IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 16, 2020, accepted June 25, 2020, date of publication June 29, 2020, date of current version July 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005543

A Framework for Optimal Worker Selection in Spatial Crowdsourcing Using Bayesian Network

NOR ANIZA ABDULLAH[1], (Member, IEEE), MOHAMMAD MUSTANEER RAHMAN[1], MD. MUJIBUR RAHMAN[1], AND KHAIRIL IMRAN GHAUTH[2]

[1]Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[2]Faculty of Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63100, Malaysia

Corresponding author: Nor Aniza Abdullah (noraniza@um.edu.my)

This work was supported by the University of Malaya Research Grant under Grant RP032D-16SBS.

ABSTRACT Spatial Crowdsourcing (SC) is a new paradigm of crowdsourcing applications. Unlike traditional crowdsourcing, SC outsources tasks to distributed potential workers, and those who accept the task are required to travel to a predefined location to complete it. Currently, the primary aim of SC is to maximize the number of matched tasks or to minimize the travelling distances of the workers. However, less focus is given in matching the right tasks to the right workers, particularly in a heterogeneous tasks environment. To address this lacking, our work provides an efficient framework for selecting optimal workers for every task with various specification (geographical proximity, domain types, and expiration times), based on workers' attributes (task domain-specific knowledge, expertise or performance history, distance to task location, and task workload distribution). We introduce the use of Bayesian Network in modelling and selecting optimal workers, and use k-medoids partitioning technique for tasks clustering and scheduling. Our experimental results on both synthetic and real-world large datasets have shown that our proposed approach has outperformed other baseline approaches, in terms of low average error rate and fast execution time.

INDEX TERMS Spatial crowdsourcing, worker selection, Bayesian network, task allocation, task matching accuracy, computational efficiency.

I. INTRODUCTION

Crowdsourcing refers to an emerging distributed problem-solving paradigm that incorporates the power of both human computations and machine intelligence. It is an arrangement where public crowds contribute towards solving requested tasks in the form of open calls with some incentives [1], [2]. Many emerging online crowdsourcing platforms such as Amazon's Mechanical Turk (mTurk) provides commercial crowdsourcing services [3]. However, online crowdsourcing may not be suitable when the tasks need to be completed at a predetermined physical location [4]. As the alternative, Spatial Crowdsourcing (SC) is introduced [1]. SC is a crowdsourcing platform that outsources different types of spatio-temporal tasks to workers, wherein spatial data is required to enable a worker to travel to a physical location to complete the task. The spatial data include location, time, mobility, and contextual information.

The associate editor coordinating the review of this manuscript and approving it for publication was Longxiang Gao.

In SC, a requester submits task's specification to a SC-server, for example, a task to capture pictures of the flood-affected area around Penang, Malaysia. The server then crowdsources the task among available workers in the crowd. Once the worker has accepted the task, he/she will travel to the predetermined location to perform the task. Once the worker has completed the task, and documented the event, the requester is notified of the task accomplishment. The worker is rewarded accordingly. Currently, there are numerous applications in the SC platform, ranging from several domains such as tourism, intelligence, disaster response, journalism, general labour, and urban planning [5]. SC has stimulated a series of recent industrial successes such as Citizen Sensing (Waze), P2P ride-sharing (Uber), Real-time Online-To-Offline (O2O) services (Instacart and Postmates) and on-demand staffing service platforms (Wonolo).

There are two modes of tasks assignment in the SC platform, which are Worker-Selection (WS) and Server Assigned Task (SAT). In WS, a worker chooses a task from the published tasks on the server [6]. The server does not have full

control over the task assignment [1]. As a result, there is a probability that unpopular tasks could be left unassigned. On the other hand, in the SAT mode, the server assigned tasks directly to the workers. SAT mode is more popular than the WS due to its ability to optimise available resources by offering control over the workload balancing for each worker, hence increases the task assignment rate [1], [7].

In both modes of task assignment, optimal worker selection is an important issue [1], [8]–[10], because allocating inappropriate worker for a task may negatively affect the quality of the completed task. We define this assignment problem as a matching problem between tasks and workers. Currently, the main objective of most of the existing task matching solutions is to maximize the number of matched tasks [11] or to minimize worker's travelling distances to task location [12]. Additionally, most of the existing worker model for task allocation assume that workers' knowledge and expertise are independent of tasks [1], [13]. Nonetheless, in a heterogeneous SC, there will be various types of tasks, and each task belongs to diverse domains (such as observations, operations, general labour, drivers, cleaning and event staffing), and workers may have the ability to work in more than one domains. However, each worker may possess different levels of knowledge and expertise for tasks from different domains, and higher the expertise in a particular domain, higher the expectancy of task completion [1]. Consequently, the domain-specific knowledge and the expertise of the workers are considered in this study, which plays a crucial role in selecting optimal workers for task allocation. Moreover, in SC, the total travelling distance that a worker requires to travel from his location to the location of the task is related to his cost of travel which ultimately influences his willingness to travel to the task location in order to perform it [14], [15]. For instance, one would end up rejecting a task request where he is required to travel long distances just to solve a task [15]. Hence, the total travelling distance should be minimised. Furthermore, for the newly registered users, it is necessary that the SC platform assigns tasks to them as soon as possible as they enter into the system [15]. This will encourage the new workers and will also ensure equal opportunities for the platform workers to maintain their long term participation, enabling the SC platform to optimize its resources, minimize task completion time, and meeting the real-time demand of task assignment [1], [9] [16]. As a result, the task workload distribution among the workers shows a decisive role in worker selection in task matching which should be improved [1], [15]. Therefore, it is imperative to model an optimal worker selection mechanism that will depend on three of the most fundamental factors based on the dedicated task requirements: i) domain-specific knowledge and expertise to identify the expert workers in a particular domain, ii) distance to task location to minimise the workers' travelling cost, and iii) task workload distribution to improve the workload balancing among the platform workers. Besides these factors, the mechanism should be computationally efficient to meet the real-time matching demands on SC platform.

This study proposed a framework to improve the efficiency of task matching by considering workers' attributes and tasks' specifications prior to select optimal workers for a specific task. Therefore, the ultimate aim of the framework is to allocate the right task to the right worker. The primary components of our proposed task matching framework are as follows:

1) Task clustering and scheduling, in which tasks are clustered according to their types of domains and geo locations, and scheduled by expiration times by using k-medoids algorithm, 2) Optimal workers' selection by using the Bayesian Network model, to enable the right task is allocated to the right worker. The Bayesian Network model is a probabilistic graphical model which is used for reasoning under uncertainty [17]–[19].

The following contributions are the main outcomes of this study:

1. A framework for efficient task matching in heterogeneous SC tasks setting. To the best of our knowledge, our work is the first to address the matching problem as the means to optimize the efficiency of task-to-worker matching.
2. A tasks clustering and scheduling mechanism by using the k-medoids partitioning technique in order to mitigate the difficulty in the task matching with various specifications based on domain types, geolocation proximity, and expiration times.
3. An optimal worker selection approach by using the Bayesian Network model makes it easier to deliver efficiency in matching the right task to the right worker.
4. Simulations and empirical findings on the benchmarking performances against baseline SC task-matching approaches by using large synthetic and real-world datasets.

The rest of the paper is organised as follows: Section 2 presents the related work. Section 3 describes the problem formulation. Section 4 describes the proposed framework. Section 5 reports the experimental results and discussion followed by the conclusion in section 6.

II. RELATED WORK

In SC task allocations, worker selection strategy plays a significant role in the achievement of efficiency, and the quality of the completed tasks [1]. There have been numerous studies on task allocations in SC. Reference [20] proposed a SC platform known as gMission, which supports task publishing and worker recruitment. It allocates tasks based on workers' locations, and tackles the load balancing issue among the platform workers. In the gMission platform, a time-weighted kNN (k-nearest neighbour) algorithm was adapted so as to facilitate the location-based task allocations. Reference [3] proposed a real-time SAT-based SC framework for task allocations by using the Greedy algorithm. The aim of the framework was to maximize the number of assigned tasks under budget constraints. Reference [21] designed a real-time, and budget-aware task allocation mechanism to maximize the

number of assigned tasks, and to improve the expected quality of the completed task under limited budget constraints. The mechanism also considers the distance of each worker from the tasks and the worker's performance in previously assigned tasks. A revised Greedy algorithm was then used to automatically allocate the tasks to the appropriate workers, based on the workers' travelling distances; it also calculates the corresponding rewards for each completed task. Similarly, [22] proposed a reliable diversity-based SC framework which can maximize the task completion rate, and address tasks diversity. The framework is capable to dynamically assigning time-constrained spatial tasks to workers. It also utilises three approximation approaches, including Greedy, sampling, and divide-and-conquer algorithms to assign workers to spatial tasks, with the aim to improve the completion reliability, and the spatial/temporal diversities of spatial tasks. On the other hand, [10] used a location-aware fog platform to identify the most suitable workers by learning their performance history. They proposed a worker selection model using the Greedy-based Upper Confidence Bound algorithm, focusing on learning about the workers' skills. The model can predict worker's performance on each spatial task prior to selecting appropriate worker for the task. The model also aims to improve the long-term utility of the platform.

Reference [23] proposed a framework for optimising task allocation by making sure that each task surpasses its quality threshold. In other words, each task does not exceed the cost limit, and workers are not over-utilised or under-utilised. This is achieved by balancing the workload among the workers. The framework also adopts the Computation of Crowd Indexes Deterministic algorithm, which is based on the Greedy algorithm. The aim is to address the task allocation problem that relates to human factors such as worker's expertise, wage requirements, and worker's availability. In [24], the Greedy approximation algorithm was proposed as a measure to address the QSTA (QoS-Sensitive Task Assignment) problem in task assignment wherein some tasks need to be assigned to a number of workers with minimum total rewards, but without compromising the quality of the performed tasks.

