IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 22, 2019, accepted November 8, 2019, date of publication November 12, 2019, date of current version December 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953045

An Efficient Preference-Based Sensor Selection Method in Internet of Things

ZENGWEI ZHENG [1], YANYUN TAO [1,2], YUANYI CHEN [1,3], FENGLE ZHU [1], AND DAN CHEN [1]
[1] School of Computer and Computing Science, Zhejiang University City College, Hangzhou 310015, China
[2] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
[3] Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697, USA

Corresponding author: Yuanyi Chen (chenyuanyi@zucc.edu.cn)

This work was supported in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant 61802343, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LGF19F020019 and Grant LGN20F020003, and in part by the Hang Zhou Key Laboratory for IoT Technology and Application.

ABSTRACT While it is well understood that the Internet of things (IoT) can facilitate numerous applications (e.g., environmental supervision, forest fire prevention and Intelligent farming), it also brings a significant challenge for efficiently selecting sensors that meet users' preference and specific requirement from millions of heterogeneous sensors. In this paper, we propose an improved fast nondominated sorting algorithm for efficiently preference-based sensor selection in IoT. Specifically, this proposed method mainly includes three parts: 1) Offline constructing R-tree to search sensor resources and narrowing the size of dataset according to user's preference; 2) Using an improved fast nondominated sorting approach to get nondominated front; 3) Employing TOPSIS to characterize every sensor option of the nondominated front. In order to illustrate the usability of the model, we conduct experiments on several simulation datasets. Experimental results show that this method outperforms several baselines in terms of both response time and accuracy.

INDEX TERMS Multi-criteria decision, R-tree structure, sensor selection, Internet of Things, the fast nondominated sorting algorithm.

I. INTRODUCTION

In the Internet of things(IoT), the Internet is used as medium to control all kinds of physical devices for collecting real-time environmental data and transferring the data to middleware's specific function modules. Finally the middleware returns the results to users. At present, the Internet of things is widely employed in intelligent home, intelligent city, intelligent healthcare, intelligent transportation and other fields. For example, apps on smartphones enable many sensors to monitor traffic jams and optimize drives in intelligent transportation. Type-B ultrasonic can use ultrasonic sensors, a type of acoustic sensor, to know every move and development status of the fetal, which can avoid some unexpected circumstances. Similar sensor applications are numerous in daily use.

Due to the development of communication technology and the emergence of various intelligent objects, a large number of physical objects are connected to the Internet of things. Cisco predicts that the number of objects in the Internet of things will increase to 50 billion by 2020 [1]. Therefore, a large amounts of sensor data are expected to be generated continuously. And generated date contains two kinds of data, one to describe sensors and the other capture from realistic environment. From this perspective, the biggest challenge for services selection in Internet of things is to discover sensor resources in large-scale heterogeneous environments and choose the best option according to users' specific needs and constraints.

In recent years, many scholars have made great progress in balancing users' multiple requirements of sensor by using multi-criteria decision analysis algorithms (MCDA). ViSIoT [2], CASSARAM [3], CASSF [4] and other Internet of things frameworks search for resources based on multi-criteria decision analysis algorithm with users' preference of several specific sensor's indicators in the Internet of things. The process of sensor resources selection can be divided into two phases: 1) Using static query to get a set of available sensor resources mainly according to user's preliminary

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Anwar Hossain.

requirements of sensors, such as location, sensor types and etc. Users can obtain the candidate sensor resources by querying database or other way. 2) Applying multi-criteria decision analysis algorithm (MCDA) to characterize and sort candidate sensor resources with user's preferences of sensor attributes, and returning results to user.

In multi-criteria decision analysis algorithms, the Technique for the Order of Prioritization by Similarity to Ideal Solution (TOPSIS) is widely used. TOPSIS has low time complexity, and it can provide optimal option with a set of preliminary constraints. But TOPSIS cannot take all the constraints given by users into account at the same time. According to the research results of Nunes *et al.* [21], only a small number of sensors recommended by TOPSIS are in nondominated front. The time complexity and space complexity of the fast nondominated sorting algorithm based on Pareto optimal principle are very lofty, but it has high accuracy. In order to absorb the advantages of these two algorithms, we propose an improved fast nondominated sorting algorithm. The contributions of our research are four-fold as follow:

