table 1 KIC task features

Features	*Data format*	*Feature description*	*Processing method*
Task title	Textual	The title of the KIC task	BERT
Task description	Textual	The textual description of the introduction of the KIC task, service seeker's objectives and requirements, etc.	BERT
Task category	Categorical	The type of the KIC task	One-hot
Post location	Categorical	The geographical location where the KIC task is posted	One-hot
Budget	Numerical	The budget that the service seeker is willing to pay	Z-score standardisation
Execution period	Numerical	The execution period for workers to complete the task	Z-score standardisation

Table 2 Worker features

Features	*Data format*	*Feature description*	*Processing method*
Self-introduction	Textual	The self-introduction of the worker	BERT
Preferred task category	Categorical	The type of task that the worker is interested in or competent for	One-hot
Location	Categorical	The location where workers live or work at	One-hot
The length of time on the platform	Numerical	The length of time after registration on the platform	Z-score standardisation

The *WorkerEst* module will mine the preferences of w based on the worker-task interactions and output the fused representation of w. The registration information and the historical records of w are both utilised. To differentiate the contribution strength of different features, a self-attention mechanism is employed to adaptively assign weights to these features. Thus, the fused representation of w is a weighted combination of the feature representation of w and the tasks he/she has performed before.

3.2 Heterogeneous features preprocessing

The heterogeneous features of tasks and workers are multimodal, commonly including textual, categorical and numerical features. To learn their vector representations, different methods are employed to process these features. Tables 1 and 2 present the commonly used features of tasks and workers and the processing methods used, respectively. It is noted that other types of features and processing methods can be employed as desired, depending on the specific research or business objectives.

As for textual features, deep learning-based text embedding techniques, such as BERT (Devlin et al., 2019) and convolutional neural network (CNN) (Pereira et al., 2022; Wang et al., 2022) can be employed to transfer them into semantic-aware vectors, as they can capture and keep the underlying semantic information in textual data (Devlin et al., 2019). In this paper, BERT is selected. For each text, we select verbs, nouns and adjectives and remove the stop words in the text. Then BERT can map each word to a dense vector with a fixed size. The vector representation of the text is obtained by averaging the vectors of all words in the text.

To handle the categorical features, one-hot encoding or N-of-one encoding are employed. We then feed obtained vectors into the corresponding embedding layer to obtain their dense vectors, respectively.

Numerical features are relatively easier to handle. To eliminate the influence of different measurement units, the Z-score function is first adopted to normalise each number as shown by equation (1).

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where x is the feature value, μ is the average feature value of tasks, and σ is the variance of these feature values. After that, the normalised numbers are sent into an embedding layer to obtain the corresponding vector.

After obtaining the vector representations from heterogeneous features, we concatenate these vector representations and then send the vector into a MLP to generate the task feature representation $\boldsymbol{v}^t$ and the worker feature representation $\boldsymbol{v}^w$, respectively. $\boldsymbol{v}^t$ and $\boldsymbol{v}^w$ can be viewed as the primary modelling of task requirements and worker preferences, respectively. However, $\boldsymbol{v}^t$ may be less informative because of the possible missing details in the task description.

3.3 Task requirements modelling based on task-task interactions

Inspired by the common operation in GNN-based recommender systems where the features of neighbour nodes are incorporated to build an accurate representation of a target node (Hu et al., 2019; Hao et al., 2021; Zhang et al., 2022a), we incorporate the interactions between t and its similar tasks to further strengthen the expressiveness of $\boldsymbol{v}^t$. It is assumed that similar tasks may share similar requirements and the information of similar tasks can be a strong signal to infer the characteristics and requirements of a target task.

The similarity between tasks can be obtained by equation (2).

$$CP(t, t') = \frac{a_1 sim\left(e_t^{title}, e_{t'}^{title}\right) + a_2 sim\left(e_t^{dep}, e_{t'}^{dep}\right)}{a_3\left|f_t^{budget} - f_{t'}^{budget}\right| + a_4\left|f_t^{time} - f_{t'}^{time}\right|} \cdot x_{t,t'}$$
$$x_{t,t'} = \begin{cases} 1, & if\ f_t^{cate} = f_{t'}^{cate} \\ 0, & if\ f_t^{cate} \neq f_{t'}^{cate} \end{cases} \tag{2}$$

where t' is a completed task. e_t^{title}, e_t^{dep}, $e_{t'}^{title}$ $e_{t'}^{title}$ and $e_{t'}^{dep}$ are the respective vector representation of t and t' learned by BERT from the task title and task description, as discussed in Section 3.2. f_t^{budget}, $f_{t'}^{budget}$, f_t^{time} and $f_{t'}^{time}$ denote the task budget and execution period of t and t', respectively. $sim(\cdot)$ is a similarity function that measures the similarity between vectors. In this paper, cosine similarity is adopted. f_t^{cate} and $f_{t'}^{cate}$ means the task category of t and t'. If t and t' belong to the same task category, $x_{t,t'} = 1$. a_1, a_2, a_3 and a_4 are scale parameters for weighing these features.

