

Research on Task Allocation of Crowdsensing User Activity Based on the ResNet Neural Network

Li Zhang^{1,3,a*}, Shukui Zhang^{2,3,b}, Ye Tao^{2,c}, Pingquan Wang^{1,d}, Hao Long^{4,e}

¹Yangzhou Polytechnic Institute, Yangzhou, Jiangsu 215006, China

²School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China

³Jiangsu Provincial Key Laboratory for High tech Research of Wireless Sensor Networks, Suzhou, Jiangsu 215006, China

⁴Xuzhou College of Industrial Technology, Xuzhou, Jiangsu 221140, China

Currently, there is a lack of multi-dimensional research on sensing task allocation in mobile crowdsensing networks. This paper proposes a task allocation algorithm based on the positive association between sensing entities. First, sensing activity is measured by the number of successful completions of sensing tasks by participating users, divided into direct and indirect activities. Meanwhile, a correlation model is constructed by combining direct and indirect activities. Furthermore, to improve the success rate of sensing task allocation, the ResNet neural network is trained to eliminate the gradient vanishing problem. Moreover, the AM Softmax cross-entropy loss function is adopted to narrow the gap between peers and expand the gap between different classes, thereby improving the accuracy of sensing task allocation. Finally, the constructed association model and the classification neural network are integrated for sensing task allocation. Experimental verification indicates that the proposed algorithm has significantly enhanced the accuracy of task allocation and can better provide accurate task allocation services.

Keywords: mobile crowdsensing; mobile users; activate; task allocation

1. INTRODUCTION

In recent years, mobile crowdsourcing (MCS) [1] has become a popular sensing mode for exploiting the collective sensing capability of numerous mobile users. Unlike traditional sensor networks that rely on specialized sensor deployment, any user-aware device carrying mobile devices equipped with multi-functional sensors, such as smartphones and tablets, can become a data source in MCS. For instance, for traffic monitoring, users can report road conditions on their way home [2], [3], or collect noise information for noise pollution monitoring during leisure walks [4]. Moreover, users can play the role of information requesters by actively requesting some mobile users to collect sensing data to meet their needs.

Owing to its efficiency and scalability in collecting various types of sensor data, MCS has a wide range of applications, such as public safety [5], [6], environmental monitoring [6], [7], healthcare [8], signal map construction [9], and indoor positioning [10].

At present, there is a certain understanding of MCS task allocation. Usually, there are three roles in MCS: users participating in sense activities, requesting users sending sensing task requests, and MCS platforms. The main task of the MCS sensing platform is to distribute sensing tasks appropriately to non-sensing users and provide feedback on the required sensing data to the corresponding task requesters [11]. In practice, it is important to consider the appropriate allocation of sensing users and tasks and to match users with tasks. Only when the matching between users and sensing tasks is high, can the MCS platform improve their matching.

Therefore, the main contributions of this article are as follows:

*Corresponding author. ^aE-mail: greenwuhu@126.com, ^bZhangsk@suda.edu.cn, ^c20165227044@stu.suda.edu.cn, ^dpingquanw@163.com, ^elonghhao@163.com

- (1) New ideas are proposed to address the issue of user recruitment for group intelligence sensing. Unlike the existing approach of combining the selection of sub-regions with the selection of users, this study makes decisions on the selection and recruitment of users for tasks based on user activity attributes, and a set of users suitable for completing multiple tasks are recruited.
- (2) A user recruitment strategy is proposed based on residual network crowdsensing. Firstly, the user attributes of the problem are modeled to obtain the sensing user activity in the model. To provide users with a high matching degree of sensing tasks, the residual neural network algorithm and the additive margin Softmax loss (AM Softmax) cross-entropy loss function are combined for training with sensing users' sensing activities. To find strongly-matched sensing tasks for users and reduce the discrepancies between different users' sensing activities, the AM Softmax function is used. By recruiting users for sensing tasks through the close connection between users, the accuracy of user recruitment is further improved.
- (3) The proposed algorithm was validated using data from 45 sensing devices over about 9 months at the Massachusetts Institute of Technology, and the proposed user active sensing task recruitment strategy based on the ResNet neural network was superior to other benchmark strategies.

The remainder of this article is organized as follows. Firstly, previous related research is discussed in Section 2. Then, in Section 3, the issue of user recruitment is proposed, and the relevant concepts of user recruitment are defined. Section 4 provides a detailed explanation of the model for user recruitment and the process based on the deep residual network. The accuracy of user recruitment is evaluated in Section 5, and Section 6 concludes this article.

2. REVIEWS OF RELEVANT DOMESTIC AND INTERNATIONAL STUDIES

To find the optimal task participants for the sensing of time constraints in tasks, researchers have investigated this problem in different ways. Wang et al. [12] proposed the recruitment of users for time-sensitive group intelligence sensing tasks under variable user conditions. Due to the time sensitivity and heterogeneity of the sensing task, the time spent by participants depends on not only the distance traveled, but also the time users require to complete all tasks. Guo et al. [13] studied multi-tasking problems in both intentional and unintentional situations, using time-sensitive and delay-tolerant strategies. For a large number of sensing tasks and thousands of users, in order to obtain a satisfactory user set, heuristic ideas were applied to genetic algorithms, and the superiority of the model was experimentally verified. To maximize the income of new employees, Wu et al. [14] estimated the time spent by new employees completing tasks; the researchers used a constraint optimization strategy with

unknown parameters, and finally, the optimization problem was solved through genetic algorithms to achieve a new strategy for generating employee tasks. Xiao et al. [15] adopted an approximate greedy algorithm to complete the multi-user sensing task to minimize sensing time based on the deadline and sensing activity time of the sensing task.

Other scholars have conducted studies covering as many sense areas as possible or setting a greater number of tasks to be completed. Chen et al. [16] achieved vehicle trajectory coverage and minimized the number of assigned vehicle task users through crowd tracking, where vehicles were located through the photographer's filming environment (including location and direction) and road network. Tao et al. [17] investigated privacy protection for online task allocation with the goal of minimizing the total distance, and they designed a privacy mechanism based on a hierarchically well-separated tree to avoid ineffective task allocation due to the sensing of task location information. Jiang et al. [18] addressed the high cost and instability of transmission in the task allocation process of crowdsensing. The participant selection algorithm with the lowest cost and the participant selection algorithm with the least number of users and the lowest user incentive cost were proposed, and the overall performance was verified by experiments. Wang et al. [19] improved mobile crowdsourced user recruitment based on prediction. To achieve the probability allocation of interest points, a semi-Markov model was utilized to minimize data costs as a predictive mobile-aware user recruitment strategy. Wang et al. [20] proposed a two-stage resource-balancing task allocation scheme with the optimization goal of minimizing task response time and balancing resource constraints. A resource-balanced task allocation method based on resource occupancy was proposed for edge servers, which selects a task set for edge servers and then implements a participant resource-balanced task allocation strategy based on improved ant colony optimization algorithms. Wang et al. [21] constructed a hybrid user model, proposed a list-level ranking learning strategy, and developed a collaborative ranking method to address the issues of low data quality and high incentive costs caused by low user activity in group intelligence sensing. The users' historical behaviors were analyzed, inferior sensing users were filtered out, and a mixed user model was constructed using similarity. Meanwhile, probability matrix decomposition was used to predict the willingness of participating users. Based on this, the task allocation accuracy was improved, and sensing user movement distance was reduced. Jiang et al. [22] solved the participant selection problem under multi-task allocation conditions by establishing a participant service quality model based on the historical task completion status of participating users and utilizing their cumulative reputation and willingness to participate. The platform revenue and participant revenue were ensured as much as possible. Meanwhile, the service quality of sensing users in sensing activities was improved. Wang et al. [23] aimed to reduce user trajectories and task allocation by selecting users through state maps and dependency maps and using the best set of images collected by cameras to cover the tracking area, thereby achieving the best sensing user set and minimizing data costs.