Reference [25] proposed a multi-worker selection mechanism to assign workers to their most preferred tasks. The mechanism adopts the Gale-Shapley Matching Game Selection (GSMS) which is based on game theory, to solve the multi-worker multi-tasking allocation problem in mobile crowdsourcing. Similarly, [12] proposed a Group-based Multi-task Worker Selection (GMWS) framework that allocates a group of workers to a cluster of tasks with the aim of maximising the QoS of tasks and minimising the travelling distance required by the employed workers. GMWS clusters tasks based on their geographical proximity using k-mediods algorithm and deploys genetic algorithm to allocate a group of workers to task clusters. It also incorporates a meta-heuristic approach using tabu search algorithm for scheduling tasks in a cluster for each worker to minimize the completion time. Likewise, [26] presented a group-oriented crowdsourcing framework that outsources tasks to naturally existing worker groups through social networks. The authors proposed two algorithms to select workers, followed by a selection of workers within a group with a leader and selection of workers within a group without leaders. The experimental result reported that the group-oriented approaches could achieve better synergy performance (synergy, consistency, conflict), adaptability, and effectiveness on reducing costs, over individual-oriented and team formation approaches in task allocation.

Reference [27] considered a dynamic participant recruitment problem for heterogeneous spatial crowdsourcing tasks with different temporal and special requirements. The proposed framework could minimise the sensing costs (a fixed cost per selected worker, such as energy cost while being active) while satisfying certain levels of probabilistic coverage (i.e., total task coverage is equal to or larger than a predefined coverage threshold). To address this problem, one offline and two online Greedy algorithms were proposed.

Even though there have been several approaches for spatial task allocations, the following limitations still remained. First, most of the approaches do not consider heterogeneous tasks specifications and workers' attributes for a multiple tasks allocations scenario. Secondly, most of the task allocation models assumed that domain-specific knowledge of all workers is the same regardless of various tasks domains. Nevertheless, in practice, workers may be able to perform tasks from different domains, but they may have different levels of knowledge and expertise for each domain tasks. Finally, current works in SC task allocation have addressed three major optimization problems which are maximizing the number of allocated tasks, maximizing the quality of completed tasks, and minimizing the system costs. However, to the best of our knowledge, none of the studies had attempted to efficiently match the right tasks to the right workers, taking into consideration heterogeneous tasks with various specifications, and specific workers' attributes. This study is, therefore, aims to address this gap.

III. PROBLEM FORMULATION

To illustrate the problem, let us consider a simple SC system with three open spatial tasks (T_1, T_2 *and* T_3), taken from three different domains, involving ten workers available in an area of 5 km $\times$ 5 km. Each task has a specific location, domain type ($C_{T1,A}$, $C_{T2,B}$ and $C_{T3,C}$), and expiration time. Similarly, each worker would demonstrate a certain level of expertise (performance on previously completed tasks) in a particular task domain via a rating score and a task workload balancing score. Table 1 presents the tasks and the workers' list, where tasks are scheduled for allocations. Taking into consideration that the tasks are from different domains, and workers have different attributes, there are two approaches to allocate the available tasks. The first scenario depends on minimising the distance travelled by the workers to perform the task. Each task should be $T_1 < W_5, W_1, W_8, W_3>$, $T_2 < W_{10}, W_6 >$ *and* $T_3 < W_9, W_7, W_4,$

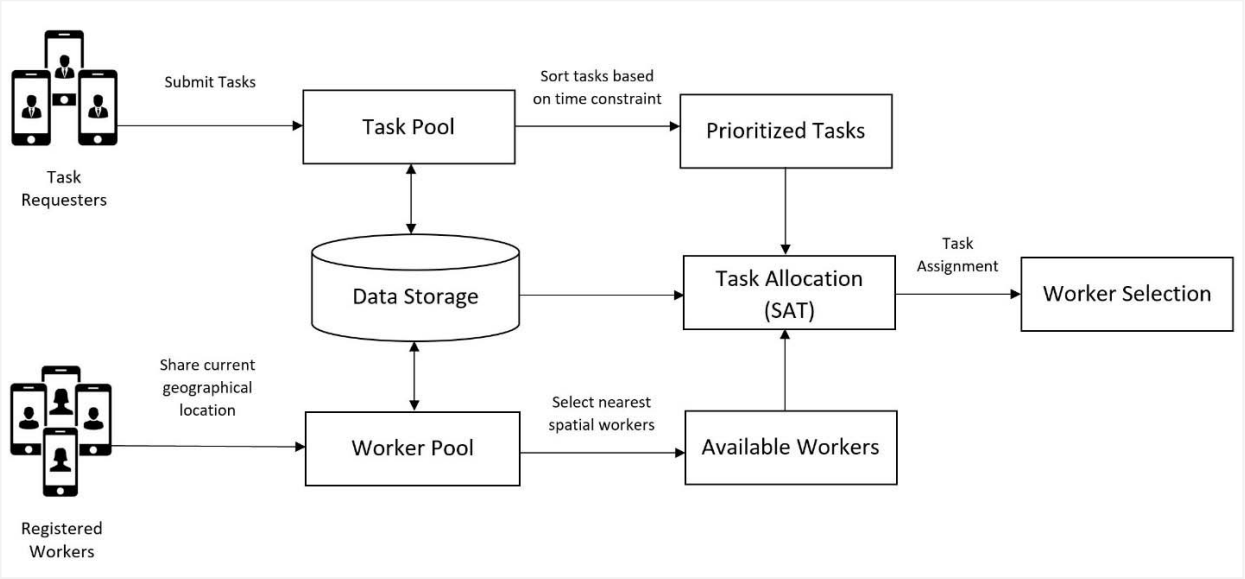

FIGURE 1. Online process diagram.

TABLE 1. Workers' attributes and tasks' specifications.

Task No.	Tasks	Task Domain (C)	Workers (W)	Domain-Specific Knowledge (K)	Expertise (R)	Distance (m)	Workload Balancing (F)
1	T_1	$C_{T1,A}$	W_1	$K_{T1=1}$	4.8	1695	3
3	T_3	$C_{T3,B}$	W_2	$K_{T3=1}$	5	1859	2
1	T_1	$C_{T1,A}$	W_3	$K_{T1=1}$	4	2206	4
3	T_3	$C_{T3,C}$	W_4	$K_{T3=1}$	4.5	1940	0
1	T_1	$C_{T1,A}$	W_5	$K_{T1=1}$	3.8	1650	3
2	T_2	$C_{T2,B}$	W_6	$K_{T2=1}$	4.5	1690	1
3	T_3	$C_{T3,C}$	W_7	$K_{T3=1}$	5	1850	0
1	T_1	$C_{T1,A}$	W_8	$K_{T1=1}$	3.7	1929	1
3	T_3	$C_{T3,C}$	W_9	$K_{T3=1}$	5	1830	3
2	T_2	$C_{T2s,B}$	W_{10}	$K_{T2=1}$	5	1620	5

$W_2 >$ respectively. By only minimizing the travelling distance, the quality of the task matching is not always ensured. The reason is that the quality also depends on workers' other attributes such as their domain knowledge, expertise scores on specific task domains, and task workload distribution scores. In the second scenario, workers' domain-specific knowledge, expertise, and task workload balancing are considered together with the travelling distance. The travelling distance can influence the selection of the right workers for the purpose of maximising the quality and efficiency to complete the tasks while also minimising the distance workers need to travel. Due to the fact that the workers' selection is based on domain-specific knowledge, expertise, distance and workload balancing, the set of probable workers suitable for selection must be ordered as $T_1 < W_1, W_5, W_8, W_3 >, T_2 < W_6, W_{10} >$ and $T_3 < W_9, W_7, W_2, W_4 >$, respectively. Although for T_1, the travelling distance of W_5 is lesser than W_1, W_1 receives a higher priority than W_5 because W1 has a higher performance score than W5. In looking at T_2, it can be seen that even though W_{10} has higher performance score and less distance value than W_6, it was only prioritised next to W_6 due to its high workload balancing score. The reason is due to the need to ensure equal workload distribution across multiple participants. In this case, a higher workload balancing score would result in a lesser probability of being selected. Therefore, for T_3, W_7 was prioritised over W_9, and W_2 was prioritised over W_4. The reason is mainly due to their low load balancing scores.

IV. PROPOSED FRAMEWORK

This study proposes a framework for efficient task matching in SAT-based settings. Figure 1 demonstrates the online diagram of the proposed approach. Figure 2 shows the proposed framework. The requesters send requests for tasks to be completed to the SAT server, i.e. T_1, T_2, T_3, T_4, and T_5. Each request contains information on tasks' specifications such as location, time, type and description. Tasks T_1, T_2 and T_4 are from the same task domain; hence they are represented by the same blue colour. Tasks T_3 and T_4, each representing task from two different domains, with green and yellow colours, respectively. In the server, the task specifications' information is passed to the task scheduler, which is responsible for scheduling the tasks. The data storage stores information regarding the tasks and the workers. It also stores task-to-worker matching histories in support of future task allocations.