Based on the similarity score $CP(t, t')$, we can identify a similar task set of t, denoted as NT ($|NT| = L$). However, not all the features of similar tasks are useful to complement the requirements of t. The most relevant features need to be extracted. Attention is a mechanism for flexibly selecting the relevant part of context information, which can facilitate global learning (Vaswani et al., 2017; Zhang et al., 2022b). Thus, in this article, a multi-head self-attention mechanism (Vaswani et al., 2017) is adopted to capture the complex complementary relationships among these tasks with dynamic weights and select the most relevant features in similar tasks. Then the weight of nt_l ($nt_l \in NT$) is measured by the scaled dot-product attention under a specific attention head x as equation (3).

$$\alpha_{t,nt_l}^x = softmax\left(\frac{\left(\boldsymbol{v}^t \boldsymbol{W}_Q^x\right)\cdot\left(\boldsymbol{v}^{nt_l}\boldsymbol{W}_K^x\right)^T}{\sqrt{d_k}}\right) \tag{3}$$

where $\boldsymbol{W}_Q^x, \boldsymbol{W}_K^x \in \mathbb{R}^{d\times d_k}$. d is the dimension size of $\boldsymbol{v}^t$. $\sqrt{d_k}$ is the scale factor, which can avoid an overly large value of the inner product. $softmax(\cdot)$ is a normalisation function that adjusts the weights in the range [0, 1] as shown by equation (4).

$$g(z) = softmax(z) = \frac{1}{1+e^{-z}} \tag{4}$$

Each task vector $\boldsymbol{y}_t^x$ under the attention head x is obtained by summing all the weighted vectors by equation (5).

$$y_t^x = \alpha_{t,t}\left(\boldsymbol{v}^t \boldsymbol{W}_V^x\right) + \sum_{l=1}^{L}\alpha_{t,nt_l}\left(\boldsymbol{v}^{nt_l}\boldsymbol{W}_V^x\right) \tag{5}$$

where $\boldsymbol{W}_V^x \in \mathbb{R}^{d\times d_v}$. Different task representations are derived from each attention head x. Each head can capture diverse aspects in distinct subspaces by projecting original task vectors to many different subspaces. Task vectors from all heads are concatenated as below.

$$\boldsymbol{y}_t^{\sim} = Concat\left(\boldsymbol{y}_t^1, \ldots, \boldsymbol{y}_t^x, \ldots, \boldsymbol{y}_t^H\right) \tag{6}$$

where H denotes the total number of attention heads. We further feed the concatenated vector into a MLP to obtain the task fused representation $\boldsymbol{a}^t$ as follows:

$$\boldsymbol{a}^t = MLP\left(\boldsymbol{y}_t^{\sim}\right) \tag{7}$$

3.4 *Worker preferences estimation based on worker-task interactions*

Historical records reflect worker's preferences for different task characteristics. Also, workers are likely to work on similar tasks in the future. To capture a worker's task selection behaviour, a multi-head self-attention mechanism is adopted to measure the contribution level of different task features in estimating worker preferences. Since the *WorkerEst* module is symmetric with the *TaskMod* module, the derivation of the worker fused representation $\boldsymbol{a}^w$ follows the same procedures as the task fused representation $\boldsymbol{a}^t$, i.e., equations (3)–(7). To save space, we do not introduce too much.

3.5 *Model training*

Task-worker interactions can be regarded as one-class preference data (Lin et al., 2014) which cannot differentiate the task-worker matching degree. To handle the one-class problem, we regard the model training as a ranking problem (Rendle et al., 2009). Given t, we construct a contrastive worker pair to specify the matching order. A positive worker w^+ is the one who has participated in t, while a pseudo-negative worker w^- refers to the one who has not participated in t. Then we have the matching order $\langle t, w^+\rangle \succeq \langle t, w^-\rangle$. Accordingly, we have $r_{tw^+} \geq r_{tw^-}$ where r_{tw} denotes the matching degree between t and w by the inner product of $\boldsymbol{a}^t$ and $\boldsymbol{a}^w$:

$$r_{tw} = \left(\boldsymbol{a}^t\right)^T \boldsymbol{a}^w \tag{8}$$

Then the objective of model training is to maximise the loss of a mini-batch B as shown in equation (9).

$$Loss = \frac{1}{|B|}\sum_{(t,w^+,w^-)\in B} -\ln\sigma\left(r_{tw^+} - r_{tw^-}\right) + \lambda\|\Theta\|^2 \tag{9}$$

where $\sigma(\cdot)$ is a sigmoid function, $\lambda\|\Theta\|^2$ is a coefficient of regulariser to controlling overfitting, and B is the collection of training instances.