The issue of sensing task allocation has been studied from different perspectives. For example, Estrada et al. [24] input sensing tasks through intelligent terminal devices, delayed or rejected sensing task requests with unpredictable user context, and proposed particle swarm optimization-based multi-objective task sense user recruitment under the premise of maintaining user reputation. Cheung et al. [25] investigated heterogeneous user-distributed task selection with different initial locations, destinations, costs, speeds, and reputations for sensing tasks. By combining sensing technology with crowdsourcing technology, time-sensitive and location-related information was effectively collected. Du et al. [26] utilized motion and acoustic sensors on mobile devices to characterize customer actions and interactions, thereby identifying potential buyers in physical store retailers and revealing how customers interact with each other during activities. Luo et al. [27] studied the assignment problem of the system for multi-agent mobile sensing tasks comprising multiple heterogeneous robots. The main focus is on how robots can handle complex real-world environments, and self-healing allocation strategies were proposed for sense region information maps and information maps. Yao et al. [28] obtained visual and meteorological datasets through crowdsourcing and spider robots, respectively, to address the challenging issue of capturing air pollution with μ of less than 2.5. The proposed scheme has obvious advantages in accuracy, recall, and other aspects for the fine particles. Wu et al. [29] developed a participant selection method based on user willingness in their research on mobile group intelligence sensing tasks to reflect user willingness. The user's willingness to perform sensing tasks was quantified by integrating information such as user characteristics, task attributes, and real-time environment. The effectiveness and rationality of the mechanism were verified for balancing user willingness and user participant selection.

As evident, sensing task allocation has received much attention, and there are various research strategies available. To improve the accuracy and recall of sensing tasks, this article studies the correlation between the activity subjects, users, and tasks in sensing activities. Then, a user-active sensing task allocation strategy is proposed based on the ResNet neural network, and its impact on sensing task allocation is analyzed.

3. SENSING MODEL

To further understand the impact of sensing task allocation, this article considers the users' participation in sensing activities and investigates the impact of user activities on sensing activities. User activity refers to the number of times a user completes sensing tasks during their participation in the activity and whether the user strives to complete assigned tasks. The continuous improvement in the performance of sensors and wearable and gyroscopic devices enables sensing activities to benefit from the smart devices (SD) carried by users. Participating users may form potential connections when perceiving tasks depending on whether they accept them or not. The acquisition of sensing tasks involves users

directly asking for tasks, which is known as Direct Initiative (DI), and another way is through other user introductions to accept tasks, which is known as Indirect Initiative (II).

(1) Direct Initiative

The direct request of users to accept tasks or the sensing of tasks assigned to them by the platform is called 'user-direct activity'. DI indicates the user's completion of the accepted tasks, so it provides a comprehensive evaluation of the positivity of one SD directly towards another.

In sensing activities, the purpose of the sensing network is the ultimate completion of tasks, and sensing is meaningful only when the tasks of the sensing platform are completed. Therefore, the sensing activity is measured according to the successful completion of the sensing task. Direct activity refers to the ratio of users' successful completion of sensing tasks in sensing activities to their participation in sensing tasks.

$$DI = \frac{\text{Successfully completed sensing task}}{\text{participated in sensing task}} \quad (1)$$

The initial value of user activity is defined based on the number of times intelligent devices participate in sensing tasks within the network:

$$INIT_DI_i = \frac{\min(DI_i, \alpha)}{\alpha} \quad (2)$$

where $\min(DI_i, \alpha)$ denotes taking the minimum value of DI_i and α ; DI_i denotes the number of sensing tasks completed by the i -th sensing device; α denotes the number of times all intelligent devices directly interact with DI_i to complete the sensing task in a time interval; $INIT_DI_i$ represents the value of i , and it falls within (0,1).

The direct activity of smart devices is not only related to the number of tasks completed between them. With time, the number of associations between intelligent agents will also change, thereby reflecting the timeliness of initiation over time. Therefore, the logistic function [30] shown in equation (3) is utilized to balance the weakening of positivity over time:

$$f(t_{i,j}) = \frac{1}{1 + e^{t_{i,j}}} \quad (3)$$

where $t_{i,j}$ represents SD_i for SD_j . $f(t_{i,j})$ represents the time difference between the interactions of j , and it is a monotonically increasing function with values between (0, 1). To more effectively depict the influencing factors of positivity over time on sensing tasks, the value of the logistic function varies over time, and an impact factor is added to characterize the weighted difference evaluation criteria, as shown in equation (4):

$$DI_{i,j} = \begin{cases} 1 & \left| f(t_{i,j}) - f(t_{i,j}) \right| \leq \beta \\ 0 & \left| f(t_{i,j}) - f(t_{i,j}) \right| > \beta \end{cases} \quad (4)$$

where, $DI_{i,j}$ represents SD_i for SD_j activity; β represents the evaluation factor, where X is the threshold. If the weighted time difference of the intelligent device falls within $(\beta - X, \beta + X)$, this indicates that the user perceives successful activity interactions and, successful activity interactions are accumulated. The calculation of direct activity is shown in (5):

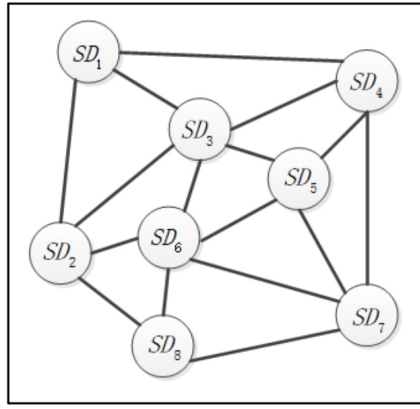


Figure 1 Sensing Task.