The worker module contains information about the registered crowd of workers, such as their personal information, location, domain-specific knowledge, level of expertise (which is based on performance history or rating), and workload balancing score. The information on the geolocation of

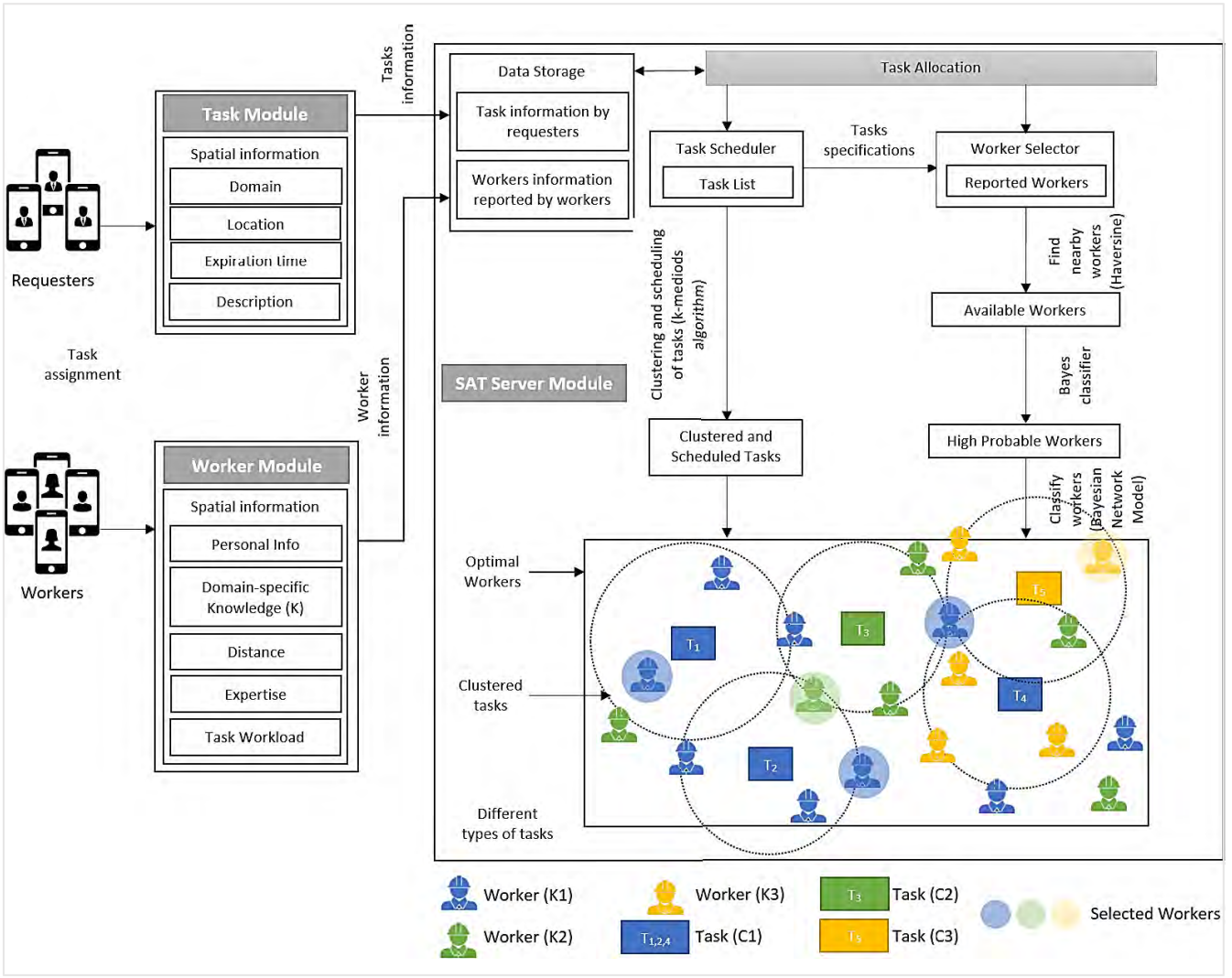

FIGURE 2. **Proposed framework.**

every mobile worker is continuously fed into the workers' module, thus enables the system to find candidate workers who are currently near to the task location. Simultaneously, information on available workers' for the tasks will be fed into the worker selector. The worker selector will first classify the workers based on their attributes' information and task requirements, and then it will determine a set of candidates for optimal workers selection. Once a task is completed, the requester receives the completion report from the system, and both the task's and worker's information is updated accordingly in the data storage. During this time, a crawler is continuously working inside the server to check for any incoming task requests. For every two seconds of task allocation round, it creates a batch of requested tasks, and we define this as a timestamp. It subsequently allocates each of the tasks from the timestamp to suitable workers. If any task is unallocated, it will then be added to the next timestamp. The steps would be repeated until each of the available tasks is being allocated from the timestamp.

The proposed framework for this study features three modules - a task module, a worker module, and a server module. The following section will further describe each of the modules, along with their functionalities.

A. TASK MODULE

The task module is responsible for maintaining information regarding the requested spatial task. It receives incoming task requests from the requesters so as to allocate the tasks to the workers. This module encompasses the following elements.

1) TASK DOMAIN (C)

Tasks are of different types and from different domains. There are various domain-specific location-based tasks [1]. For example, crowdsourced delivery services like 'UberEATS' and 'deliveroo' [28]. These domains are where people get the additional opportunity to work by carrying individuals or objects to be delivered to specific locations. Or, a crowdsensing mobile app that requires people to go to a particular area,

and report any activities that would danger the environment such as illegal waste dumping or polluted river. In this task, people may need to have some knowledge about environmental sustainability and possible hazard that would negatively affect the environment. Hence, we believe it is important for our framework to consider various forms of tasks from various domains, such as general labour, warehouse operations, washing and cleaning, event staffing, and plumbing.

2) TASK EXPIRATION TIME

Tasks are of two types - urgent and normal tasks based on task expiration time [1]. Task time can be calculated from the starting and ending times stated by the task requesters in the task specifications. For normal tasks, a worker can start at a suitable time, and he/she does not need to finish immediately, for example, preparing public facility reports in a natural disaster area for the next few days. In comparison, urgent tasks need to start immediately; for example, a restaurant owner might need a waiter urgently within the next couple of hours on a busy weekend to serve the crowded restaurant. Depending on the calculated task expiration time for each task and other related parameters, task scheduling will be carried out on a collection of tasks.

3) LOCATION

Location represents the geolocation of a requested task. In SC, a worker needs to visit a specific location so as to complete a task. For instance, a worker may need to visit a restaurant at a specific address as requested by the task requester to perform the cleaning task.

4) DESCRIPTION

Each task has a different task requirement or specification, as specified by the requester. The description also incorporates the necessary skills of a worker as well as general duties and responsibilities. It may also state the payment for the completed task.

B. WORKER MODULE

Mobile workers are one of the most important elements in the SC platform, and they are also termed as moving workers [22]. While on the move, they can receive task invitations located in a nearby location. The invitation may include task description, task time, task location and payment. In the proposed framework, the task module captures, stores and maintains information about workers' attributes. This module encompasses the following elements:

1) PERSONAL INFORMATION

To perform the crowdsourcing tasks, users need to register on the platform as platform workers. The registered users provide some personal information, such as their identification numbers, name, age, profile picture, and contact details.

2) DOMAIN-SPECIFIC KNOWLEDGE (K)

Workers may have knowledge of more than one types of tasks, and each task may belong to different domains [1]. Domain-specific knowledge can be derived from the worker's previous work experiences for a specific task domain. In our proposed framework, we use the value of 1 to represent the presence of knowledge, and 0 to represent the absence of knowledge in a specific domain.

3) EXPERTISE (R)

Expertise is referring to the worker's work performance for a particular task in a particular domain. Reference [1] measures worker's expertise based on rating scores upon task completion, which were given by the task's requesters. In our study, a worker's level of expertise is also obtained by analysing his/her performance history. This is calculated by averaging the ratings that represent worker's performance on the accomplished tasks, as rated by the task requester. In our proposed framework, we mapped the result onto a scale of 1 to 5. 0 represents the lowest and 5 represents the highest.

4) WORKLOAD (F)

Workload is a parameter used to represent the current amount of task load of each worker. This information is crucial for the system to monitor and ensure equal distribution of workload among candidate workers for every task in the SC platform. In this study, the workload was calculated from the worker's work history (i.e. the number of tasks completed within a specific time span). The result is mapped onto a scale of 0 to 5.

5) DISTANCE (D)

When a task is posted, the server tracks the location of the reported workers. It then calculates the travelling distances of each worker to the requested task location. In this study, we assume workers who were located within 5000 meters from the task as having a higher chance of being selected than those who were not within the vicinity.

C. SAT SERVER MODULE

The Server Assigned Task (SAT) module is responsible for allocating tasks to optimal workers. When SAT module receives tasks specifications and workers attributes from the task and the worker modules, it will first schedule the tasks. Then it will analyse information both the scheduled tasks and available workers to select the most appropriate workers for every task. The selection process involved two layers of filtering to ensure the right tasks will be allocated to the right worker. Details of the processes are further explained as follows:

1) DATA STORAGE

The data storage stores information on both the requested task specifications and workers' attributes, including their work history and past performances. It received the information from the Task and the Worker modules. The information that is acquired periodically for every batch of tasks is stored and updated in the data storage, prior to further processing by the task allocation component which is mainly comprised of Task Scheduler and Worker Selector.

2) TASK SCHEDULER

Task Scheduler receives a list of tasks from the Task module, which also contains information about tasks specifications provided by the task requesters such as geographical proximity, domain types, and expiration times. Subsequently, it clusters tasks based on the geographical proximities and domain types and schedules them by expiration times (high to low priority) using the k-mediods algorithm. The k-medoids is a classical partitioning technique of clustering; it is reminiscent of the k-means algorithm [12]. The k-medoids have proven to be more robust to outliers due to their ability to minimize the distance between all points in the cluster, and not just between the points and cluster centre, as in the case of other popular clustering algorithms such as k-means [12].

3) WORKER SELECTOR

Worker Selector acquires information of the reported workers and the task specifications from the workers' pool and the task scheduler respectively. Using this information, it detects the available workers for each task based on the calculated distances of the reported workers to the task location within the minimum travelling distance from the task location. This process is accomplished by using the Haversine formula [14]. Subsequently, from the set of available workers, the Worker Selector will continue to identify high probable workers for the task by using Bayes theorem. The identification process will be based on workers' domain knowledge, expertise, travelling distance, and their task workload distribution. From the set of high probable workers, the Worker Selector will execute the Bayesian Network algorithm to select a set of optimal workers for every task. The optimal workers are ranked in decreasing order of importance for the selection. Meaning that those who possess the highest probability value (based on expertise, travelling distance, and task workload balancing) is the most optimal worker for the task, and therefore he/she would be ranked at the top of the list. However, if he/she does not accept the task for a due reason, then the offer goes to the second ranked worker in the list, and so forth. Figure 3 shows the worker's selection process in the matching tasks T_1 and T_2, each from different task domains.