4 Experiments and evaluation

4.1 *Preparations*

4.1.1 *Dataset*

As there is no public dataset available for KIC recommendation, we cooperated with ZBJ.COM, a popular KIC platform founded in 2006 in China. We collected one year's transactions from November 2019 to November 2020 and remove tasks that have incomplete information, are not completed or have less than five workers involved. Table 3 presents the statistics of the dataset. We select 20% of tasks and all their interactions for prediction performance testing in comparison experiments, and set the percentage of tasks in the training set as 20%, 40%, 60%, and 80%. It is noted that all tasks in the testing set are cold-start.

Table 3 The statistics of the dataset

Number of tasks	6,453
Number of workers	3,990
Number of seekers	5,665
Total interactions between tasks and workers	49,356
Min number of participating workers in each KIC task	5
Max number of participating workers in each KIC task	26
Average number of participating workers in each KIC task	7.65
Min number of participated tasks of each worker	1
Max number of participated tasks of each worker	678
Average number of participated tasks of each worker	12.37

4.1.2 Baselines

To show the effectiveness of HFMIRec, we compare our model with the following methods:

- *BOW* (Ambati et al., 2011): it is a bag-of-words approach that computes workers' preferences toward a new task by averaging the number of overlapped word terms in task descriptions. In this paper, we extract the nouns, verbs and adjectives in task descriptions as word terms.
- *TaskRec* (Yuen et al., 2015): it is a unified PMF model that infers worker ratings from their interacting behaviours to capture worker-task preferences, work-category preferences and task-category preferences.
- *BPR* (Rendle et al., 2009): Bayesian personalised ranking is a popular factorisation-based method, which uses a pairwise approach to build the training data and then sorts task-worker pairs directly, rather than scoring individual workers. Only task IDs and worker IDs are used to learn both task latent vectors and worker latent vectors.
- *DREX* (Wu et al., 2011): DREX aims to recommend workers who have participated in the most similar tasks to the new task. Cosine similarity based on the TF-IDF of word terms in task descriptions is utilised to measure the similarity between tasks, and workers are then ranked and selected according to their popularity measured by participation frequency.
- *BTR* (Dai et al., 2020): Bayesian task recommendation is a hybrid recommendation method that combines task features into BPR to recommend suitable workers for new tasks. To tackle the task cold-start problem, a neural network is adopted to transform the high-dimensional task features into a low-dimensional semantic vector. Only worker IDs are used in the model to learn the latent vectors of workers.
- *DHRec* (Yu et al., 2022): it combines multiple commonly used recommendation methods, namely FM, MF and MLP to comprehensively leverage the explicit and implicit features of both tasks and workers, thus solving the cold-start problem.

4.1.3 Parameter setting

For TaskRec, BPR, BTR and DHRec, the dimensions of the latent representations of both tasks and workers are set as 5. For BTR, the number of task categories is set as 100. For DREX, the number of neighbour tasks to the new task is set as 20 for the best performance. The main parameters and the default values used in HFMIRec are listed in Table 4. For fairness, we use gird search and fine-tune all the parameters by adopting adaptive moment estimation (Adam) (Kingma and Ba, 2014). Specifically, we initialise model parameters with Xavier (Glorot and Bengio, 2010). Each experiment is repeated three times, and we report the average values.

Table 4 Main parameters

Parameter	*Value*
Learning rate λ	0.001
Batch size	1,024
BERT embedding dimension	1,024
Worker and task embedding dimensions d	64
Number of worker's historical records M	10
Number of similar tasks L	10
Epochs	1,000
Number of attention heads H	8
Dropout rate	0.2
Number of negative samples	10

4.2 Evaluation metrics

To compare the prediction quality of HFMIRec, we utilise the following metrics, i.e., precision, recall, normalised discounted cumulative gain (NDCG) and hit ratio (HR) for performance evaluations. The larger values of these metrics indicate better performance of worker recommendation. In this paper, we assume to recommend top-K workers for each new task. Using $R(t)$ and $Tr(t)$ denotes the recommended worker set and true participating worker set, respectively, the four metrics can be defined as equations (10)–(13).

$$Precision@K = \frac{|R(t) \cap Tr(t)|}{|R(t)|} \tag{10}$$

$$Recall@K = \frac{|R(t) \cap Tr(t)|}{|Tr(t)|} \tag{11}$$

$$NDCG@K = \frac{1}{IDCG} \times \sum_{k=1}^{K} \frac{2^{r_{t,k}} - 1}{\log(k+1)} \tag{12}$$

where NDCG is the maximum possible DCG for a given set of recommendations and r_{tk} is 1 if the recommended worker at position k have participated in the task t and 0 otherwise.

$$HR@K = \frac{\sum_{t \in T_N} I(Tr(t) \cap R(t) \neq \Phi)}{|T_N|} \tag{13}$$

where $I(\cdot)$ is an indicative function.