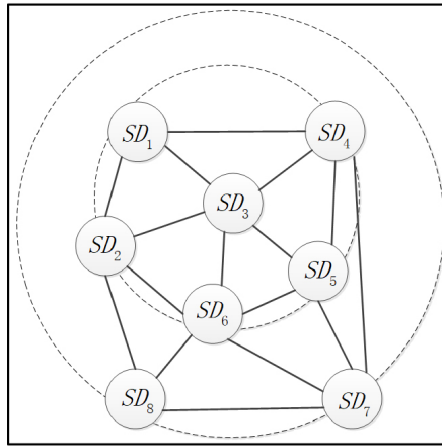


Figure 2 Sensing Task Correlation.

$$DI = INIT_DI_i + \frac{\text{total}_{\text{success}(i,j)}}{\text{total}_{\text{all}(i,j)}} \quad (5)$$

where, $\text{total}_{\text{success}(i,j)}$ represents the number of successful interactions between SD_i and SD_j , and $\text{total}_{\text{all}(i,j)}$ represents the interactions between SD_i and SD_j all times.

(2) Indirect Initiative

The indirect initiative is calculated based on the minimum world theory, where any two independent nodes can be connected by up to six edges. When searching for nodes that are not directly connected, the connections with the strongest data transmission activity between intelligent agents are found first.

In Figure 1, SD_i represents the i -th smart device, and any two smart devices have a link between them, indicating that they have successfully completed the sensing task. The more they complete sensing tasks with each other, the closer their activity is.

Figure 2 shows the correlation diagram between SDs for calculating the indirect positivity between i and other SDs at the current time point of a certain SD_i [31]. Taking SD_i as the starting point, it will have a direct correlation with SD_j with a arrange of j around i , as the first level; Compare with SD_j , SD_k is arranged in the order of SD_i , which is the second layer at the center of the circle. In this way, a concentric circle is formed with i as the center

for the convenience of calculation. This helps to obtain the shortest and most positive correlation. The association relationship in Figure 1 is simplified by removing excess edges between nodes, and only links between different layers are retained. In Figure 2, the SD_3 hierarchical relationship diagram between intelligent devices centered around 3 is used. The first layer involves ($SD_1, SD_2, SD_4, SD_5, SD_6$), which are all consistent with SD_3 and are directly connected; the second layer involves (SD_7, SD_8), which are directly associated with (SD_2, SD_4, SD_5, SD_6) [32].

Assuming that the current SD_i is on layer 2, the calculation of indirect positivity SD_i for the SD_j of layers 3 and the above is shown in formula (6):

$$II(i, j) = \frac{1}{2^{L_j-1}} \times \frac{1}{1 + e^{\frac{n}{2}}} \quad (6)$$

where, $II(i, j)$ represents SD_i and SD_j indirect correlation degree of j ; L_j represents SD_j the layer where j is located; n represents the layer from SD_i to SD_j . There are n paths in total. Considering practical problems and the small world principle, only the fifth layer was calculated in the experiment. When there are multiple paths, choosing a path from one layer to another depends on selecting the path that can complete multiple sensing tasks. One path represents completing a task, while multiple paths allow the sensing user

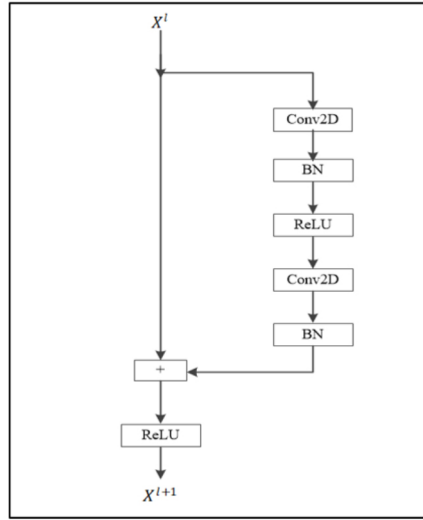


Figure 3 Residual Building Block of ResNet.

to complete multiple sensing tasks with the aim of enhancing their activities for the sensing task. For example, node SD_7 is on the second layer, so $L_7 = 2$; there are 2 paths from SD_i to SD_7 ($SD_i \rightarrow SD_5 \rightarrow SD_7$ and $SD_i \rightarrow SD_6 \rightarrow SD_7$), so $n=2$; Then, for SD_i and SD_7 , the indirect positivity of 7 is $(1/21)(1/(1 + e - 1)) \approx 0.37$.

The activities between intelligent devices can be either direct or indirect. SD_i and SD_j complete sensing tasks with each other, and SD_i and SD_j are calculated. The activity between j should first consider SD_i and SD_j . The direct activity for direct interaction between j should also be considered in the sensing network SD_j for SD_i the indirect positivity of i , as shown in equation (7).

$$SUM_{active} = \theta \times DI_{i,j} + (1 - \theta) \times II(i, j) \quad (7)$$

Equation (7) compute Smart Device SD_i and SD_j , Considering two special situations in reality. One is SD_i and SD_j when direct interaction is very frequent, and it mainly relies on direct interactions between intelligent devices; The second is SD_i when there is little interaction with intelligent devices directly connected to it, SD_i and SD_j . The positivity between j relies mainly on the indirect positivity of sensing networks and other intelligent devices. Therefore, in equation (7), to maintain relative balance, the dynamic equilibrium factor of θ and $1-\theta$ is used. The activity between computing intelligent devices is very important, and the calculation of d is shown in equation (8):

$$\theta = \min \left(1, \frac{|N_{i,j}|}{N_{min}} \right) \quad (8)$$

where, $|N_{i,j}|$ is the number of interactions for j ; N_{min} denotes the minimum number of interactions that can only be used to directly calculate the activity of the interaction party and the interaction N of different agents. The value of N_{min} varies, and it depends on the agent's direct and indirect activity.

4. RESIDUAL NETWORKS AND MODEL ALGORITHMS

4.1 Residual network and cross entropy function

By processing existing trajectory data and continuously adjusting the parameters for calculating the activity of smart devices, the sensing task activity of each smart device is calculated. Meanwhile, the cost-sensitive MetaCost algorithm [33] is utilized to ensure a high recall rate. Then, the residual network algorithm is adopted, and the AM Softmax cross-entropy loss function [34] is used to ensure high accuracy.

ResNet and its variants are widely used neural networks that can effectively handle the vanishing gradient problem through skip connections. The structure of the original basic ResNet is shown in Figure 3.

To achieve accurate matching of the sensing task, the calculated activity values of each intelligent user may vary greatly. For a more accurate classification of the user's activity value, the normalization operation is carried out first to narrow the activity gap between different users; meanwhile, the RELU activation function is utilized to address the issues of gradient explosion and disappearance. Further, batch normalization is performed to balance the residual branch size related to jumps. Therefore, the residual network and AM Softmax cross-entropy loss function are used, as shown in Figure 4.

The AM Softmax cross-entropy loss function is used to enable the model to learn the characteristics of large distances between classes, and improve the accuracy of the model.