- ***High Probable Workers Selection***. We defined high probable workers as those who possessed a higher level of expertise, shorter travelling distance from the task location and lesser task workload, as compared to other reported workers who are available to work in a particular task domain. In order to select high probable workers for each task in the heterogeneous SC tasks environment, we first classify the available workers into two classes - high probable and less probable workers. Bayes theorem is used for the classification process. The process is based on the following input factors - worker's expertise (R), worker's distance from the task location (D), and worker's workload balancing for a particular task domain (F). These input factors are presented as the events in the Bayes theorem, for assessing the probability of selecting a high probable worker for a task. The success of the classification process will classify workers into two classes (i.e. high and low) for each input factors, i.e., high expertise or low expertise, long distance or short distance, high workload or low workload. For simplicity of the problem space, only high probable workers were considered. It is reasonable to assume that the spatial server would not assign a task to a worker with low expertise value, located far from the task location and has performed the highest number of tasks in a particular time span. In doing so, we set a threshold point for each of the events (expertise, distance, and workload), as represented by equation 1 in the classification, where TP_E is the threshold point for each of the events, E, and N is the total number of available workers. For instance, for the event expertise, the threshold point would be greater than the midpoint of the average of the sample rating scores.

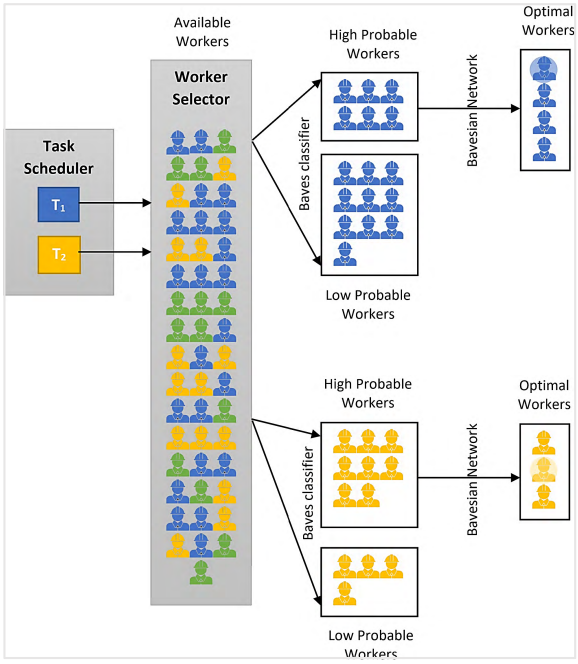

FIGURE 3. Worker selection process in task matching.

$$TP_E > \frac{1}{2}\frac{\sum E_i}{N}; \quad i = 1 \; to \; N \tag{1}$$

After applying Bayes theorem on each event, equations 2, 3 and 4 were derived for expertise, distance, and workload, respectively, as shown below. Note that equations 3 and 4 showed the inverse probability relation. This means that the probability of choosing any worker is higher when distance and workloads were low, while equation 2 indicates that the probability of choosing any worker is higher when the worker has high expertise. In this study, prior belief (prior conditional probability) of each of the events is calculated through simulations done on the experimental dataset. An estimation rule proposed by [29] was adopted for equation 5.

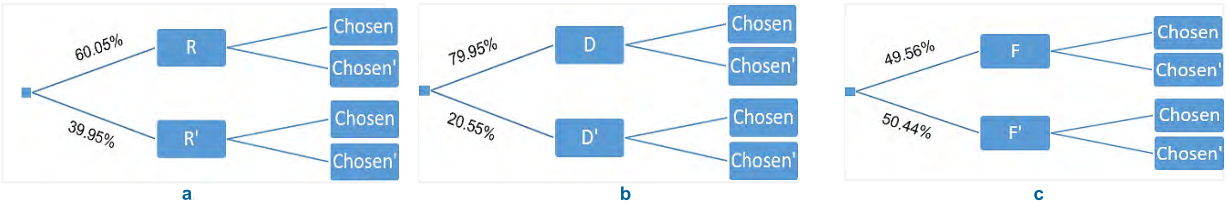

FIGURE 4. (a) Applying Bayes theorem on candidate's expertise. (b) Applying Bayes theorem on candidate's distance. (c)Applying Bayes theorem on candidate's workload balance.

Subsequently, using prior belief, the posterior belief (posterior conditional probability) can be obtained.

In the context of the simulations, five datasets, and each having 10K records of workers (see section 5 that demonstrates the dataset used), were used. Each dataset contains the workers' expertise rating scores in different categories of tasks, travelling distance from the task locations, and the task workload balancing scores. Initially, we assumed that there were two classes - a high score (>50%) and a low score (<50%), for each of the events, as prior belief. After simulating on each dataset, a mean score is achieved from the five datasets for each of the events as prior belief using equation 5. Findings from the simulation results depicted that the chosen worker has:

i 60.05% probability of having high level expertise, and 39.95% probability of having low level expertise.

ii 20.55% probability of being positioned closer to task location, and 79.95% probability of being positioned further from the task location.

iii 50.44% probability of having the least number of tasks performed in a given time span, and 49.56% probability of having the most number of tasks performed.

Figures 4a, 4b and 4c show the visual representation of Bayes theorem applied to each of the events. Subsequently, by using the prior conditional probability, the posterior conditional probability was calculated for each event using equations 2, 3 and 4. The calculated posterior probabilities ***are*** then used to obtain high probable workers for each event.

- ***Optimal Workers Selection.*** Optimal workers for a task were selected from the set of high probable workers by using the Bayesian Network model. This was accomplished by following the three-step process proposed by [30] who used the Bayesian Network for investigating effective wayfinding in airports. The steps involved constructing a conceptual model structure, defining the model state and quantifying the model. In this work, the conceptual model interprets the significant factors constituted by the nodes which influenced the selection of the right workers. The interactions between the nodes were represented by the directed arrows. Primarily, the Bayesian Network model is made up of discrete nodes. Each node is categorised into a small number of states. Considering the context of our problem, the states were chosen in a meaningful way which are of distinct values and mutually exclusive. Finally, the quantification of the nodes was performed by using information obtained from the various sources, such as the related literature, experimental data, simulation results, and statistical models.

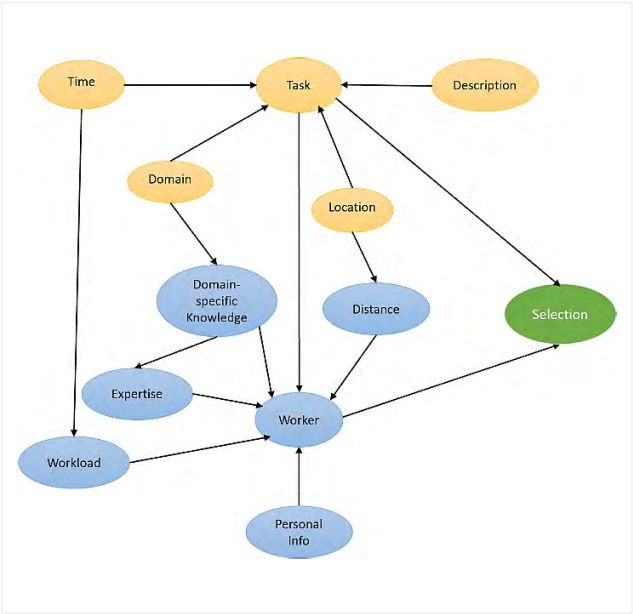

FIGURE 5. Bayesian network model for worker selection.

i. Conceptual Model Structure: The conceptual model serves as the basis of the workers' selection model. This was done by using the Bayesian Network. Our conceptual model was developed based on the efficient worker's selection mechanism for task allocation in SC. To achieve this, a thorough review of the workers' selection process for allocating tasks in the SC platform was conducted. The acquired information was then used to construct the nodes and the interactions between the nodes, as shown in Figure 5. Each of the nodes in the network represents the variables, and the incoming arcs represent the required nodes for predicting the right set of workers for the tasks. From this network, it is possible to scan different conditional independence relationships. This network also expresses the causal relationships between the variables which form a directed acyclic graph that represents the nodes and the relationships

between them. For example, the requested spatial task variable is represented by the task node. It is directly connected with the other four nodes, i.e. time, category, location, and description. The state of the task node is directly influenced by the connected nodes, which represent the conditional independence relationships among the variables. On the other hand, four primary variables comprising of task domain knowledge, expertise, distance, and workload would directly influence the decision on whether a particular worker should be selected or not. The decision also depends on the outcome of the task variable. The relationships imply that optimal worker selection is conditionally dependent on those variables. It is also worthy to note that the domain-specific knowledge is conditionally dependent on the domain of the requested task while the expertise of the worker is dependent on the categorical knowledge of the worker. This is because the worker's categorical knowledge indicates the kind of task that he/she is good at. Similarly, distance is influenced by the location of a task because a worker needs to go to the specified place from his/her current location so as to perform the required task.

ii. Defining the Model States: In this phase, the nodes obtained in (i), and the associated information gathered from the literature during the construction of the conceptual model, were used to define and assign the states. After the nodes were defined, each was given binary states in order to construct a robust model. The nodes, their descriptions, and states are listed in Table 2.

iii. Model Quantification: In order to quantify the nodes of the Bayesian Network, we used information regarding spatial workers and tasks by adopting the datasets used by previous researchers [5], [8]. The attained posterior probabilities of the high probable worker for each event (worker's expertise, distance and workload) were used to quantify the edges. Subsequently, the posterior probabilities of each event were summed up to obtain the final probability score $P(S)$ for each workers using equation 6. The set of these workers possessing final probability scores are defined as the set of optimal workers. Workers from this set would be ranked in a descending order based on their probability scores. The worker with the highest optimal score is to be considered for the requested task. If the candidate is unavailable, then the next candidate who has the second highest score would be considered for selection and so forth. In equation 6, $w1$, $w2$, and $w3$, with $w1+w2+w3 = 100$ are weights to allow trade-off among the three events.