4.3 Results and analysis

4.3.1 Performance evaluation

Table 5 reports the performance of HFMIRec and baselines on four evaluation matrices with K equal to 3 and 5 on ZBJ.COM datasets with different sparsity of training data. The last row of each child table indicates the percentage of improvement gained by HFMIRec compared to the best baseline under the corresponding experimental settings.

As shown in Table 5, HFMIRec achieves the best performance on all datasets with different training data sparsity and significantly outperforms the best baseline. Specifically, on the metrics of Precision@3 and NDCG@3, HFMIRec outperforms BTR by 31.3% and 33.5% on the dataset with 80% training data sparsity, 33.8% and 30.4% on the dataset with 60% training data sparsity, 53.1% and 57.3% on the dataset with 40% training data sparsity, 49.8% and 48.6% on the dataset with 20% training data sparsity, respectively. HFMIRec shows good prediction performance on the dataset with high data sparsity, indicating that it not only can handle the task cold-start problem but also can alleviate the high data sparsity in the dataset.

BOW performs the worse because it just uses the task description and the representations of tasks and workers do not appear to capture the latent semantic meaning of the text by employing the bag-of-words. TaskRec performs poorly as well as it utilises only workers' historical behaviours on tasks based on the worker-task interaction matrix, thus it cannot tackle the task cold-start problem studied in this paper. Instead of transforming workers' implicit interaction behaviours into ratings for participation prediction, BPR optimises the latent representations of both tasks and workers with the pairwise ranking loss on workers' implicit feedback. Consequently, it can rank more suitable workers higher than less suitable ones.

Compared with both content-based and collaborative filtering models, the hybrid models, namely DREX, DHRec and BTR achieve relatively higher performance. DREX obtains general performance as it only considers workers who participated in neighbour tasks, which may exclude the best suitable workers. Although DHRec leverages both explicit and implicit features of tasks and workers, it learns the implicit representation of cold-start tasks separately. Therefore, its performance is not good as expected. BTR achieves the best performance among the six baselines and largely outperforms BPR. This suggests the effectiveness of incorporating heterogeneous task features and learning the semantic-aware task representations in KIC for cold-start task recommendation.

Table 5 Performance comparison on ZBJ.COM dataset

80% training data	*Precision@3*	*Recall@3*	*NDCG@3*	*Hit@3*	*Precision@5*	*Recall@5*	*NDCG@5*	*Hit@5*
BOW	0.0023	0.0009	0.0008	0.0062	0.0020	0.0014	0.0005	0.0085
DREX	0.1575	0.0622	0.0462	0.3989	0.1475	0.0955	0.0307	0.5290
BPR	0.1574	0.0671	0.1559	0.4014	0.1467	0.1044	0.1502	0.5291
BTR	<u>0.1929</u>	<u>0.0841</u>	<u>0.1981</u>	<u>0.4582</u>	<u>0.1784</u>	<u>0.1276</u>	<u>0.1882</u>	<u>0.6043</u>
DHRec	0.1778	0.0771	0.1739	0.4355	0.1521	0.1078	0.1587	0.5461
TaskRec	0.0355	0.0146	0.0511	0.0950	0.0315	0.0218	0.0493	0.1418
HFMIRec	**0.2533**	**0.1009**	**0.2644**	**0.5500**	**0.2277**	**0.1501**	**0.2435**	**0.6754**
Improvement	31.3%	19.9%	33.5%	20.0%	27.6%	17.6%	29.4%	11.8%
60% training data	*Precision@3*	*Recall@3*	*NDCG@3*	*Hit@3*	*Precision@5*	*Recall@5*	*NDCG@5*	*Hit@5*
BOW	0.0021	0.0009	0.0007	0.0062	0.0026	0.0020	0.0006	0.0124
DREX	0.1614	0.0637	0.0476	0.4059	0.1527	0.0989	0.0318	0.5391
BPR	0.1508	0.0648	0.1527	0.3816	0.1274	0.0912	0.1379	0.4922
BTR	<u>0.1868</u>	<u>0.0792</u>	<u>0.1985</u>	<u>0.4482</u>	<u>0.1586</u>	<u>0.1110</u>	<u>0.1771</u>	<u>0.5404</u>
DHRec	0.1612	0.0670	0.1664	0.3986	0.1430	0.1011	0.1539	0.5106
TaskRec	0.0307	0.0138	0.0409	0.0851	0.0275	0.0206	0.0405	0.1277
HFMIRec	**0.2499**	**0.1002**	**0.2589**	**0.5709**	**0.2166**	**0.1437**	**0.2336**	**0.6878**
Improvement	33.8%	26.5%	30.4%	27.4%	36.6%	29.4%	31.9%	27.3%

Note: The bold and underlined values indicate the best and second performance, respectively.