- We construct R-tree to retrieve sensor resources statically. The R-tree is constructed to retrieve sensor resources in two-dimension space including sensor location and type, then user can obtain sensor dataset.
- We narrow the size of sensor dataset according to user preferences of sensor attributes. The dataset is copied into n copies, where n is the number of attributes. N copies are respectively related to n sensor attributes and the sensor resources in every copy are sorted according to specific sensor attribute. The final size of dataset is determined by the experiment result. Then the number of sensor option obtained from respective copies is determined with the proportion of the preference value of this sensor attribute to the total preference value.
- We propose an improved fast nondominated sorting algorithm to get the nondominated front of dataset, which reduces the time complexity and space complexity than original algorithm.
- We illustrate this model in several simulation datasets, which are generated according to the change range of sensor attributes value. And the experimental results show that the model is superior to the existing sensor selection models.

The rest of the paper is organized as follows: Section II investigates the related work of sensor selection under the Internet of things. The proposed sensor selection model is introduced in Section III. Section IV reports and discusses the experimental results. Finally, we present our conclusions and future work in Section V.

II. RELATED WORK

In this section, we first discuss the relevant work of sensor selection in the Internet of things, and then summarize the differences between our proposed model and previous work. Zhou [5] proposed that existing sensor selection methods can be divided into two categories: context-based selection methods (e.g., searching based on location and sensor type) and content-based selection methods (e.g., searching sensors that can generate specific values), which are described in the following two aspects:

Context-based sensor selection method. Context-based sensor selection methods rely on the different types of contextual information available in the sensor description. Nunes had set up a IoT visualization platform ViSIoT [2], wher e OpenWeatherMap1 as cloud sensor repository for storing the state and other relevant information of the sensor, such as their coordinates, battery level and price information. GSN as a data processing system and this model use TOPSIS for characterizing and sorting the sensor resources. Its main purpose is to help the technical and non-technical users find and use sensors by this platform. Perera presented a IoT sensor selection model CASSARAM [3] based on context awareness, which considers the sensor attributes, such as reliability, accuracy, battery life and etc. Then it employed the Weighted Euclidean distance based on Comparative Priority Index algorithm(CPWI) to characterize and sort sensors with the user's preference values of sensors attribute and user's ideals values of sensor. Neha proposed the VSBR [7], which characterizes sensors resources by calculating the weighted index (PBWI). The above two models require the user to input the specific sensor preference values. Gong proposed CASSF framework [4], where RDF graph was used to store sensor information instead of relational database for ensuring semantic information furthest, and TASI algorithm [9], [10] was created to improve the overall search efficiency. Ocean model [8] presented by Carlson, which introduced the Ambient Ocean search engine and this engine retrieves information based on arbitrary contextual data, such as location, sensor information and etc. Nithya *et al.* [29] proposed a clustering method based on cluster and cluster head information to optimize sensor selection in the Internet of things, which realized fast and effective retrieval. Ramachandran *et al.* [30] proposed a method based on cluster, where sensors were grouped into clusters based on their energy level and proximity. Then user queries were processed using natural language. Bharti proposed VoI based Sensor Ranking Mechanism VoISRAM [31], which considers its application context and balances the use of specific QoS requirements and energy consumption to get the best option. Inspired by ant clustering algorithm, Ebrahimi *et al.* [33] proposed an effective context-aware method to cluster sensors with similar context information in sensor semantic coverage network. This model has obvious scalability and faster running speed in cluster sensors and sensor search. Deshpande *et al.* [34] implemented an ant based clustering algorithm and made novel modifications to this algorithm. The algorithm samples the whole sensor space and outputs the corresponding clusters, where users can choose the best clusters according to their own needs. Given the limitations of memory, power and latency in the Internet of things, Sharma *et al.* [35] deployed unmanned aerial vehicles (UAVS) as a dynamic node and used a radial basis function kernel support vector machine to solve and optimize customer queries. Nithya *et al.* [29] proposed an approach for optimal selection of IoT sensors based on cluster and cluster head formation for faster and effective retrieval. In order to prolong the lifetime of network, Wang *et al.* [36] proposed a centralized clustering algorithm based on spectral partitioning and a distributed implementation of the clustering method based on fuzzy C-means in cluster head selection.