TABLE 2. Nodes and states of the worker selection Bayesian network model.

Node Definitions	Description	States
Categorical knowledge	Past experience of a worker on particular task type	Present, Absent
Ratings	An indicator of the worker's performance on completed tasks.	High, Low
Distance	Distance between the task location to the worker's physical location.	Long, Short
Workload	Distribution of task load among registered platform workers for a specific time duration (distribution of tasks).	Max, Min
Task	Tasks requested by the requesters for assigning to suitable workers.	Present, Absent
Start-time	The time to start a task.	Exact, Inexact
End-time	The time to end a task.	Exact, Inexact
Category	Types of tasks.	Common, Uncommon
Location	Location of the task to be performed.	Informative, Uninformative
Worker	Registered spatial workers.	Available, Unavailable
Selection	The process of selecting optimal worker.	Effective, Ineffective

In worker selection, all the events do not carry the same influence in selecting the right worker. Therefore, the weighted sum-based multi-criteria evaluation scheme [31], [32] is utilized to prioritize the selection of variables (events) that are more important to the worker selection process. The model provides a systematic process for finding a solution based on many criteria. This is achieved by assigning weights to each criterion or group of criteria, based on their importance to the problem state. Weights (in percentage) are assigned to each criterion so that the total weights would add up to 100%. In order to define weights, an assumption is made by adapting and extending the heuristics defined by [14]. The extended four heuristics may be

$$P(R_i \mid Chosen) = \frac{P(Chosen \mid R)\, P(R_i)}{\mathrm{P(Chosen}|R_i)P(R_i) + \cdots + P(Chosen|R_n)P(R_n)} \tag{2}$$

$$P(D_i \mid Chosen) = 1 - \frac{P(Chosen \mid D)\, P(D_i)}{\mathrm{P(Chosen}|D_i)P(D_i) + \cdots + P(Chosen|D_n)P(D_n)} \tag{3}$$

$$P(F_i \mid Chosen) = 1 - \frac{P(Chosen \mid F)\, P(F_i)}{\mathrm{P(Chosen}|F_i)P(F_i) + \cdots + P(Chosen|F_n)P(F_n)} \tag{4}$$

$$The\ prior\ belief\ for\ a\ given\ class = \frac{number\ of\ samples\ of\ the\ particular\ class}{total\ number\ of\ samples} \tag{5}$$

TABLE 3. Example of the processed dataset for spatial task location.

Task No.	Task domain Id	Location Id	Latitude	Longitude
1	1	145064	53.3648119	-2.2723465833

useful in reducing the complexity of the problem of worker selection, as follows: i) Among workers situated within the same distance from the task location, those with higher expertise score and lesser task workload, would be given higher preference, ii) Among workers with the same expertise and task workload scores, those who are closer to the location would be given higher preference, iii) Among workers with the same expertise and task workload scores, those who have higher expertise would be given higher preference, and iv) Assumption is made that tasks possess close geographical proximity in the problem space.

$$P(S) = w_1 * P(R \mid Chosen) + w_2 * P(D \mid Chosen) + w_3 * P(F \mid Chosen) \quad (6)$$

V. EXPERIMENT

The evaluation of the proposed framework was done in two phases. In phase 1, we measured the accuracy of matching heterogeneous tasks to optimal workers. In phase 2, we evaluated the efficiency of the framework by measuring the real-time performance of the task matching. In both phases, we measured the performance of our proposed Bayesian Network-based (BN) task matching against three other approaches. The approaches are based on three baseline algorithms which are Greedy algorithm [3], [10], [21], [22], kNN, time-weighted kNN algorithm [20] and Genetic Algorithm [12]. In the current experiment, both the synthetic and real-world datasets were used to simulate a realistic SC scenario so as to demonstrate the feasibility of the proposed approach. In the following subsections, we describe our experimental datasets, settings, procedures and evaluation metrics.

A. DATASETS

Our datasets were extracted from Gowalla[1] and Yelp[2] through its public API. Gowalla is a location-based social networking website where users share their locations by checking-in. The friendship network is undirected. The datasets consist of 196,591 nodes and 950,327 edges. There was a total of 6,442,890 check-ins of users over the period from Feb. 2009 to Oct. 2010. In comparison, the Yelp datasets consist of a subset of their businesses, reviews, and user data for use in personal, educational, and academic purposes. To prepare the datasets for evaluation in this study, we processed the raw datasets that contained spatial task locations, and spatial workers, around the task location, according to the experimental requirement. A sample structure of the processed datasets that composed of the requested tasks and available workers are shown in Table 3 and Table 4, respectively. To calculate the distance of each worker from the spatial task location, we used the Haversine formula [33]. With the knowledge of workers' current longitude and latitude, the Haversine formula helps us to determine who among the workers has the shortest distance to a task's location within its sphere radius.

[1]https://snap.stanford.edu/data/loc-gowalla.html
[2]https://www.yelp.com/dataset/challenge

TABLE 4. Example of the processed dataset for spatial workers.

Worker Id	Domain-specific Knowledge	Expertise (R)	Distance (m)	Workload Balancing (F)
1	1	4.8	1695	3
3	1	4	2204	4
5	1	3.8	1650	3
8l	1	3.7	1929	1

B. EXPERIMENTAL SETTINGS

The experiment was executed on an Intel Core i5-4200U CPU @1.60 GHz with 8GB of RAM. A uniform implementation was provided for all the tested algorithms used in this experiment. Golang programming language [34] was used for implementing our model, as well as other baseline algorithms involved in the experiment. Concurrency and high-performance are the big features of Golang as a language.

C. EVALUATION METRIC

We measured the accuracy of matching heterogeneous tasks to optimal workers by using average error rate, as inspired by [14] using the quality control runtime approach, ground truth [2], [35]. Note that in the context of crowdsourcing systems, there are various quality assessment approaches that allow one to measure quality attributes (such as worker expertise may be measured through questionnaires [2], the accuracy of task allocation through evaluating task matching [14]), and one of them is by using ground truth [35]. Ground truth compares answers with a gold standard. Gold standard is a list of known/actual answers that you can use to evaluate in the ground truth. The average error rate, e, is the ratio of incorrect aggregate results to the total number of tasks in an experiment, $N_{\mathcal{T}}$, see equation 7 [14].

$$\mathrm{e} = \frac{1}{N_{\mathcal{T}}} \sum_{k=1}^{N_{\mathcal{T}}} 1[\mathrm{O}^{\mathcal{T}}\mathrm{k} \neq \mathrm{Tr}^{\mathcal{T}}\mathrm{k}] \quad (7)$$

The incorrect aggregate result is referring to the aggregation of matching results that are conflicting with the ground truth data. The average error rate decreases if matching accuracy increases, which means that more relevant tasks can be matched to the right workers.

During the experiment, each of the spatial tasks is allocated to the workers one by one. We assume that the ground truth of a task $\mathcal{T}i$ is binary. This assumption is reasonable as it is true in many real-world situations (e.g., whether the pictures are taken at a particular spot or not). Let, $\mathrm{Tr}^{\mathcal{T}i} \in \{0, 1\}$, be the ground truth of a task $\mathcal{T}i$. 1 indicates a right match between a worker to a task, otherwise 0. It explains how accurate an approach is in matching each task with an optimal set of workers while producing less number of errors. Fewer error rates indicate higher matching accuracy.

D. EVALUATION PROCEDURE

The evaluation procedures were divided into two phases involving two different sets of experiments. Experiment 1 evaluated the task matching accuracy while experiment 2 measured the runtime performance of the proposed model. Each experiment was repeated for the other three approaches for comparative analysis. Details of the evaluation procedures are further explained below.

1) MATCHING ACCURACY COMPARISON

The accuracy in the task matching is measured based on how many errors are produced when evaluating the ground truth data against the gold standard. The lesser is the average error rate, and the higher is the accuracy of the task matching approach. In this experiment, five sets of workers and tasks were considered as our datasets. Each dataset consists of fifty workers and five tasks. Each task has its specifications (i.e. task location, domain type, expiration time, and description). Similarly, each worker has unique attributes (i.e. worker personal information, domain-specific knowledge, expertise, distance, and task workload).

The first step of the experiment involved matching each of the tasks with a suitable worker from the first dataset. Each task was matched with multiple workers; hence we assume if one worker is unavailable or unwilling to take the task, the next worker would be selected, and the process continues until the task is being allocated. In doing so, the candidate workers for every task need to be ranked. The ranking of the workers was performed by taking the assumption that a worker with the highest rank would be the one who exhibits the best work performance, requires short distance to travel, and has completed only a small number of domain specific tasks within a specific period of time. Initially, the ranking of the workers was produced using manual task allocations based on human's judgement. Eventually, five sets of ranked workers were produced for five tasks for the first dataset, which serves as the gold standard data for our evaluation.

In the second step of the experiment, by using the same dataset, we executed our task matching algorithm and evaluated its accuracy aiming to match the right task to the right worker. The process outputs a set of ranked workers for each task; as a result, a total of five sets of workers was produced for five task allocations, which served as the ground truth data. We then compared the ground truth against the gold standard and used the average error rate to report the percentage of the discrepancy between them. The lower the percentage of average error rate, the higher is the accuracy in the task matching. Subsequently, the same step was repeated for the remaining four datasets. Similarly, the average error rates in the task matching of other baseline models were also measured by using the same datasets, and following the same evaluation procedures for comparative analysis.

2) TIME PERFORMANCE COMPARISONS

In this evaluation, we measured the running times of our proposed approach and other baseline approaches by using the same datasets. We reported the running times in allocating the incoming tasks to the right worker for a round of every 2 seconds, following a batch-based task assigning mode by [36]. For each timestamp, we considered a number of tasks, $n = [1,10]$, and the number of workers, $w = [1, 5000]$. For each round within a timestamp, we calculated the average time (in milliseconds) of 50 iterations, as well as the standard deviation, and then reported the results.

E. RESULT AND DISCUSSION

The results, in terms of task matching accuracy, and time performance, is discussed. Noteworthy that we provided uniform implementations, and settings for all the approaches used when comparing the results.