Table 5 Performance comparison on ZBJ.COM dataset (continued)

40% training data	*Precision@3*	*Recall@3*	*NDCG@3*	*Hit@3*	*Precision@5*	*Recall@5*	*NDCG@5*	*Hit@5*
BOW	0.0059	0.0024	0.0020	0.0170	0.0046	0.0032	0.0011	0.0209
DREX	0.1557	0.0612	0.0462	0.3881	0.1478	0.0955	0.0309	0.5259
BPR	0.1404	0.0620	0.1444	0.3589	0.1209	0.0875	0.1312	0.4610
BTR	0.1589	0.0676	0.1586	0.3858	0.1609	0.1119	0.1619	0.5319
DHRec	0.1527	0.0665	0.1541	0.3787	0.1421	0.1005	0.1479	0.5092
TaskRec	0.0265	0.0116	0.0359	0.0766	0.0250	0.0181	0.0361	0.1163
HFMIRec	**0.2432**	**0.0966**	**0.2495**	**0.5407**	**0.2156**	**0.1417**	**0.2288**	**0.6685**
Improvement	53.1%	42.8%	57.3%	39.3%	34.0%	26.6%	41.4%	25.7%
20% training data	*Precision@3*	*Recall@3*	*NDCG@3*	*Hit@3*	*Precision@5*	*Recall@5*	*NDCG@5*	*Hit@5*
BOW	0.0067	0.0027	0.0021	0.0201	0.0064	0.0044	0.0014	0.0302
DREX	0.1376	0.0545	0.0416	0.3610	0.1397	0.0905	0.0290	0.5143
BPR	0.1267	0.0566	0.1330	0.3390	0.1004	0.0735	0.1145	0.4227
BTR	0.1532	0.0654	0.1597	0.3887	0.1600	0.1116	0.1641	0.5461
DHRec	0.1485	0.0637	0.1499	0.3844	0.1316	0.0925	0.1393	0.4879
TaskRec	0.0241	0.0101	0.0335	0.0667	0.0238	0.0162	0.0340	0.1064
HFMIRec	**0.2295**	**0.0914**	**0.2374**	**0.5275**	**0.2014**	**0.1323**	**0.2159**	**0.6561**
Improvement	49.8%	39.9%	48.6%	35.7%	25.9%	18.6%	31.6%	20.1%

Note: The bold and underlined values indicate the best and second performance, respectively.

Table 6 Performance of HFMIRec with different components

Model variants	*Precision@3*	*Recall@3*	*NDCG@3*	*Hit@3*	*Precision@5*	*Recall@5*	*NDCG@5*	*Hit@5*
TID+WID	0.1574	0.0671	0.1559	0.4014	0.1467	0.1044	0.1502	0.5291
T+WID	0.1929	0.0841	0.1981	0.4582	0.1784	0.1276	0.1882	0.6043
T+W	0.2476	0.0982	0.2559	0.5593	0.2170	0.1428	0.2324	0.6778
T+NT+W	0.1748	0.0688	0.1788	0.4299	0.1541	0.1010	0.1635	0.5507
T+W+HR	0.2213	0.0884	0.2261	0.5221	0.2077	0.1372	0.2154	0.6731
T+NT+W+HR	0.1702	0.0673	0.1746	0.4012	0.1554	0.1018	0.1632	0.5329
T+NT+AT+W+HR	0.2373	0.0955	0.2450	0.5407	0.2234	0.1487	0.2336	**0.6801**
T+NT+W+HR+AW	0.1859	0.0746	0.1893	0.4415	0.1642	0.1091	0.1735	0.5507
T+NT+AT+W+HR+AW	**0.2533**	**0.1009**	**0.2644**	**0.5500**	**0.2277**	**0.1501**	**0.2435**	0.6754

4.3.2 Model analysis

In HFMIRec, several components, namely the heterogeneous features of t and w, the task-task interactions and worker-task interactions are all incorporated. Specifically, two self-attention mechanisms are adopted to capture the task-task interactions and worker-task interactions, thus learning the fused representation of t and w. For performance comparison, we combine different components in HFMIRec as below and their performances are summarised in Table 6.