Nunes *et al.* [11] think that TOPSIS as common multi-criteria decision analysis algorithms has low time complexity but bad accuracy. However, the fast nondominated sorting algorithm can provide the best tradeoff solution in the context of multiple criterions but high time complexity and space complexity, which makes it hard to execute in a large number of sensor resources. The ES model integrates the advantages of TOPSIS and the fast nondominated sorting algorithm. Kertiou proposed dynamic skyline algorithm [12] to improve the accuracy of multi-criteria decision analysis algorithm, which has been widely used in multi-criteria decision applications such as web service combination [13] and wireless AD hoc network routing. Compared with the fast nondominated algorithm mentioned above, the dynamic skyline algorithm has a lower time complexity, but it isn't safe to require users to input ideal values of sensor properties.

Content-based sensor selection method. Content-based sensor selection methods use sensor output to search specific sensor sets that match user queries [14]. Truong *et al.* [6] borrowed the idea ''search-by-example approach'' from Google image retrieval, which retrieved the relevant content with given the relevant image for avoiding the situation that user input inaccurate keywords. Taking the historical output of a sensor as a comparison, the fuzzy set was used to efficiently calculate the similarity score of the sensor, and the ranked list is recommended to the user. Ostermaier *et al.* [15] proposed a real-time search engine Dyser, which supported the use of existing Web infrastructure to publish sensors dataset and retrieve the specified sensor. Mietz *et al.* [16] proposed Bayesian network model with using the highly correlated output characteristics of many sensors to learn relevant structures from the historical data of sensors, which could get the requesting probability of sensor without knowing the current sensor output then it recommended sensor with high probability to the user. Truong *et al.* [17] proposed a lightweight prediction model based on fuzzy logic algorithm to estimate the probability of sensor option and realize content-based sensor search in the Internet of things with low communication overhead and computational efficiency. Puning *et al.* [18] proposed a sensor state prediction method to estimate the short-term state of the sensor, which could make better use of the time correlation between sensor data and accurately perceive the change trend of sensor readings in the future. Vasilev *et al.* [19] proposed a scalable model based on hypergraph representation to evaluate the cooperative relationship between sensor nodes and achieve reliable node, which well covered the dynamic characteristics of complex sensor networks. In order to improve the efficiency of content-based sensor search, Puning proposed a content-based sensor search model CSME [20], which improved the efficiency and applicability of existing sensor search systems. Zhang *et al.* [32] established a multidimensional feature selection model and a dynamic sensor data prediction model by using the complex time series constituted by various sensors, which improved the accuracy and stability of long-term prediction results of iot sensor data. Zhang *et al.* [18] described an efficient MSE structure for content-based sensor search, where the short-term state of sensors are estimated in order to find candidate ones.

Ihm proposed Grid-PPPS model [22], and this method gridded the data space using hyperplane projection, which reduced the time complexity compared with the plane-project-parallel-skyline [23] model. Bijarbooneh *et al.* [24] solved the problem of sensor optimal selection in the Internet of things by combining constraint programming (CP) and heuristic greedy algorithm. This model adopted a new data acquisition scheme ASBP, which used highly correlated spatio-temporal data in the network. Zhang et al. proposed LHPM [25], which retained the reference value of sensor output during query and realized high-precision prediction, so as to reduce communication overhead as well as the cost of storage and energy. Jiang *et al.* [26] believes that the performance of support vector machines depends on setting appropriate parameters. Therefore, cuckoo search (CS) algorithm was adopted to optimize the key parameters with avoiding the local minimum problem, which existed in traditional parameter optimization methods.

Our method belongs to the first category. Compared with the content-based sensor selection algorithm, the content-based sensor selection algorithm focuses more on users' non-functional requirements of sensors. This input of model has low requirements on users' professionalism, and it's easier to carry out services. We propose an improved fast nondominated sorting algorithm, which gives a good balance between the low time complexity of TOPSIS algorithm and the high accuracy of the fast nondominated sorting algorithm based on Pareto optimal principle, so as to provide users with efficient sensor selections.

III. SENSOR RECOMMEND MODEL

In this section, we first introduce the data structures and notations used in the model, then introduce the background knowledge, and finally state our solution.

A. PROBLEM DEFINITION

For ease of the following presentation, we define the key data structures and notations used in the proposed method. Table 1 lists the relevant notations used in this paper.

Definition 1 (Sensor Attribute): In order to describe the performance of the sensor, we use six sensor attributes $F = \{f_1, \ldots, f_6\}$ including life, sensitivity, measurement accuracy,

TABLE 1. Notations used in the paper.