1) ANALYSIS OF MATCHING ACCURACY

Figure 6 shows the performance of the proposed approach, and other baseline approaches, in terms of reduction in the average error rates in the task matching comparisons. The horizontal axis represents the five datasets used in our experiment. The vertical axis represents the different error rates for each of the approach. The experimental results show that the average error rate of the proposed BN-based approach was 10.4%. While the average error rate for the task matching approaches that used the Greedy, the kNN, the Weighted-kNN and Genetic algorithms stood at 72.8%, 77.2%, and 71.6% and 67.2% respectively. These results indicated that the proposed approach is able to select the most optimal worker for every task in a heterogeneous SC tasks environment. In other words, the proposed approach provides the highest accuracy in matching the right tasks to the right workers when compared to other baseline approaches. The reasons could be attributed to the following:

a. The Greedy algorithm always computes the optimal solution with just a single attempt, from starting to ending of a data point on a sorted dataset. It never goes back to re-examine the decisions or consider other alternatives. For instance, after executing the Greedy algorithm on a sorted dataset which was based on the distance of the probable workers from the task locations, it was observed that the resulted output was still not optimal for other input variables, such as expertise rating scores and workload. Another reason is that the Greedy algorithm only takes one variable at one single instance, and finds an optimal

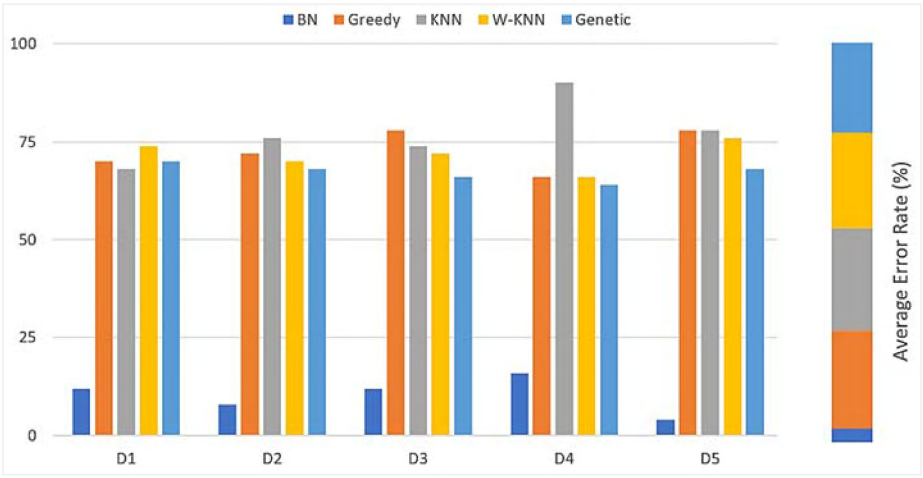

FIGURE 6. Average error rate.

solution for it. It does not consider other variables at that instance. We also found that when the Greedy algorithm was first executed on the workers' expertise, and executed it again on the outcome from the first match but this time the matching is based on the task workload scores, the resulting output was not optimised for both expertise and workload. This finding indicates that the Greedy algorithm performed better when the datasets contained only a single criterion (variable).

b. The kNN (k-nearest neighbour) algorithm uses the local neighbourhood computation, and it searches through the datasets for the k-most similar instances. It then computes the distance by using Euclidian distance rules between data points and generates the nearest neighbour list. When computing distances between data points, each attribute or variable is normally weighed in the same way. It gives the same preference to all variables of a dataset by assigning the same weights while calculating the k-nearest neighbour distance of the instance. This means that variables which were less significant would also carry the same weight on the distance when compared to more important variables. However, in order to find the set of optimal workers, other variables such as expertise, distance, and workload may carry different weights on the worker's selection. This may cause the kNN algorithm to be less accurate (accuracy 22.8%) in selecting optimal workers for every task.

c. The achieved accuracy of the W-kNN algorithm in the experiment was 28.4%. Unlike the kNN, the W-kNN algorithm assigns variables with different weights according to the impact of each variable in the worker selection process. However, the reason for achieving a low accuracy rate could be mainly due to the non-optimized weights of the variables in the selection process.

d. The genetic algorithm (GA) is a metaheuristic algorithm that is inspired by the evolutionary ideas of natural selection in the attempt to search for the optimal solution of an optimization problem [37], [38]. The search is accomplished by imitating the operation of a population evolution. It first forms a population of candidate solutions to a problem, and then new solutions are formed by ''breeding'' the best solutions from the population's members, resulting in the formation of new generations. The population then evolves through many generations, and the search stops when the best solution is obtained. Genetic algorithms are particularly useful for solving the problem of group formation in a group-based recruitment model [12], [39], such as assigning a group of workers to a cluster of tasks. However, GAs is not directly suitable for solving constraint optimization problems- where the goal is to optimize an objective function with respect to some variables in the presence of constraints on those variables [40]. For example, in a single task allocation problem, workers' selection for a requested task depends on the worker's domain-specific knowledge, expertise, travelling distance and task workload balancing. These variables may carry different weights on the worker's selection decision making, as explained in the first section. Moreover, GAs cannot effectively solve problems in which the only fitness measure is a single right or wrong measure (like decision problems) [40]. For instance, for selecting the most appropriate worker for a task from a group of optimal workers in a single task allocation problem, the process involves complex decision making based on several variables as there is no way to converge on the solution. In these cases, a random search may find a solution as quickly as a GA. The aforementioned reasons may cause low accuracy (32.8%) for GA observed in the experiment with respect to the formulated problem.

e. The proposed BN-based algorithm produces a high accuracy of 89.6% in selecting the optimal workers for every heterogeneous task of SC. The achieved higher accuracy is due to BN's ability to provide support for decision making, and can collate, organise and formalise

information from various sources [30]. It is effective, especially in a complex situation such as optimal workers selection problem, where the decision depends on various factors that influence the selection. The data related to factors may be sparse, and so each piece of available information need to be utilised. BN can combine different sources of information in a mathematically coherent manner, incorporate data with different accuracies and allow the combination of data measured on different levels of accuracy to be undertaken. This means the proposed BN model combines workers' information related to task domain-specific knowledge, expertise or performance history, distance to task location, and task workload distribution to select optimal workers for task allocation. Therefore, a worker is classified depending on whether he/she is suitable for a task, given the attributes of the worker. The classification outcomes, either high or low probable workers. Eventually, we obtain the high probable workers by excluding low probable workers through probabilistic predictions for each of the attribute. Furthermore, in BN, variables are not strongly correlated to each other, given the classification node [41]. In a complex environment as mentioned where results depend on various sources of information, may need to introduce or exclude any variable into the system for tuning system performance, it may still present accurate classification result in a large number of datasets by individually identifying and mapping each of the variables with the model classifier.

2) ANALYSIS OF RUNTIME PERFORMANCE

In experiment 2, we aim to analyse the running time of the proposed approach, for each round of the task matching, for the number of tasks, $n = [1, 10]$, and the number of workers, $w = [1, 5000]$, against other baseline approaches.

In this process, we assume each round to have two seconds of a time interval, and there is a total of six timestamps. Each timestamp has five rounds, which implies one task allocation life cycle of 60 seconds of each approach, were evaluated in this experiment. Table 5 shows the average execution time of 50 iterations (in milliseconds) for each of the rounds, along with the standard deviations. For each round of task allocation, the number of tasks and workers varied from 1 to 10, and 1 to 5000 respectively.

The result shows that the larger the number of tasks, the higher the execution time for task matching, for all approaches. For example, if the number of tasks was three, it took lesser time than when the number of tasks was 10. In both cases, the number of workers was the same, which is 1000 for timestamp 1 and 2. This could be due to the matching mechanism consumed more execution time when allocating a large number of tasks to a large number of workers at a single instance. Note that in most of the rounds, for a large number of workers, the BN performed better than other approaches, and the Genetic performed better than the kNN, W-kNN and Greedy.

Figure 7 shows the results of the runtime performance comparisons of the approaches. The results indicate a task allocation lifecycle of 60 seconds which comprised of task matching between 171 tasks and 55640 workers by the SAT server, using six timestamps. The overall observation showed that our proposed Bayesian Network-based task matching approach performed the best for all timestamps (average execution time 20.72 ms), and Greedy revealed the least performance (32.62 ms). The Genetic showed the second best performance (23.21 ms), and the KNN (25.24 ms) and W-KNN (26.75 ms) showed the third and fourth best performance respectively in most of the cases.

Our proposed approach (BN) also produced the fastest task matching predictions when applied to large datasets as compared to other approaches, due to its less complex calculations. For instance, the parameters of the proposed BN approach i.e., the apriori and conditional probabilities, were 'learnt' or determined by using a deterministic set of steps. This involved two fundamental operations, which first calculated the prior and class conditional probabilities, and then counting and dividing. There was no iteration, epoch or optimisation of a cost equation. Moreover, in our approach, there is no error back-propagation involved, thus speed up the training process.

In contrast, Genetic algorithm (GA) took comparatively more time than BN. This is due to its concept of group formation in Group-based Multi-Task Worker Selection which starts from group size equals 1, and it is incremented gradually up to the maximum group size. This leads the GA to consider various attributes of each of the workers in the population. Each individual in the population is evaluated and given a fitness score based on how well they solve the particular problem. The higher the individual's fitness score, the greater their probability of evolving. The repeated fitness function evaluation results in increasing the rate of mutation, which may cause high computational time [40].

The kNN algorithm, including the W-kNN, took a long time to calculate because of the distance calculations required for each new cases found among the instances. As the classification time was directly related to the number of data [42], it means that the bigger the dataset, the more extensive distance calculations need to be performed. This caused the classification process to become extremely slow. It is observed that W-kNN took relatively higher time than the kNN. This is due to the former's additional arithmetic operations such as multiplication of weights with each of the variables (expertise, distance and workload) which may cost higher computational times.