Softmax is a normalized exponential function [35]. It is a normalization function because its denominator is like $\sum_{i=1}^N e^{w_i x_i}$; also, it is an exponential function because it can be regarded as an $ef(x)$ function [36]. Softmax is used to limit its value to (0, 1), and it expands the difference between allocations through the exponential function. The Softmax function is shown in equation (9):

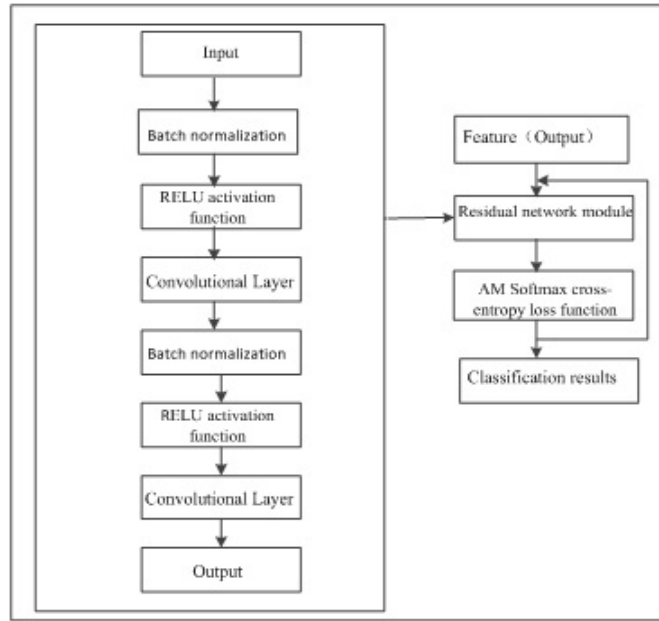


Figure 4 Residual Network Model Process.

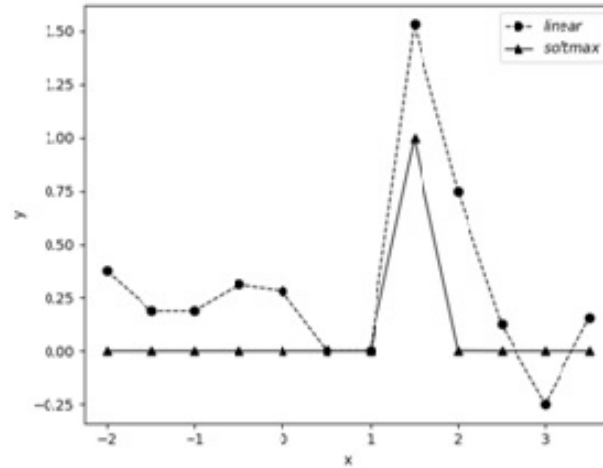


Figure 5 Linear and Softmax functions.

$$S(y_j) = \frac{e^{w_j x_j}}{\sum_{i=1}^N e^{w_i x_i}} \quad (9)$$

In Figure 5, through the Softmax function, the maximum value is changed, and the non-maximum value is weakened. This feature is very suitable for classification because, in multi-classification, each input corresponds to only one category, which helps to find the expected category and weaken the unwanted category, thereby finding users that match the sensing task.

The AM Softmax function belongs to the metric learning strategy of reducing intra-class distance and increasing inter-class distance. In Figure 6, Softmax can distinguish only the boundary between categories (as shown by the green dashed line), while the AM Softmax function can reduce the intra-class distance and increase the class spacing, thereby narrowing the class interval to the target region range while generating margin-sized class spacing [35].

The AM Softmax formula is shown in (10):

$$\begin{aligned} AMS(y_j) &= \frac{e^{s*(L2(w_{y_j}^T)L2(x_j)-m)}}{e^{s*(L2(w_{y_j}^T)L2(x_j)-m)} + \sum_{i,i=j}^N e^{s*L2(w_{y_i}^T)L2(x_i)}} \\ &= \frac{e^{s*(\cos \theta_{y_j} - m)}}{e^{s*(\cos \theta_{y_j} - m)} + \sum_{i,i=j}^N e^{s*\cos \theta_i}} \end{aligned} \quad (10)$$

where $AMS(y_j)$ represents the AM Softmax function; $\cos \theta_{y_j}$ represents x_j in the category of y_j ; $\cos \theta_{y_j} - m$ indicates that there is at least a distance of m between the regions of the categories. The value of m needs to consider whether there are clear boundaries in the allocation of data in reality. If the boundaries are clear, the value of m can be set larger. If the boundaries are not clear, the value of m needs to be set smaller according to the actual situation. $s*$

$(\cos \theta_{y_j} - m)$ denotes the conversion of $\cos \theta_{y_j}$, The function value increases by s times because the cosine value

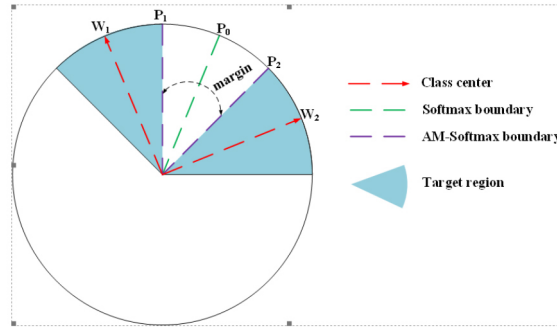


Figure 6 Softmax and AM Softmax functions.

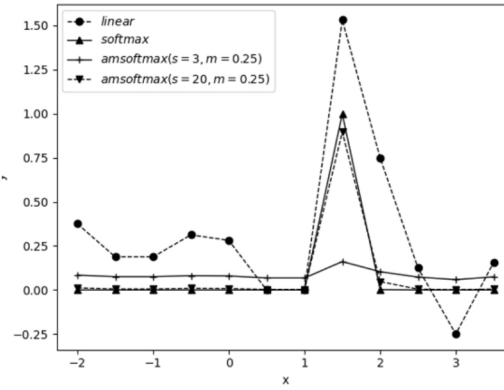


Figure 7 Linear functions, Softmax, and AM Softmax functions.

interval is $[0,1]$, which is too small to effectively distinguish differences. After increasing s times, Softmax can improve the difference of distribution, produce an obvious "Matthew effect", and improve the convergence rate.