SYMBOL	DESCRIPTION
F	The set of sensor attributes
f_i	The i^{th} sensor attribute
p_i	The user preference value of the i^{th} sensor attribute
D_i	The dataset of the i^{th} Fastlist's copies
m_{ij}	The j^{th} sensor attribute value of i^{th} sensor attribute
$Max(f_i)$	Max value of the i^{th} sensor attribute
$Min(f_i)$	Min value of the i^{th} sensor attribute
G	The objective function
$< S_l, S_t >$	Tuple structure for R tree retrieval
S_l	The position of the sensor
S_t	The sensor type
$Fastlist$	The dataset of sensors
$Fastlist'$	The dataset of sensors after reducing
$DFront$	The first nondominated front
n	The size of $Fastlist'$

response time, start time and energy consumption in this paper.

Definition 2 (The Objective Function): Sensor selection frames as solving multi-objective optimization. For the six sensor attributes described in Definition 1, the objective functions are defined as $G = \{Max(f_1), Max(f_2), Max(f_3), Min(f_4), Min(f_5), Min(f_6)\}$. In order to simplify the solution process, we take the negative values of three sensor attributes of response time, start time and energy consumption. So the objective function is $G = \{Max(f_1), Max(f_2), Max(f_3), Max(f_4), Max(f_5), Max(f_6)\}$.

Definition 3 (The Retrieving Structures Used in R-Tree): R-tree is used for spatial data storage, which can be used to store spatial information on maps, such as restaurant addresses. In this paper, the R-tree is built by constructing a two-dimension tuple $<S_l, S_t>$, where S_l represents the position of sensor and S_t represents the type of sensor. Inputing the sensor tuple $<S_l, S_t>$, we can quickly obtain the required sensor dataset.

Definition 4 (The Dataset of Sensor): After retrieving R-tree to obtain sensor dataset, we change sensor dataset into an array Fastlist $= [[m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}], \ldots, [m_{n1}, m_{n2}, m_{n3}, m_{n4}, m_{n5}, m_{n6}]]$ according to the sequence of sensor attributes.

Definition 5 (Nondominated Front): In multi-criteria decision analysis problems, there may be conflicts and incomparability between multiple criterions. One solution may be the best in one criterion, and the worse in another. We select two sensor options (S1 and S2) randomly. When every sensor attribute value in S1 is no less than the corresponding sensor attribute value of S2, we call S1 dominant S2. In other words, if the sensor option S1 is not dominated by other sensor data, S1 is called the nondominated solution. The set of all nondominated solutions is the nondominated front.

B. BACKGROUND

1) TOPSIS ALGORITHM

Supposing we have sensor options set L $= \{l_1, l_2 \ldots, l_m\}$ and decision criterion set C $= \{c_1, c_2 \ldots, c_n\}$, then the value a_{ij} represents that the attribute value of l_i is c_j(i = 1,2…m; j = 1,2…, n). The constructed decision matrix A $= (a_{ij})_{m\tilde{A}-n}$ is used to describe the multi-criteria decision problem, and the specific formula is as follows:

$$A = \begin{array}{c c} & \begin{array}{ccccc} c_1 & c_2 & c_3 & \cdots & c_n \end{array} \\ \begin{array}{c} l_1 \\ l_2 \\ \vdots \\ l_m \end{array} & \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{pmatrix} \end{array}$$

The sensor selection frames as a multi-criteria decision analysis problem. Among the multi-criteria decision analysis algorithms, TOPSIS algorithm has a strong explanatory sorting principle. The TOPSIS aims to find the best solutions, where the best option is nearest to the optimal solution and farthest to the inferior solution [27]. The time complexity of TOPSIS is O(n^2), where n respects the size of sensor dataset. TOPSIS algorithm is summarized as follows:

1. Normalize the matrix A to A' according to the following Equation 1.

$$a'_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{z}(a_{ij})^2}} \tag{1}$$

where a_{ij} represents the value of the j^{th} attribute of the i^{th} sensor, and z is the number of rows in matrix A;

2. Each sensor attribute's minimum values and maximum values in matrix A' are calculated to get the optimal point and the worst point, which is represented by a_{+j} and a_{-j} respectively, as shown in Equation 2.

$$a_{+j} = \max_{1<x<l} a'_{ij}, a_{-j} = \min_{1<x<l} a'_{ij} \tag{2}$$

3. Euclidean distance of every sensor option to a_{+j} and a_{-j} are computed and represented by s_{i+} and s_{i-}, as shown in Equation 3,4.