In the Greedy algorithm where the optimal result was required, the problem was solved in stages. In each stage, one input was considered for a given problem, and if that input was feasible, then it would be included in the solution. Therefore, by including all those feasible inputs together, an optimal solution can be found. However, for many problems, there was no guarantee that making locally optimal

TABLE 5. Comparison of algorithms in terms of execution time (milliseconds) on uniform dataset.

No of Tasks	No. of Workers	BN	kNN	W-kNN	Greedy	GA
Timestamp 1						
1	50	0.4373 ± 0.12	0.5681 ± 0.13	0.7768 ± 0.15	0.4196 ± 0.60	0.99 ± 0.31
3	2500	16.328 ± 2.5	23.228 ± 3.09	25.472 ± 3.01	34.215 ± 4.s02	36.423 + 0.34
5	500	6.253 ± 1.32	8.72 ± 1.66	9.12 ± 1.88	9.765 ± 1.99	8.34 + 1.52
10	2000	38.628 ± 4.11	41.55 ± 5.78	44.13 ± 5.34	45.9896 ± 5.89	35.40 ± 4.76
3	1000	7.268 ± 1.47	9.396 ± 1.98	9.92 ± 1.66	18.92 ± 2.34	6.66 ± 2.02
Timestamp 2						
10	500	2.011 ± 0.15	22.331 ± 3.65	23.769 ± 2.89	9.486 ± 2.34	9.85 ± 2.54
1	100	0.622 ± 0.04	1.23 ± 0.87	1.278 ± 0.99	3.835 ± 0.97	1.02 ± 0.89
9	50	1.57 ± 0.89	2.43b ± 1.34	2.52 ± 1.21	2.021 ± 0.87	1.01+1 ± 0.99
10	1000	24.628 ± 4.32	27.55 ± 4.90	28.95 ± 3.20	29.691 ± 3.56	19.70 ± 3.92
4	3000	34.544 ± 5.56	40.458 ± 6.04	45.976 ± 6.20	56.898 ± 8.28	36.64 ± 7.67
Timestamp 3						
8	4000	53.767 ± 7.88	55.671 ± 8.52	56.262 ± 9.02	66.060 ± 10.12	56.50 ± 9.92
6	4500	47.24 ± 4.32	54.906 ± 7.99	56.531 ± 9.30	62.223 ± 9.15	48.805 ± 6.87
5	3000	38.544 ± 4.22	42.458 ± 5.43	45.75 ± 5.23	48.433 ± 6.25	36.668 ± 5.62
1	1000	2.940 ± 0.97	5.459 ± 1.23	6.624 ± 1.01	15.958 ± 3.43	11.28 ± 3.87
9	400	1.769 ± 0.11	1.929 ± 0.78	1.930 ± 1.11	16.815 ± 3.89	9.13 ± 2.97
Timestamp 4						
8	2500	38.66 ± 4.89	44.042 ± 5.32	47.705 ± 5.67	42.010 ± 6.23	43.81 ± 5.79
6	1500	20.968 ± 3.85	23.78 ± 3.03	23.94 ± 3.57	34.325 ± 4.89	21.12 ± 4.67
10	2500	41.87 ± 5.40	45.65 ± 6.53	47.9 ± 6.46	50.751 ± 9.12	46.68 ± 8.78
6	200	3.92 ± 1.23	4.245 ± 0.99	5.01 ± 1.34	5.007 ± 1.45	3.42 ± 1.34
1	5000	10.686 ± 2.34	22.264 ± 3.34	21.98 ± 3.89	70.441 ± 14.29	16.46 ± 2.32
Timestamp 5						
10	5000	77.075 ± 16.2	84.5 ± 17.39	84.94 ± 17.52	86.647 ± 19.31	83 ± 12.87
8	3000	44.66 ± 6.34	47.71 ± 9.44	49.42 ± 9.99	53.118 ± 9.21	50.54 ± 8.91
1	1500	3.13 ± 1.12	6.33 ± 1.34	7.43 ± 1.78	21.690 ± 3.29	5.41 ± 0.98
5	1500	15.25 ± 3.42	21.72 ± 4.50	22.03 ± 3.22	36.738 ± 6.89	15.15 ± 3.12
3	3500	26.328 ± 4.21	33.233 ± 5.88	39.472 ± 4.56	51.851 ± 7.97	26.69 ± 6.31
Timestamp 6						
10	1200	27.57 ± 3.99	29.55 ± 3.99	30.01 ± 4.19	28.977 ± 4.38	26.42 ± 4.90
1	40	0.418 ± 0.09	0.523 ± 0.23	0.667 ± 0.05	1.02 ± 0.11	0.49 ± 0.17
3	3200	19.328 ± 2.38	33.22 ± 4.20	38.72 ± 5.29	50.826 ± 8.12	21.98 ± 5.98
9	100	2.54 ± 0.54	3.2 ± 1.12	3.39 ± 0.89	3.991 ± 0.90	2.71 ± 0.78
5	1300	12.51 ± 2.89	19.12 ± 3.19	19.92 ± 2.87	28.433 ± 3.72	13.985 ± 3.12

improvements in a locally optimal solution could provide an optimal global solution. Furthermore, the Greedy algorithm made decisions based on the information it has at any one step, without considering the overall problem space. Hence, it often failed to solve cases that involve a complex problem, such as the Knapsack problem [43], which involved deciding which subset of items to be considered, based on a set of items, for achieving optimal results. Since optimal worker selection in task matching is a complex problem due to the various task specifications and workers attributes, it led the Greedy algorithm to find optimal choice for each of the attributes, at the expense of high computational time. To sum up, our experimental findings indicate that the proposed approach was able to outperform other baseline approaches in

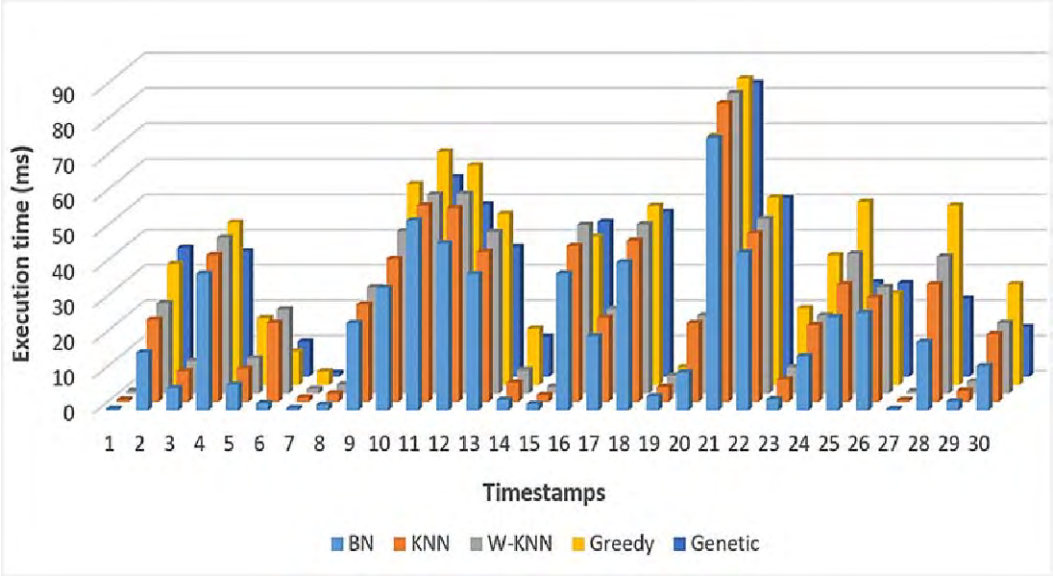

FIGURE 7. Time performance analysis of a 60 secs task allocation lifecycle.

terms of the efficiency of the task matching when allocating heterogeneous tasks to SC workers.

VI. CONCLUSION

Task matching efficiency is still an issue in a heterogeneous task allocation environment for Spatial Crowdsourcing. In this study, we proposed a framework that can efficiently select the optimal workers for every heterogeneous task in SC, based on the knowledge about tasks specifications (geographical proximity, domain types, and expiration times) and workers' attributes (workers' domain-specific knowledge, expertise, travelling distance, workload). Tasks are clustered and scheduled by using k-medoids partitioning technique. Bayesian Network is then used to select and match optimal workers for every task. To determine the efficiency of our proposed framework, we evaluate the accuracy of its task matching approach, and its runtime performance. The accuracy of the task matching process is determined by the percentage of the average error rate. Our experimental results show a significantly low average error rate in our proposed approach as compared to other approaches, which is at 10.4%. The runtime performance of our approach against other baseline approaches is measured by the average execution time. The results demonstrate that our approach has the fastest average execution time at 20.72 ms, followed by GA (23.21 ms), KNN (25.24 ms), W-KNN (26.75 ms) and Greedy (32.62 ms).

In conclusion, our proposed framework is proven feasible to efficiently match the right task to the right worker in a heterogeneous tasks specifications and diverse workers' attributes. For our future work, we would incorporate the aspects of trustworthiness into the framework, and investigate the impact of the proposed approach on both workers and tasks requesters satisfactions.

REFERENCES

[1] B. Guo, Y. Liu, L. Wang, V. O. K. Li, J. C. K. Lam, and Z. Yu, "Task allocation in spatial crowdsourcing: Current state and future directions," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1749–1764, Jun. 2018.

[2] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh, "Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–40, Apr. 2018.

[3] L. Tran, H. To, L. Fan, and C. Shahabi, "A real-time framework for task assignment in hyperlocal spatial crowdsourcing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, pp. 1–26, Feb. 2018.

[4] J. Wang, Y. Wang, D. Zhang, J. Goncalves, D. Ferreira, A. Visuri, and S. Ma, "Learning-assisted optimization in mobile crowd sensing: A survey," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 15–22, Jan. 2019.

[5] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Trans. Spatial Algorithms Syst.*, vol. 1, no. 1, pp. 1–28, Aug. 2015.

[6] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Comput.*, vol. 16, no. 5, pp. 36–44, Sep. 2012.