Setting the hyperparameters in Figure 7 can effectively increase the allocation differentiation, achieving a nearly Softmax effect. In the experiment, setting s to 30 leads to a better effect; If s is too small, there is no significant difference in allocation, which is not conducive to gradient descent. The AM Softmax function adopts an additive interval margin strategy to feature [37] and normalize the target logit of the Softmax loss. The loss function is based on the previous margin scheme, but it is simpler and more interpretable. Compared to A-Softmax, which can apply only unfixed angular spacing, it applies fixed angular spacing. Also, m directly acts on the cosine distance. The AM Softmax cross-entropy loss function is shown in (11):

$$L_{AMS} = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{s \cdot (\cos \theta_{y_i} - m)}}{e^{s \cdot (\cos \theta_{y_i} - m)} + \sum_{j=1, j \neq y_i}^c e^{s \cdot \cos \theta_j}} \quad (11)$$

As the number of layers in the neural network gradually increases, the model may encounter problems such as gradient vanishing and performance degradation. The prediction model uses the residual network algorithm and batch normalization processing method [38] to solve these two problems. The residual network was proposed by He K et al. [39] in 2015. Compared with other networks, its most obvious feature is the addition of a jump structure. This structure

allows the information from the previous layer in the model to be directly transmitted to the subsequent layers, thereby effectively solving the problem of performance degradation caused by the increase in the number of model layers [40]. The AM Softmax cross-entropy loss function is shown in Formula (11).

$$L_{AMS} = -\frac{1}{M} \times \sum_{i=1}^M \log \frac{e^{s \cdot (\cos \theta_{w_i, x_i} - m)}}{e^{s \cdot (\cos \theta_{w_i, x_i} - m)} + \sum_{j=1, j \neq w_i}^z e^{s \cdot \cos \theta_{j, x_i}}} \quad (12)$$

The parameters in equation (12) are listed in Table 1:

4.2 User Activity Crowdsensing Task Allocation Based on The ResNet Neural Network, Uactdbnn

This section introduces a user-active sensing task allocation scheme based on the ResNet neural network. First, a model of the relationship between sensing tasks and users is established. Then, the correlation model between sensing users is studied, and a positivity model between users is established based on their completion of historical tasks in sensing activities. To improve the accuracy of sensing tasks, the sensing data is used to train the residual network model to eliminate the gradient problem. Meanwhile, to reduce the distance between the same type of tasks and lengthen the

Table 1 The Parameters in Formula (12).

symbol	define
x_i	input of the i -th training sample in the last fully connected layer
θ_{j,x_i}	the angle between the j th type of sensing data and x_i in the final fully connected layer weight
w_i	real data of the i -th training sample
M	total number of training samples
m	hyper-parameters
s	hyper-parameters

Algorithm 1 User Activity Crowdsensing Task Allocation Based On Resnet Neural Network——UACTDBRNN

Input:Initial activity for initializing smart devices

Output:Ab[n], Aa[n], Indicates user activity before and after processing

1: procedure UACTDBRNN (Ab, Aa)

2: initialization Ab ,TS←0

3:for(i=1; i < n; i++)

4: for (j=1; j < m; j++)

5: activity of computing intelligent devices;

6:calculate the activity between intelligent devices;

7: end for

8: end for

9: for (i=1; i < n; i++)

10: calling residual network algorithms to train intelligent devices;

11 if(activity within the scope of sensing tasks)

12: select Sensing Users;

13: endif

14:endifor

15: return Ab [i] , Aa [i]

16: endUACTDBRNN

distance between different classes, the AM Softmax cross-entropy loss function is used for normalization, making it possible to accurately select suitable users for sensing tasks.

4.2.1 Algorithm Description

In mobile sense networks, the allocation of sensing tasks is random. If a certain association relationship between users can be found from trajectory data, it can improve the accuracy of sensing task allocation. The rapid development of the Internet has made it difficult for users in the network to trust each other. In terms of sensing users who can establish connections or have successfully collaborated, the opportunity for cooperation once again will be greatly enhanced. This paper proposes the UACTDBRNN algorithm to establish a positive model among users, accurately identify users who can complete the sensing task, enhance the likelihood of sensing users accepting the sensing task, and ensure the completion of the sensing task; To improve the accuracy of sensing task allocation, the residual network model is introduced and trained with the sensing data. Meanwhile, the AM Softmax cross-entropy loss function is used to normalize the data. The accuracy of the algorithm was measured from multiple perspectives in the experiment.

4.3 Algorithm performance analysis

The Time complexity of the UACTDBRNN algorithm is $O(n^2)$, and its space complexity is $O(n)$, where n and m

respectively represent the number of published tasks and the number of users receiving the sensing task.

In algorithm 1, the time complexity of the loop in lines 3 to 8 is co-determination by the number of participants sending sensing tasks and the number of participants receiving tasks. The loop in lines 9 to 14 indicates that the time complexity of task time supervision is $O(n)$. In the UACTDBRNN algorithm, when the number of senders and receivers tends to n , the number of algorithm cycles is n^2 , and the time complexity of the UACTDBRNN algorithm is $O(n^2)$.

When the number of sensing tasks and the number of users participating in the sensing task are both n , the variable that needs auxiliary storage in algorithm 1 is user activity, and the auxiliary space that needs storage is Ab[n] and Aa[n]. Therefore, the space complexity of the algorithm is $O(n + n + O(Abn) + O(Aan))$, so the total space complexity is $O(n)$.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Experimental Setup

(1) Dataset

The experiment was conducted on a Dell desktop computer (i7-11800 CPU, 16 GB main memory, 1TB SSD+1T disk) using Java as the development language and Eclipse as the development platform. The algorithm's attention was

Table 2 The Parameters of the AM Softmax cross-entropy loss function.

Convolutional kernel size	Number of convolutional kernels	strides	AM-Softmax	
			hyper-parameters	hyper- parameters
			s	m
2×2	16	1	30	0.25

validated using a real dataset provided by MIT Reality Mining [41]. This dataset records 97 smart devices carried by MIT faculty and students, including information such as movement time, task contact location, call duration, and user movement location for about 9 months. There are approximately 1 million records, and approximately 400000 trajectory data were selected for the experiment. Some of the devices have many smart devices connected to them, while others have a small number of connected smart devices, leading to a significant difference in the number of connected sensing devices between the two. Therefore, some data were deleted. Meanwhile, very small or large data (such as data with user activity less than 0.005 or greater than 10) were removed from the experimental results set, and the sensing user and task numbers were added to facilitate data organization. To collect data for the same task, 45 smart devices were finally selected to verify the experimental results.

(2) Evaluation indicators

It is crucial to verify the accuracy of algorithm allocation tasks. Mean absolute error (MAE) is used as one of the evaluation metrics [42], [43]. This method measures the accuracy of the algorithm by the average error between predicted and actual values. The smaller the MAE value, the higher the accuracy of the algorithm. MAE is calculated with:

$$\theta_{MAE} = \frac{\sum_{i \in A_{test}} |p_i - q_i|}{n} \quad (13)$$

where p_i represents the predicted value of the target task, q_i represents the actual value of the target task, A_{test} represents the set of sensing task tests, and n represents the number of sensing task test sets.

The performance of the UACTDBRNN algorithm is measured for accuracy, precision, recall, F-measure, and runtime [44]. In these metrics, true positive (TP) represents the number of positive samples predicted by the model as positive; true negative (TN) represents the number of negative samples predicted by the model as negative; false positive (FP) represents the number of negative samples predicted by the model as positive; false negative (FN) represents the number of positive samples predicted by the model as negative.