$$s_{i+} = \sqrt{\sum_{j=1}^{l}(a'_{ij} - a_{+j})^2} \tag{3}$$

$$s_{i-} = \sqrt{\sum_{j=1}^{l}(a'_{ij} - a_{-j})^2} \tag{4}$$

4. Set a measurement index to characterize and sort each sensor option according to the computed value. The index formula is as shown in Equation 5.

$$c_{i+} = \frac{s_{i-}}{s_{i+} + s_{i-}} \tag{5}$$

2) THE FAST NONDOMINATED SORTING ALGORITHM

The fast nondominated sorting algorithm [27] is used to sort the options in the nondominated front according to the pareto optimal principle. The fast nondominated sorting algorithm's time complexity is $O((mn)^2)$ and storage space are $O(n^2)$, where n is the size of sensor dataset and m is the number of criterions. Given a sensor dataset P, the fast nondominated sorting algorithms need to compute two parameters of each sensor i in P including np and S_p, in which np refers to the number of option in P that dominate sensor i, and S_p refers to the option set dominated by sensor i, the algorithm is summarized as follows:

1. For every option of P, find the sensor option with np = 0 and save them in the set F_1. The set F_1 represents the first nondominated front.

2. For every option j in S_p which is dominated by sensor in F_1, update the corresponding value np to np = np-1.If np = 0, we store j in set F_2.

3. Take F_2 as the current sensor set, repeat step 2 and replace $F_{n-1}$and F_n with F_1 and F_2 respectively, until the whole sensor set is layered.

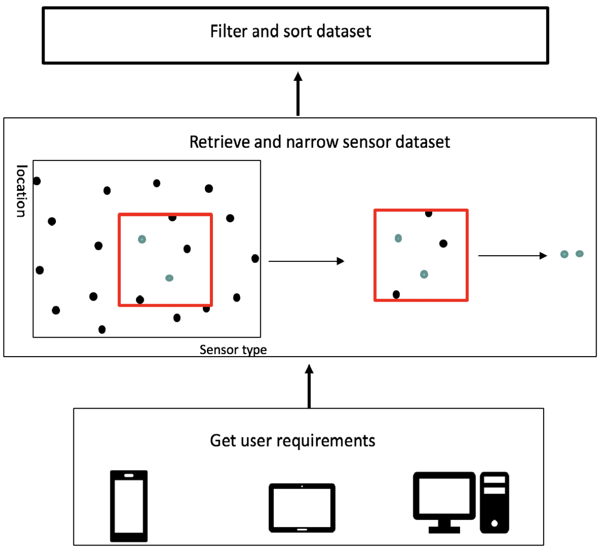


FIGURE 1. The flow chart of the model .

C. SENSOR SELECTION MODEL

The time complexity of TOPSIS is low, and it can efficiently characterize and sort the sensor dataset, but the given selection has a low proportion in the nondominated front. In order to solve this problem, scholars combined the fast nondominated sorting algorithm to propose the ES model [11]. The ES model reorders the selections given by TOPSIS and recommends them to users, which increases the proportion of the selections in the nondominated front but also increases the processing time by tens of times. Our model only obtains the first nondominated front to reduce the time complexity to O(n). The flow chart of the model is shown in figure 1:

Step 1: Get the sensor dataset. The Internet of things includes a large number of sensors. Using the traditional method to static retrieve sensor dataset is time-consuming. We use R-tree to reduce the time of repeated retrieval and improve the retrieval efficiency. According to the two-dimensional tuple $<S_l, S_t>$, the sensor dataset is retrieved and obtained. According to the order of the six sensor attributes, we change obtained dataset into array Fastlist = $[[m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}], \ldots, [m_{k1}, m_{k2}, m_{k3}, m_{k4}, m_{k5}, m_{k6}]]$, where $m_{k1}, m_{k2}, m_{k3}, m_{k4}, m_{k5}, m_{k6}$ respectively represents sensor option's six sensor attributes value.