- *TID+WID:* the IDs instead of heterogeneous features of both t and w are used to learn the latent vector of t and w. This model is equivalent to BPR mentioned in Section 4.1.2.
- *T+WID:* the heterogeneous features of t are employed to learn the latent vector of t, while only the ID of w is used to learn the latent vector of w. This is BTR in Section 4.1.2.
- *T+W:* it only uses the feature representation $\boldsymbol{v}^t$ (T) and $\boldsymbol{v}^w$ (W) learned in the *FeaPre* module from the heterogeneous features of t and w to obtain the task-worker matching degree.
- *T+NT+W:* for task requirements modelling, it uses a MLP instead of a self-attention mechanism to capture the task-task interactions and learn $\boldsymbol{a}^t$ in the *TaskMod* module. In specific, the feature representation of t (T) and its similar tasks (NT) are directly concatenated to feed into the MLP. For worker preferences estimation, the feature representation $\boldsymbol{v}^w$ (W*)* is directly used.
- *T+W+HR:* similar to *T+NT+W*, it directly uses the feature representation $\boldsymbol{v}^t$ (T) as the task requirements modelling. For worker preferences estimation, it uses a MLP instead of a self-attention mechanism to capture

the worker-task interactions and learn $\boldsymbol{a}^w$ in the *WorkerEst* module. In specific, the feature representation of *w* (*W*) and his/her historical records (*HR*) are directly concatenated to feed into the MLP.

- *T+NT+W+HR:* it uses a MLP instead of a self-attention mechanism learn $\boldsymbol{a}^t$ and $\boldsymbol{a}^w$ in the *TaskMod* and *WorkerEst* module, respectively.
- *T+NT+AT+W+HR:* it uses a MLP instead of a self-attention mechanism in the *WorkerEst* module to learn $\boldsymbol{a}^w$. AT denotes the self-attention mechanism in the *TaskMod* module.
- *T+NT+W+HR+AW:* it uses a MLP instead of a self-attention mechanism in the *TaskMod* module to learn $\boldsymbol{a}^t$. AW denotes the self-attention mechanism in the *WorkerEst* module.
- *T+NT+AT+W+HR+AW:* it is the proposed model HFMIRec in this paper.

Effect of heterogeneous features: as shown in Table 6, *T+W* outperforms *TID+WID* and *T+WID*, indicating that incorporating heterogeneous features of tasks and workers can largely improve the recommendation performance in KIC. Meanwhile, the employment of heterogeneous features of tasks can alleviate the task cold-start problem as T+WID behaves better than *TID+WID* on all the evaluation metrics.

Effect of multiple interactions: Table 6 shows that HFMIRec that with a self-attention mechanism in both the *TaskMod* module and the *WorkerEst* module, achieves the best performance among all the variants that use MLP instead. *T+NT+AT+W+HR* and *T+NT+AT+W+HR* all outperform *T+NT+W+HR*. These observations imply that the self-attention mechanism is effective in adaptively extracting the relevant and important information of similar tasks or historical records, thus capturing the multifarious task-task interactions and worker-task interactions and enhancing the expressiveness and accuracy of the fused representation of *t* and *w*. Further, it is also observed that *T+W* achieves better performance than *T+NT+W*, *T+W+HR* and *T+NT+W+HR*. A reasonable explanation is that directly combining the information of similar tasks or historical records will introduce noise when modelling task requirements and estimating worker preferences.

5 Conclusions

Accurately understanding and characterising task requirements and worker preferences are critical for automatic worker recommendation for KIC tasks. Meanwhile, KIC recommendation encounters the task cold-start problem in nature as only newly posted tasks need to be matched with workers. This paper proposed a novel deep neural framework based on the heterogeneous features of both tasks and workers and multiple interactions between tasks or between tasks and workers to recommend qualified workers to new KIC tasks, thus handling the task cold-start problem and enhancing the accuracy and expressiveness of the vector representation of both tasks and workers. Comprehensive experiments illustrate the effectiveness of the proposed model.

This paper has some limitations. We assume that worker preferences are static. While in reality, worker preferences and task participation patterns are dynamic. Thus designing a new KIC recommendation system that models worker's dynamic preferences is a research direction in the future. Further, service seekers also have preferences in selecting workers. In this paper, we did not take service seekers' hiring preferences and behaviours into consideration, which will be another research direction of our future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Project No. 52105507 and 52275476), and the Humanities and Social Sciences Research Planning Project of Chongqing Education Commission in 2022 (No. 22SKGH433). The authors would like to thank ZBJ.COM for providing the data. The authors also thank the members of the team of R&D of Technology Consulting Service Platform for their assistance and guidance.

References

Aldhahri, E., Shandilya, V. and Shiva, S. (2015) 'Towards an effective crowdsourcing recommendation system: a survey of the state-of-the-art', *2015 IEEE Symposium on Service-Oriented System Engineering*, pp.372–377.

Ambati, V., Vogel, S. and Carbonell, J. (2011) 'Towards task recommendation in micro-task markets', *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence. Publishing*, pp.80–83.

Dai, W., Wang, Y., Ma, J. and Jin, Q. (2020) 'BTR: a feature-based Bayesian task recommendation scheme for crowdsourcing system', *IEEE Transactions on Computational Social Systems*, Vol. 7, No. 3, pp.780–789.