① The accuracy, Acc, is defined as the percentage of correctly classified instances in the total number of instances.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (14)$$

② The precision, P, is the total number of true cases divided by the sum of false positive cases and true cases. The higher the P-value, the lower the false positive case.

$$P = \frac{TP}{TP + FP} \quad (15)$$

③ The recall rate, R, is the sum of true examples divided by false counterexamples and true examples.

$$R = \frac{TP}{TP + FN} \quad (16)$$

④ F-measure is the harmonic mean of P and R, which is widely used in information retrieval and other supervised machine learning tasks. When P and R cannot be achieved simultaneously, F-measure is commonly used, and the larger the value, the better.

$$F = \frac{2P \times R}{P + R} \quad (17)$$

⑤ The Run time, T_{run} , denotes the time interval from the beginning time T_{start} to finishing time T_{finish} required to run a single experiment.

$$T_{run} = T_{start} - T_{finish} \quad (18)$$

(3) The parameters of the AM Softmax cross-entropy loss function are listed in Table 2, where s and m are two super parameters. If the value of m is too large, the model has difficulty converging, and m is set to 0.25 in the algorithm.

5.2 Experimental results and analysis

(1) θ_{MAE} impact

The accuracy of sensing task allocation θ_{MAE} when the values of β are 1, 2, and 3, is shown in Figure 8.

As shown in Figure 8, for any number of users, when $\beta = 2$, the allocation accuracy of the proposed algorithm reaches its optimal level, which is the difference in the sensing task activity between any two sensing users. This is mainly because when the value of β is high, the sensing number of successful user interactions increases; when the value of β is small, the sensing number of successful user interactions decreases, causing positive results to deviate from the mean.

The value range of the positivity parameter θ in equation (7) is (0-1), and its impact on the accuracy of sensing task allocation θ_{MAE} is shown in Figure 9.

In Figure 9, when the value of d is too large or too small, the proposed algorithm does not perform well in terms of task allocation accuracy. When $d = 0.6$, θ_{MAE} reaches the minimum value, indicating that when the positivity value is 0.6, the algorithm achieves the best allocation accuracy. If $d = 0$, positivity equals indirect positivity, and the sensing positive relationship between the task receiver and publisher

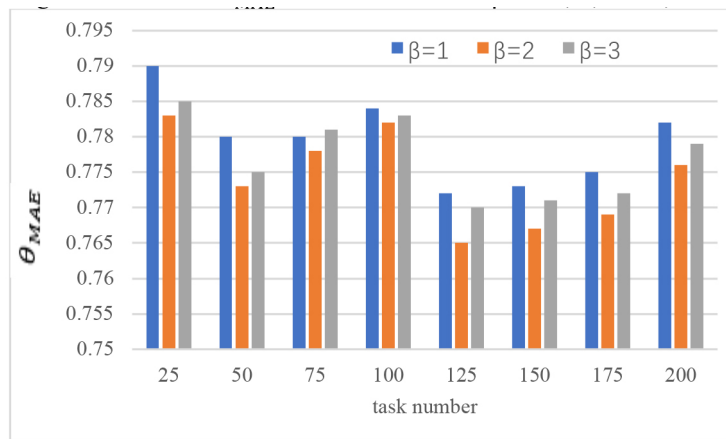


Figure 8 The impact of different β value pairs on θ_{MAE} .

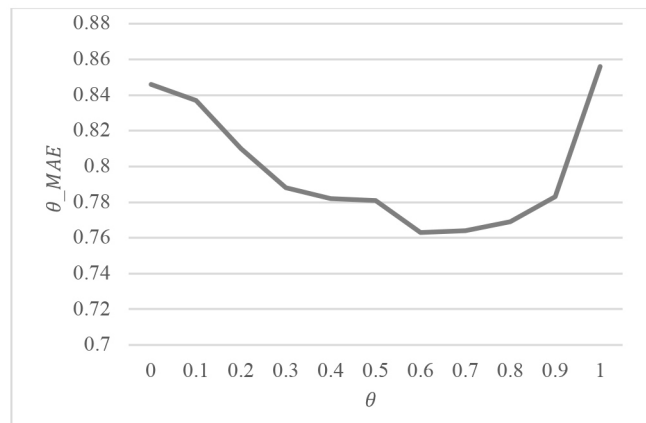


Figure 9 The impact of parameter d on θ_{MAE} .

is not strong, resulting in a deviation from the mean; If $d = 1$, positivity equals direct positivity, and there are not many sensing users directly related to the allocation of sensing tasks. Once certain sensing users have a large impact on overall positivity, it will cause the accuracy of task allocation to deviate from the mean.

(3) Comparison of classification performance

Accuracy is one of the basic indicators of a classifier. To evaluate the effectiveness of basic residual network algorithms, experiments were carried out on the MIT Reality Mining dataset using multiple sense data ranging from 100 to 500 to evaluate the classification accuracy. Figure 10 presents the classification accuracy for different training scales. By comparing and analyzing the random algorithm, maximum algorithm, and NearstFirst algorithm [45], it can be seen that the proposed algorithm obtains 0.8% higher accuracy than the existing random algorithms and about 0.4% higher accuracy than the NearFirst algorithm.

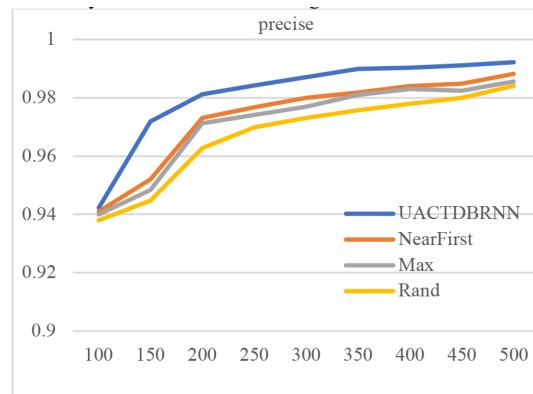
Figures 10(a) and 10(b) demonstrate that the UACT-DBRNN algorithm performs better in terms of accuracy and recall compared to random algorithms:Max algorithms and NEARFIRST algorithms. The classification in the UACTDBRNN algorithm becomes more accurate as the number of features used increases. Meanwhile, there are more accessible tasks that have a higher level of activity to choose from, which also improves the indicator value.

In the experiment, the recall rate of the residual network algorithm is significantly higher than that of other basic algorithms. This is because the residual network and the cross-entropy function can more accurately distinguish different categories. A high recall rate can match most sensing tasks, while only a few sensing tasks are misclassified. Figure 10(d) shows the comparison of the F-measure between the proposed algorithm and other existing algorithms. It can be seen that the proposed algorithm has a slightly higher F-measure than other methods.