The size of Firstlist obtained by retrieving the R-tree based on the two-dimensional tuples of sensor is still large, so it is necessary to narrow the scope of sensor dataset while it contains all nondominated front as much as possible. First, we copy the sensor array Firstlist into q copies, where q is the number of sensor attributes. In this paper, we select six sensor attributes including life, sensitivity, measurement accuracy, response time, start time and energy consumption, so the q equals to 6. We reorder the sensor resources in each copy according to specific sensor attributes, so we get six sets $[list1 \ldots list6]$. Next, the optimal size of sensor dataset n should be set according to the experiment results. When n equals to 900, the accuracy of the model is the highest, the results are shown in figure 3. Then, according to the proportion of this sensor attribute preference value in the sum of the six sensor attribute preference values, the required number of sensor options of the each copy in n is determined, as shown in Equation 6. Finally, the sensor dataset with the number of n is integrated into $Fastlist' = [[m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}], \ldots, [m_{n1}, m_{n2}, m_{n3}, m_{n4}, m_{n5}, m_{n6}]]$, where n equals to 900;

$$D_i = \frac{p_j}{\sum_{j=1}^{6} p_j} \times n \tag{6}$$

where D_i represents the dataset of the i^{th} cope after reducing, p_j represents the users' preference value of the j^{th} sensor attribute value;

Step 2: An improved fast nondominated sorting algorithm. The fast nondominated sorting approach is used to sort the options in fronts regarding their nondominated levels. However, this algorithm is very tedious and the time complexity is $O(mn^2)$, where m represents the number of criteria and n represents the number of available sensor options. The scale of the Internet of things is so large that using the fast nondominated sort algorithm to make recommendations among millions sensor options is time-consuming. Moreover, it is not necessary to sort all the sensor options according to the nondominated level in a sensor recommendation system, and it just needs to find the optimal solution set. Therefore, we improved the fast nondominated sorting algorithm to obtain the sensor options with the highest nondominated sorting in the shortest time. This algorithm was improved by combining the ideas of quick sorting algorithm, and the time complexity was reduced to $O(n)$, where n equals to 900, as shown in Algorithm 1.

After getting sensor dataset *Fastlist'* by step 1, we compute the nondominated front by the proposed model as shown in Algorithm 1. First, as shown in Line 1 ∼ 7, we define a function to obtain the dominance relation between sensor option SensorOption1 and sensor option SensorOption2. Next, as depicted in Line 10 ∼ 12, we assign values to the parameters used in the model. In Line 13 ∼ 23, We compare

Algorithm 1 An Improved Fast Nondominated Sorting Algorithm

```
input: Dataset:Fastlist' = [[m11, m12, m13, m14, m15, ..., m1k], ..., [mnk, mnk, mnk, mnk, mnk, ..., mnk]].
output: The first nondominated front: DFront
1: function MYCOMPARE(SensorOption1, SensorOption2)
2:     if SensorOption1[i] >= SensorOption2[i], i ∈ [1,k] then
3:         return TRUE
4:     else
5:         return FALSE
6:     end if
7: end function
8: function QUICKDOMINATE(Fastlist', start, end)
9:     if start < end then
10:        i ← start
11:        j ← end
12:        base ← Fastlist'[i]
13:        while i < j do
14:            inter ← mycompare(Fastlist'[i], base)
15:            if base dominate Fastlist'[i] then
16:                Fastlist'[i] ← −1
17:                i ← i + 1
18:            else
19:                Fastlist'[i] ← −1
20:                base ← Fastlist'[i]
21:                break
22:            end if
23:        end while
24:        while i < j do
25:            inter ← mycompare(Fastlist'[j], base)
26:            if base dominate Fastlist'[i] then
27:                Fastlist'[j] ← −1
28:                j ← j − 1
29:            else
30:                j ← j − 1
31:            end if
32:        end while
33:    end if
34:    DFront=QuickDominate(Fastlist', i, len(Fastlist') − 1)
35:    return DFront
36: end function
```

base node with $Fastlist'[i]$ according to pareto's optimal solution principle, If the base node dominates $Fastlist'[j]$, then $Fastlist'[i]$ is assigned to -1 and i is increased by 1. If the $Fastlist'[i]$ dominates the node base, $Fastlist'[i]$ is assigned to base node. Then, as shown in Line 24 $\sim$ 32,We compare base node with $Fastlist'[j]$according to pareto's optimal solution principle, If the base node dominates $Fastlist'[j]$, then $Fastlist'[j]$ is assigned to -1 and j is decreased by 1. If the $Fastlist'[j]$ dominates the node base, j is decreased by 1. Finally, we redefine the input value of this model as shown in Line 34.

Step 3: Sort using the TOPSIS algorithm. Referencing to user's preferences and the physical significance of each sensor attribute, TOPSIS algorithm is used to characterize each sensor resource in the first nondominated front, rank and recommend it to users.

IV. EXPERIMENT EVALUATION

In this section, we report on the results of a series of experiments conducted to evaluate the performance of the proposed sensor recommend model. We first describe the settings of experiments including datasets, comparative methods, evaluation methods and user interface. Then we report and discuss the experimental results.