Devlin, J., Chang, M-W., Lee, K. and Toutanova, K. (2019) 'BERT: pre-training of deep bidirectional transformers for language understanding', *Human Language Technologies*, pp.4171–4186, North American Chapter of the Association for Computational Linguistics, Minneapolis, Minnesota, USA.

Erat, S. and Krishnan, V. (2012) 'Managing delegated search over design spaces', *Management Science*, Vol. 58, No. 3, pp.606–623.

Fu, Y., Shen, B., Chen, Y. and Huang, L. (2021) 'TDMatcher: a topic-based approach to task-developer matching with predictive intelligence for recommendation', *Applied Soft Computing*, Vol. 110, No. 2021, pp.1–13.

Geiger, D. and Schader, M. (2014) 'Personalized task recommendation in crowdsourcing information systems – current state of the art', *Decision Support Systems*, Vol. 65, pp.3–16.

Ghezzi, A., Gabelloni, D., Martini, A. and Natalicchio, A. (2018) 'Crowdsourcing: a review and suggestions for future research', *International Journal of Management Reviews*, Vol. 20, No. 2, pp.343–363.

Glorot, X. and Bengio, Y. (2010) 'Understanding the difficulty of training deep feedforward neural networks', *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp.249–256.

Gong, Y. (2017) 'Estimating participants for knowledge-intensive tasks in a network of crowdsourcing marketplaces', *Information Systems Frontiers*, Vol. 19, No. 2, pp.301–319.

Hao, B., Zhang, J., Yin, H., Li, C. and Chen, H., (2021) 'Pre-training graph neural networks for cold-start users and items representation', *WSDM '21: Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp.265–273.

Howe, J. (2006) 'The rise of crowdsourcing', *Wired Magazine*, Vol. 14, No. 6, pp.1–4.

Hu, L., Jian, S., Cao, L., Gu, Z., Chen, Q. and Amirbekyan, A. (2019) 'HERS: modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation', *Proceedings of the AAAI Conference on Artificial Intelligence*, pp.3830–3837.

Jiang, Z., Huang, Y., Beil, D.R.J.M. and Management, S.O. (2021) 'The role of problem specification in crowdsourcing contests for design problems: a theoretical and empirical analysis', *Manufacturing & Service Operations Management*, Vol. 23, No. 3, pp.637–656.

Kingma, D. and Ba, J. (2014) 'Adam: a method for stochastic optimization', *3rd International Conference for Learning Representations*.

Lakshmanan, B. and Anand, S. (2022) 'Deep learning-based mitosis detection using genetic optimal feature set selection', *International Journal of Bio-Inspired Computation*, Vol. 19, No. 3, pp.189–198.

Lin, C.H., Kamar, E. and Horvitz, E. (2014) 'Signals in the silence: models of implicit feedback in a recommendation system for crowdsourcing', *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec City, Québec, Canada, pp.908–914.

Liu, W.B., Wang, Z.D., Liu, X.H., Zengb, N.Y., Liu, Y.R. and Alsaadi, F.E. (2017) 'A survey of deep neural network architectures and their applications', *Neurocomputing*, Vol. 234, No. 2017, pp.11–26.

Mo, J., Sarkar, S. and Menon, S. (2018) 'Know when to run: recommendations in crowdsourcing contests', *MIS Quarterly: Management Information Systems*, Vol. 42, No. 3, pp.919–943.

Pereira, A.S., Mazza, L.O., Pinto, P.C.C., Gomes, J.G.R.C., Nedjah, N., Vanzan, D.F., Pyrrho, A.S. and Soares, J.G.M. (2022) 'Deep convolutional neural network applied to Trypanosoma cruzi detection in blood samples', *International Journal of Bio-Inspired Computation*, Vol. 19, No. 1, pp.1–17.

Raza, S. and Ding, C. (2022) 'News recommender system: a review of recent progress, challenges, and opportunities', *Artificial Intelligence Review*, Vol. 55, No. 1, pp.749–800.

Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L., (2009) 'BPR: Bayesian personalized ranking from implicit feedback', *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Quebec, Montreal, Canada, pp.452–461.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017) 'Attention is all you need', *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, USA, pp.1–11.

Wang, J., Yang, Y., Wang, S., Chen, C., Wang, D. and Wang, Q. (2021) 'Context-aware personalized crowdtesting task recommendation', *IEEE Transactions on Software Engineering*, pp.1–13.

Wang, L., Meng, Z. and Yang, L. (2022) 'A multi-layer two-dimensional convolutional neural network for sentiment analysis', *International Journal of Bio-Inspired Computation*, Vol. 19, No. 2, pp.97–107.