[7] H. Kong, "Spatial Crowdsourcing: Challenges and Opportunities," *IEEE Data Eng. Bull.*, vol. 39, no. 4, pp. 14–25, 2016.

[8] U. ul Hassan and E. Curry, "Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning," *Expert Syst. Appl.*, vol. 58, pp. 36–56, Oct. 2016.

[9] Y. Tong and Z. Zhou, "Dynamic task assignment in spatial crowdsourcing," *SIGSPATIAL Special*, vol. 10, no. 2, pp. 18–25, Nov. 2018.

[10] P. Yang, N. Zhang, S. Zhang, K. Yang, L. Yu, and X. Shen, "Identifying the most valuable workers in fog-assisted spatial crowdsourcing," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1193–1203, Oct. 2017.

[11] Y. Jiang, L. Cui, Y. Cao, L. Liu, W. He, L. Pan, Y. Zheng, and Q. Li, "Spatial crowdsourcing task assignment based on the quality of workers," in *Proc. 3rd Int. Conf. Crowd Sci. Eng. (ICCSE)*, 2018, pp. 1–6.

[12] M. Abououf, R. Mizouni, S. Singh, H. Otrok, and A. Ouali, "Multi-worker multi-task selection framework in mobile crowd sourcing," *J. Netw. Comput. Appl.*, vol. 130, pp. 52–62, Mar. 2019.

[13] Y. Zheng, G. Li, and R. Cheng, "DOCS: A domain-aware crowdsourcing system using knowledge bases," *Proc. VLDB Endowment*, vol. 10, no. 4, pp. 361–372, Nov. 2016.

[14] C. Miao, H. Yu, Z. Shen, and C. Leung, "Balancing quality and budget considerations in mobile crowdsourcing," *Decis. Support Syst.*, vol. 90, pp. 56–64, Oct. 2016.

[15] S. R. B. Gummidi, X. Xie, and T. B. Pedersen, "A survey of spatial crowdsourcing," *ACM Trans. Database Syst.*, vol. 44, no. 2, pp. 1–46, Apr. 2019.

[16] W. Sun, Y. Zhu, L. M. Ni, and B. Li, "Crowdsourcing sensing workloads of heterogeneous tasks: A distributed fairness-aware approach," in *Proc. 44th Int. Conf. Parallel Process.*, Sep. 2015, pp. 580–589.

[17] A. Abdollahzadeh, A. Reynolds, M. Christie, D. W. Corne, B. J. Davies, and G. J. J. Williams, "Bayesian optimization algorithm applied to uncertainty quantification," *SPE J.*, vol. 17, no. 3, pp. 865–873, Sep. 2012.

[18] S. Sankararaman, Y. Ling, and S. Mahadevan, "Uncertainty quantification and model validation of fatigue crack growth prediction," *Eng. Fract. Mech.*, vol. 78, no. 7, pp. 1487–1504, May 2011.

[19] M. E. Borsuk, C. A. Stow, and K. H. Reckhow, "A Bayesian network of eutrophication models for synthesis, prediction, and uncertainty analysis," *Ecol. Model.*, vol. 173, nos. 2–3, pp. 219–239, Apr. 2004.

[20] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. C. Cao, Y. Tong, and C. J. Zhang, "GMission: A general spatial crowdsourcing platform," *Proc. VLDB Endowment*, vol. 7, no. 13, pp. 1629–1632, Aug. 2014.

[21] P. Wu, E. W. T. Ngai, and Y. Wu, "Toward a real-time and budget-aware task package allocation in spatial crowdsourcing," *Decis. Support Syst.*, vol. 110, pp. 107–117, Jun. 2018.

[22] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, and J. Zhao, "Reliable diversity-based spatial crowdsourcing by moving workers," *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1022–1033, Jun. 2015.

[23] S. Basu Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Task assignment optimization in knowledge-intensive crowdsourcing," *VLDB J.*, vol. 24, no. 4, pp. 467–491, 2015.

[24] T. Hu, M. Xiao, C. Hu, G. Gao, and B. Wang, "A QoS-sensitive task assignment algorithm for mobile crowdsensing," *Pervas. Mobile Comput.*, vol. 41, pp. 333–342, Oct. 2017.

[25] M. Abououf, S. Singh, H. Otrok, R. Mizouni, and A. Ouali, "Gale-shapley matching game selection—A framework for user satisfaction," *IEEE Access*, vol. 7, pp. 3694–3703, 2019.

[26] J. Jiang, B. An, Y. Jiang, C. Zhang, Z. Bu, and J. Cao, "Group-oriented task allocation for crowdsourcing in social networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 28, 2019, doi: 10.1109/TSMC.2019.2933327.

[27] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *Proc. IEEE 12th Int. Conf. Mobile Ad Hoc Sensor Syst.*, Oct. 2015, pp. 136–144.

[28] D. Schiek and A. Gideon, "Outsmarting the gig-economy through collective bargaining—EU competition law as a barrier to smart cities?" *Int. Rev. Law, Comput. Technol.*, vol. 32, nos. 2–3, pp. 275–294, Sep. 2018.

[29] E. Jacquier and N. Polson, *Bayesian Methods In Finance*. London, U.K.: Oxford Univ. Press, 2011.

[30] A. C. Farr, T. Kleinschmidt, S. Johnson, P. K. D. V. Yarlagadda, and K. Mengersen, "Investigating effective wayfinding in airports: A Bayesian network approach," *Transport*, vol. 29, no. 1, pp. 90–99, Mar. 2014.

[31] F. Pichon, C. Labreuche, B. Duqueroie, and T. Delavallade, "Multidimensional approach to reliability evaluation of information sources," in *Information Evaluation*, vol. 9781848216, Hoboken, NJ, USA: Wiley, 2014, pp. 129–159.

[32] L. Zhao, T. Hua, C.-T. Lu, and I.-R. Chen, "A topic-focused trust model for Twitter," *Comput. Commun.*, vol. 76, pp. 1–11, Feb. 2016.

[33] C. N. Alam, K. Manaf, A. R. Atmadja, and D. K. Aurum, "Implementation of haversine formula for counting event visitor in the radius based on Android application," in *Proc. 4th Int. Conf. Cyber IT Service Manage. (CITSM)*, vol. 2016, 2016.

[34] A. A. A. Donovan and B. W. Kernighan, *The Go Programming Language*. Reading, MA, USA: Addison-Wesley, 2016

[35] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality control in crowdsourcing systems: Issues and directions," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 76–81, Mar. 2013.

[36] P. Cheng, X. Jian, and L. Chen, "An experimental evaluation of task assignment in spatial crowdsourcing," *Proc. VLDB Endowment*, vol. 11, no. 11, pp. 1428–1440, Jul. 2018.

[37] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Berlin, Germany: Springer, 2008.

[38] D. Heiss-Czedik, "An Introduction to genetic algorithms.," *Artif. Life*, vol. 3, no. 1, pp. 63–65, Jan. 1997.

[39] R. Azzam, R. Mizouni, H. Otrok, A. Ouali, and S. Singh, "GRS: A group-based recruitment system for mobile crowd sensing," *J. Netw. Comput. Appl.*, vol. 72, pp. 38–50, Sep. 2016.

[40] M. Mittal, "Comparison between BBO and genetic algorithm," *Int. J. Sci. Eng. Technol. Res.*, vol. 2, no. 2, pp. 284–293, 2013.

[41] F. S. Costa, M. M. D. S. Pires, and S. M. Nassar, "Analysis of Bayesian classifier accuracy," *J. Comput. Sci.*, vol. 9, no. 11, pp. 1487–1495, Jan. 2013.

[42] A. Ashari, I. Paryudi, and A. Min, "Performance comparison between Naïve Bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 11, pp. 33–39, 2013.

[43] J. J. Bartholdi, "The knapsack problem," in *Building Intuition* (International Series in Operations Research & Management Science), vol. 115, D. Chhajed and T. J. Lowe, Eds. Boston, MA, USA: Springer, 2008, pp. 19–31.

NOR ANIZA ABDULLAH (Member, IEEE) received the bachelor's degree (Hons.) in computer science from the University of Malaya, the master's degree in interactive multimedia from Westminster University, London, and the Ph.D. degree in computer science from Southampton University, U.K. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. She has authored or coauthored over 50 refereed publications in international journals, book chapters, and conferences. She has supervised several Ph.D. and master's students at the University of Malaya, Malaysia. She also co-supervised several Master by Research students at the Moratuwa University of Sri Lanka. Her research interests include personalized and adaptive learning, recommender system, decision support systems, big data analytics, and content-based image/video retrieval. She serves as a Reviewer for several ISI-indexed journals.

MOHAMMAD MUSTANEER RAHMAN received the B.Sc. degree in computer science and engineering from Khulna University, Bangladesh, and the master's degree in computer science from the University of Malaya, Malaysia. He is currently pursuing the Ph.D. degree in information technology with the University of Tasmania, Australia. He has eight years of industry experience in research and development. He has authored or coauthored in several publications in international journals and conferences. His research interests include spatial crowdsourcing, e-learning, affective tutoring systems, recommender systems, adaptive learning, educational technology, human–computer interaction, decision support systems, machine learning, and big data.

MD. MUJIBUR RAHMAN received the bachelor's degree (Engg.) in computer science and engineering from Darul Ihsan University, Dhaka, Bangladesh. He is currently pursuing the master's degree in computer science and information technology with University of Malaya, Kuala Lumpur, Malaysia. He has 14 years of experience in software engineering and web development. His research interests include recommender systems, crowdsourcing, spatial crowdsourcing, machine learning, cloud computing, social networking, and information retrieval.

KHAIRIL IMRAN GHAUTH received the M.IT. degree from The University of Melbourne and the Ph.D. degree from the University of Malaya. He is currently a Senior Lecturer with the Faculty of Computing and Informatics, Multimedia University. He has authored or coauthored over 20 refereed publications in international journals, book chapters, and conferences in the area of information retrieval, recommender systems, and big data.