TABLE 2. The value range of sensor attributes.

Sensor attribute	Unit	Value range
Life	y	5 or 10
Sensitivity	nA/ppm	[2,100]
Measurement accuracy	ppm	[0,1]
Response time	s	8,15 or 30
Start time	ms	70 or 300
Energy consumption	uW	2.7 or 50

A. EXPERIMENTAL SETTINGS

1) DATASETS

There is no public dataset to provide large-scale sensor contextual information. We found a public sensor website "Array of Things", which contains 22 different sensors of image processing, environment and air quality detecting. And this sensors collect data from 12 locations, such as Chicago and Taiwan. In terms of recommending sensors based on sensor attributes, the website has less types of sensors than we need. Therefore, we conducted a combination of real data of this website and synthetically generated data according to the range of sensor attribute data of the public sensor website "Array of Things". This combination of data allows to provide a large amount of sensor data and helps to better understand the behavior of sensor recommendation algorithms. The value range of sensor attributes is shown in the table 2. We take six sensor attributes mentioned above into account, in which three sensor attributes have fixed value while others have value range. In order to test the influence of dataset size on the model, we conduct four datasets of different sizes, and more details of these datasets are shown in table 3.

2) COMPARATIVE METHODS

- **TOPSIS**: Firstly, the method [27] normalizes the sensor dataset matrix. Next, computing the best point and worst point according to the objective function. Then computing the distance of every sensor in sensor dataset matrix to the best point and worst point. Finally, ranking and recommending it to users.
- **ES**: Firstly, the method [11] uses the TOPSIS to rank and sort the sensor dataset. Next, setting the SR parameter

TABLE 3. Statistics of sensor datasets.

Dataset	Dataset 1	Dataset 2	Dataset 3	Dataset 4
The total number	10000	50000	100000	200000
Uv sensors	777	3858	7632	15389
Printing sensors	778	3877	7738	15281
Carbon monoxide sensors	801	3924	7622	15459
Pressure sensors	780	3837	7746	15298
Temperature sensors	736	3834	7621	15240
Humiture sensors	732	3856	7594	15306
Humidity sensors	810	3717	7668	15492
Acceleration sensors	750	3858	7736	15165
Optical sensors	763	3910	7629	15457
Sulfur dioxide sensors	757	3858	7750	15357
Nitrogen dioxide sensors	752	3774	7835	15378
Magnetic sensors	780	3898	7710	15690
Ozone sensors	784	3799	7719	15488

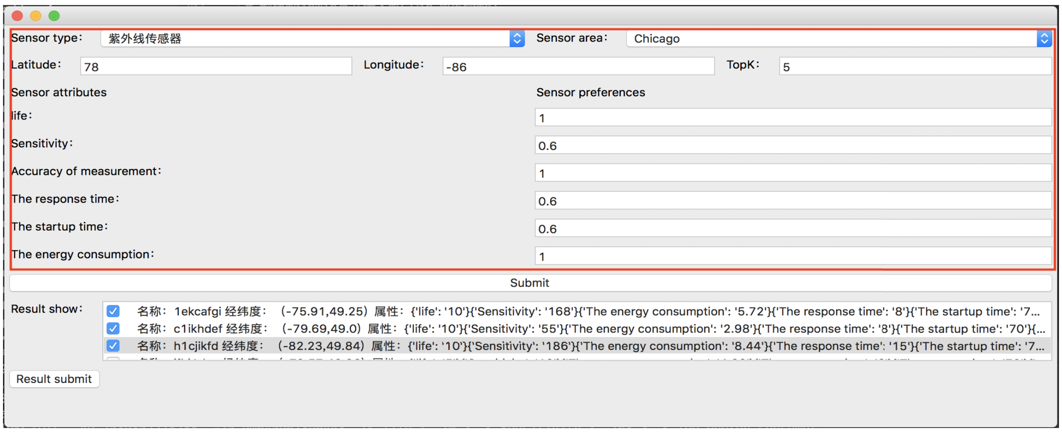


FIGURE 2. The user interface.

to limit the number of sensor dataset inputed to the fast nondominated sort algorithm. Finally, the N options that belong to the nondominated front are recommended.

- **Dynamic skyline algorithm**: Firstly, this method [12] gets the user request and computes the local dynamic skyline to narrow the sensor dataset. Then, collecting a set of locally filtered sensor data to compute the dynamic skyline. Finally, using SAW algorithm to rank sensor dataset.