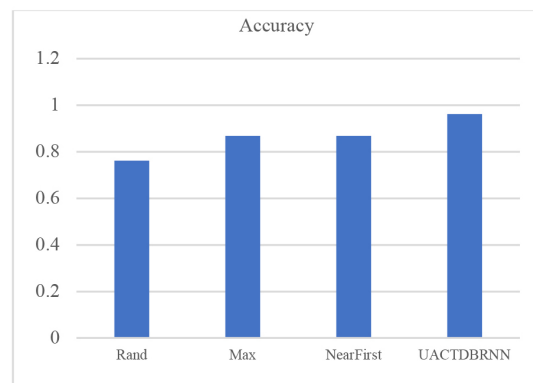
The trained residual network algorithms can provide good matching for distributed sensing tasks. Since matching is done in different classes, the time is greatly reduced, but the NEARFIRST algorithm spends more time on training. As shown in Figure 11, in the process of classification, the combination method of the residual network and AM Softmax cross-entropy loss function is used, which fully guarantees the accuracy of classification.

6. SUMMARY

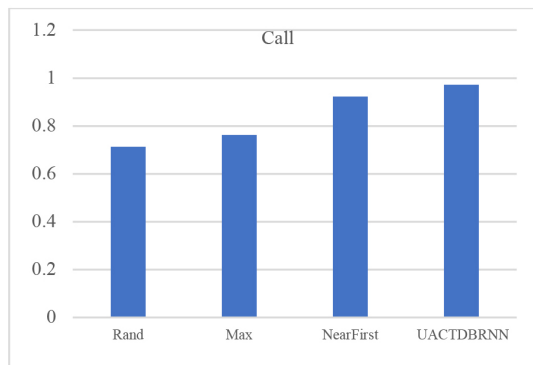
To improve the efficiency and accuracy of sensing task allocation, multi-dimensional research is conducted on the NEARFIRST algorithm. A model of user engagement in the sensing task is established, a user engagement task allocation



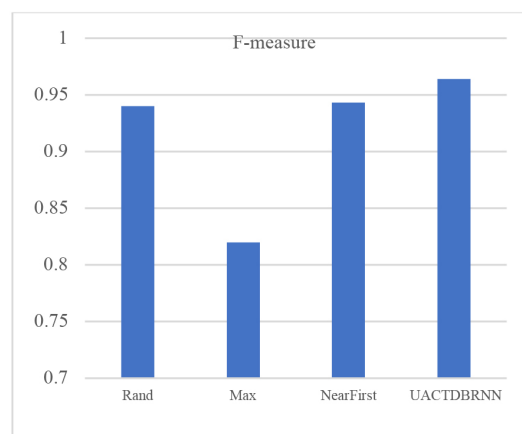
(a)



(b)



(c)



(d)

Figure 10 (a) The accuracy of different sensing tasks. (b) The accuracy of different algorithms. (c) The recall rates of different algorithms. (d) F-measure of different algorithms.

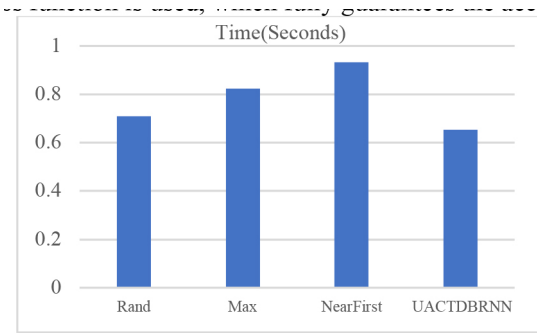


Figure 11 The running time of different algorithms.

algorithm is proposed based on the ResNet neural network, and the superiority of the NEARFIRST algorithm is verified through comparative experiments. Experimental results show that the UACTDBRNN algorithm proposed in this article improves classification accuracy by 6.3% compared to the Max algorithm and 2.4% compared to the NEARFIRST algorithm. In the future, we will continue to investigate the factors that have a negative impact on sensing task allocation, to improve accuracy, and better identify users who can complete sensing tasks.

7. FUNDING

This work was supported in part by the National Natural Science Foundation of China (61572340), Advanced Research fund (61403120402), 2022 Jiangsu Province Entrepreneurship and Entrepreneurship Doctoral Program, The Application Foundation Research of Suzhou of China (SYG201730), Science and Technology Project of Xuzhou of China (No.SNG2020073, SS202023), Research on Task Allocation Methods for AI-Based Crowdsensing 2022 Provincial "Double Innovation Plan" Doctoral Special Program. 2022 annual research startup funding for high-level talents 2408006/064,2408006/058; Research on Collaborative Decision-Making Technology in Crowd-Sensing Networks Based on Artificial Intelligence 2110011/106, Research on the Application of Intelligent Algorithms in System Control Strategies and Optimal Allocation of Resources 2110011/109.

REFERENCES

1. A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich and P. Bouvry, A Survey on Mobile Crowdsensing Systems: Challenges, Solutions, and Opportunities, in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2419-2465, thirdquarter 2019.
2. Gigwalk, 2019. [Online]. Available: <http://www.gigwalk.com/>.
3. Jana, 2019. [Online]. Available: <https://www.jana.com/>.
4. Weichaishi, 2019. [Online]. Available: <http://www.weichaishi.com/>.
5. Z. Xu et al., Multi-Modal Description of Public Safety Events Using Surveillance and Social Media, in *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 529-539, 1 Dec. 2019.
6. J. Wu, L. Hong and V. Frias-Martinez, Predicting Sensing Cycling Safety Levels Using Open and Crowdsourced Data, 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 1669-1676.
7. P. Bora, M. K. Choudhury, J. A. Valan and J. C. Dhar, A Cost-effective Mobile Sensing System intended for Continuous Monitoring of Vehicular Emissions for Air Quality Monitoring Applications, 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, India, 2018, pp. 1-6.
8. P. Shelar and U. Gogate, Performance Evaluation of FMAC Protocol in Healthcare Monitoring System based on Wireless Sensor Network, 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), Aurangabad, India, 2020, pp. 100-104.
9. S. He and K. G. Shin. "Steering crowdsourced signal map construction via Bayesian compressive sensing," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 1016-1024.
10. A. Mansour et al. Everywhere: A Framework for Ubiquitous Indoor Localization, in *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5095-5113, 15 March 15, 2023.
11. W. Liu, Y. Yang, E. Wang, Z. Han, and X. Wang, Prediction based user selection in time-sensitive mobile crowdsensing," in 2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), June 2017, pp. 1-9.
12. H. Wang, D. Zhao, H. Ma and L. Ding. MB-GVNS: Memetic Based Bidirectional General Variable Neighborhood Search for Time-Sensitive Task Allocation in Mobile Crowd Sensing, in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2219-2229, Feb. 2020.
13. B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han. ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems, *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 392-403, Jun. 2017.
14. Y. Wu, L. Tang, S. Yuan. Semantic segmentation model of multi-source remote sensing images was used to extract winter wheat at tillering stage[J]. *Scientific Reports*, 2025, 15(1). DOI:10.1038/s41598-025-98449-3.
15. M. Xiao, J. Wu, H. Huang, L. Huang, and C. Hu. Deadline-sensitive user recruitment for probabilistically collaborative mobile crowdsensing, in 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), June 2016, pp. 721-722.
16. H. Chen, B. Guo, Z. Yu, and Q. Han. Crowdtracking: Real-time vehicle tracking through mobile crowdsensing, *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7570-7583, 2019.
17. Q. Tao, Y. Tong, Z. Zhou, Y. Shi, L. Chen and K. Xu. Differentially Private Online Task Assignment in Spatial