Wei, J., He, J., Chen, K., Zhou, Y. and Tang, Z. (2017) 'Collaborative filtering and deep learning based recommendation system for cold start items', *Expert Systems with Applications*, Vol. 69, No. 1, pp.29–39.

Wu, W., Zhang, W., Yang, Y. and Wang, Q. (2011) 'DREX: Developer recommendation with k-nearest-neighbor search and expertise ranking', *18th Asia-Pacific Software Engineering Conference*, Ho Chi Minh City, Vietnam, pp.389–396.

Yang, B., Wang, X. and Ding, Z. (2021) 'Understanding service providers' competency in knowledgeintensive crowdsourcing platforms: an LDA approach', *Complexity*, Vol. 2021, pp.1–19.

Yang, K. (2019) 'Research on factors affecting solvers' participation time in online crowdsourcing contests', *Future Internet*, Vol. 11, No. 8, p.176.

Yang, K. and Dutta, S., (2021) 'Secure and efficient task matching with multi-keyword in multi-requester and multi-worker crowdsourcing', *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, Tokyo, Japan, pp.1–6.

Yang, Y., Mo, W., Shen, B. and Chen, Y. (2017) 'Cold-start developer recommendation in software crowdsourcing: a topic sampling approach', *SEKE*, pp.376–381.

Yin, X., Wang, H., Wang, W. and Zhu, K. (2020) 'Task recommendation in crowdsourcing systems: a bibliometric analysis', *Technology in Society*, Vol. 63, pp.1–14.

Yu, X.H.Y., Du, J., Wang, Z., Jiang, F. and Gong, D. (2022) 'Developer hybrid recommendation algorithm based on combination of explicit features and implicit features', *Journal of Software*, Vol. 33, No. 5, pp.1635–1651.

Yuen, M-C., King, I. and Leung, K-S. (2012) 'Task recommendation in crowdsourcing systems', *Proceedings of the First International Workshop on Crowdsourcing and Data Mining*, Beijing, China, pp.22–26.

Yuen, M-C., King, I. and Leung, K-S. (2015) 'TaskRec: a task recommendation framework in crowdsourcing systems', *Neural Processing Letters*, Vol. 41, No. 2, pp.223–238.

Yuen, M-C., King, I. and Leung, K-S. (2016) 'An online-updating algorithm on probabilistic matrix factorization with active learning for task recommendation in crowdsourcing systems', *Big Data Analytics*, Vol. 1, No. 1, pp.1–29.

Yuen, M-C., King, I. and Leung, K-S. (2021) 'Temporal context-aware task recommendation in crowdsourcing systems', *Knowledge-Based Systems*, Vol. 219, No. 2021, pp.1–13.

Zhang, J., Ma, C., Zhong, C., Zhao, P. and Mu, X. (2022a) 'Combining feature importance and neighbor node interactions for cold start recommendation', *Engineering Applications of Artificial Intelligence*, Vol. 112, No. 2022, pp.1–11.

Zhang, N., Wang, F., Chen, X., Zhao, T. and Kang, Q. (2022b) 'Spatial-temporal attention-based seq2seq framework for short-term travel time prediction', *International Journal of Bio-Inspired Computation*, Vol. 20, No. 1, pp.23–37.

Zhang, S., Yao, L., Sun, A. and Tay, Y. (2019) 'Deep learning based recommender system: A survey and new perspectives', *ACM Computing Surveys*, Vol. 52, No. 1, pp.1–38.

Zhang, X. and Su, J. (2019) 'A combined fuzzy DEMATEL and TOPSIS approach for estimating participants in knowledge-intensive crowdsourcing', *Computers & Industrial Engineering*, Vol. 137, No. 2019, pp.1–15.

Zhang, X., Gong, B., Cao, Y., Ding, Y. and Su, J. (2020a) 'Investigating participants' attributes for participant estimation in knowledge-intensive crowdsourcing: a fuzzy DEMATEL based approach', *Electronic Commerce Research*, Vol. 22, No. 2, pp.811–842.

Zhang, X., Xia, E., Shen, C. and Su, J. (2022c) 'Factors influencing solvers' behaviors in knowledge-intensive crowdsourcing: a systematic literature review', *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 17, No. 4, pp.1297–1319.

Zhang, Z., Sun, H. and Zhang, H. (2020b) 'Developer recommendation for topcoder through a meta-learning based policy model', *Empirical Software Engineering*, Vol. 25, No. 1, pp.859–889.

Zheng, L., Noroozi, V. and Yu, P.S. (2017) 'Joint deep modeling of users and items using reviews for recommendation', *WSDM '17: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp.425–434.

Zhu, J., Shen, B. and Hu, F. (2015) 'A learning to rank framework for developer recommendation in software crowdsourcing', *2015 Asia-Pacific Software Engineering Conference (APSEC)*, New Delhi, India, pp.285–292.