- Crowdsourcing: A Tree-based Approach, 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 2020, pp. 517-528.
18. W. Jiang, S. Lyu. Dynamic Multi-Task Allocation Method for Passenger Diffusion in Mobile Crowd Sensing[J]. *Electronic Journal: English Edition*, vol. 30, n. 5, pp.940-946, September 2021.
19. E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang. An efficient prediction based user recruitment for mobile crowdsensing, *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 16–28, 2018.
20. S. Wang, H.f. Jiang, C.G. Tang, etc. Task allocation method based on two-stage resource balance in cloud edge collaborative mobile group intelligence sensing [J/OL]. *Small Microcomputer Systems*: 1-10[2023-07-18].
21. J. Wang, J.X. Liu, G.S. Zhao, Z.N. Zhao. Task Recommendation Method Based on Collaborative Sorting in Mobile Group Intelligence Sensing [J]. *Journal of Electronic Science*, 2021, 049(010): 2012-2019.
22. W.J. Jiang, J.P. Chen, W.Q. Zhang et al. Selection of Participants for Multi task Allocation Based on Task Quality in Mobile Group Intelligence Sensing [J]. *Control and Decision*, 2022, 37(10): 2667-2676.
23. P. Wang et al.Graph Optimized Data Offloading for Crowd-AI Hybrid Urban Tracking in Intelligent Transportation Systems, in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1075-1087, Jan. 2023.
24. R. Estrada, R. Mizouni, H. Otrók, A. Ouali and J. Bentahar. A Crowd-Sensing Framework for Allocation of Time-Constrained and Location-Based Tasks, in *IEEE Transactions on Services Computing*, vol. 13, no. 5, pp. 769-785, 1 Sept.–Oct. 2020.
25. M. H. Cheung, F. Hou, J. Huang and R. Southwell. Distributed Time-Sensitive Task Selection in Mobile Crowdsensing, in *IEEE Transactions on Mobile Computing*, vol. 20, no. 6, pp. 2172-2185, 1 June 2021.
26. H. Du, Z. Yu, B. Guo et al. GroupShop: monitoring group shopping behavior in real world using mobile devices[J]. *Journal of Ambient Intelligence and Humanized Computing*, Vol.14, no.5, pp.6367-6378, May 2023.
27. Y. Luo, Z. Yu, H. Yin et al. Multi-agent mobile crowdsensing by pervasive machines: a robust task allocation approach[J]. *CCF Transactions on Pervasive Computing and Interaction*, 2022, 5(1):13-30.
28. M.Y. Yao, D. Tao, J.T. Wang, R.P. Gao, Kunning-Sun:MARVAir: Meteorology Augmented Residual-Based Visual Approach for Crowdsourcing Air Quality Inference. *IEEE Trans. Instrum. Meas.* 71: 1-10 (2022).
29. J.Y. Wu, Z.Y. Zhang, J.Y. Han. Participant Selection Method Based on User Willingness in Mobile Group Intelligence Sensing [J]. *Journal of Northeast Normal University (Natural Science Edition)*, 2021, 53(03): 72-80.
30. Z.G. Zheng, J. Liu, P. Wang.Time Weighted Uncertainty Nearest Neighbor Collaborative filtering Algorithm [J]. *Computer Science*, 2014, 41(08): 7-12.
31. R.Q. Wang, Y.I. Jiang, Li Y. X., and J.G. Lou. A Collaborative filtering recommendation algorithm based on multiple social trust [J] *Computer Research and Development*, 2016, 53(6): 1389-1399.
32. X.Q. Qiao, C. Yang, X.F. Li et al. A trust calculation method based on user context in social network services [J] *Journal of Computer Science*, 2007, 30(11): 1930 1939.
33. C. You. Research on Cost-sensitive Classification Methods for Imbalanced Data[A]. 2021 International Conference on Artificial Intelligence, Big Data and Algorithms[C]. Xi'an: IEEE, 2021.
34. Y.Q. Y, F. Lei, W.J. Liu. Ensemble Additive Margin Softmax for Speaker Verification[A]. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing[C]. Brighton: IEEE, 2019.
35. R. Qiang, Y. Bing, H. Kun, et al. Regularization and Iterative Initialization of Softmax for Fast Training of Convolutional Neural Networks[A], 2019 International Joint Conference on Neural Networks[C]. Budapest: IEEE, 2019.
36. S. Zhang, Y.H. Gong, and J.J. Wang. The Development of Deep Convolutional neural network and Its Application in Computer Vision [J] *Journal of Computer Science*, 2019, 42(3): 453-482
37. W. -F. Ou et al., LinCos-Softmax: Learning Angle-Discriminative Face Representations With Linearity-Enhanced Cosine Logits, in *IEEE Access*, vol. 8, pp. 109758-109769, 2020.
38. V. Thakkar, S. Tewary and C. Chakraborty, Batch Normalization in Convolutional Neural Networks — A comparative study with CIFAR-10 data. 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT), Kolkata, India, 2018, pp. 1-5.
39. K. He, X. Zhang, S. Ren et al. Deep residual learning for image recognition[A]. *IEEE Conference on Computer Vision & Pattern Recognition*[C]. Las Vegas: IEEE, 2016.
40. C.Kang, Z. Wei, R. Zheng, et al. Fabric defect detection method based on deep residual network[J]. *Journal of Chinese Computer Systems*, 2020, 41(4): 800-806.
41. N. Eagle, A. S.Pentland. Reality mining:Sensing complex social systems[J]. *Personal and Ubiquitous Computing*, 2006, 10(4): 255-268.
42. H.Y. Wang, W.B. Yang, S.H. Wang et al. A Service Recommendation Method Based on Trusted Alliances [J] *Journal of Computer Science*, 2014, 37(2): 301-311.
43. Q. Wang, J.H. Wang. Collaborative filtering recommendation algorithm combining trust mechanism and user preference [J] *Computer Engineering and Application*, 2015, 51(10): 261-265.
44. B. Alexander, B. Maximilian, Z. Gerald. The Relative Confusion Matrix, a Tool to Assess Classifiability in Large Scale Picking Applications[A]. 2020 IEEE International Conference on Robotics and Automation[C]. Paris: IEEE, 2020.
45. Guo B., Liu Y., Wu W., Yu Z.W, Q. Han.ActiveCrowd: A Framework for Optimized Multi-Task Allocation in Mobile Crowdsensing Systems. *IEEE Transactions on Human-Machine Systems*, vol. 47, no.3, pp. 392-403, June. 2017.