3) EVALUATION METHODS

In order to evaluate the efficiency of the model, the response time and the accuracy are selected as evaluation indexes to judge the proposed model. After getting sensor dataset by using user's sensor requirements to retrieving R-tree, we keep track of time until the recommended sensor is fed back to the user. This time interval is response time. As a sensor selection model, we employ a proportion of the number of user satisfying sensor options in the top-K options as the accuracy to measure the proposed sensor selection model. The accuracy formula is defined as:

$$Accuracy = \frac{U}{K} \tag{7}$$

where U represents the number of sensor options which user is satisfied, and K is the number of sensors options recommended by the model;

4) THE USER INTERFACE

In this model, users are required to input data such as sensor location, type and sensor attribute value preference. In order to facilitate users to operate, we design the user page. shown in figure 2.

As shown in figure 2, after fill the sensor information in the red square circle, the user click the "Submit" button to submit the information to the server. Then the server puts the descriptive information of the sensor recommended by model in "Result show" module. Finally, the users need to select content options in "Result show" module and submit results.

B. EXPERIMENTAL RESULTS

In this part, we evaluate the model proposed in this paper from three aspects: impact of parameter setting, response time comparison of sensor selection algorithm and accuracy comparison of sensor selection algorithm.

1) IMPACT OF PARAMETER SETTING

We need ensure an important parameter in our model, which is the number of sensor options to be processed by an improved fast nondominated algorithm. Before we conduct

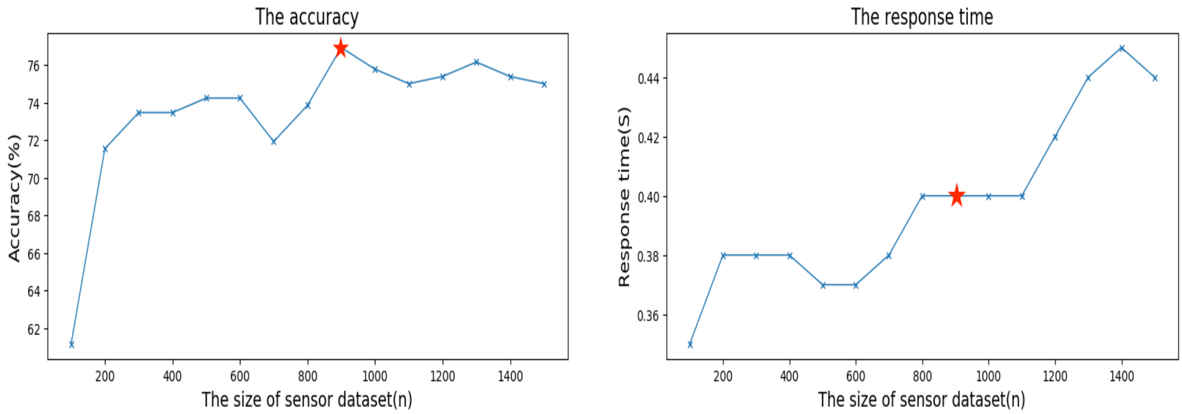


FIGURE 3. **Comparison experiments of parameter setting.**

the experiments, we need determine which dataset we need to use. In order to simulate sensor resources selection problem in the real situation, in which has a large number of sensor resources, ES model proposed by Nunes [11] and dynamic skyline model proposed by Kertiou [12] adopt multiple datasets to measure the performance of the model. And These two models has the best performance when the size of dataset is 100,000. Therefore, combined with the dataset settings in other papers, dataset3 is selected for experiment in the parameter setting part.

We conduct 15 groups experiments based on dataset3 as shown in figure 3, and parameter n is set from 100 to 1500. Experimental results including the response time and the accuracy are the average of 30 users' experimental records. According to the experimental results, the trend of accuracy is on the rise. The highest accuracy is 76.92% with that n equals to 900, and the lowest is 61.15% with that n equals to 100. When parameter n is set from 900 to 1500, the accuracy of the model reaches a stable state and fluctuates around 75%. The general trend of response time is also on rise with the increase of the number of parameter n. However, response time fluctuates greatly, and the reasons for the large fluctuation of response time include the size of sensor dataset and the duplication of sensor options in the dataset. The highest of response time is 0.45 seconds while n equals 1400, and the lowest of response time is 0.35 seconds while n equals to 100.

According to the results shown in 15 groups experiments, the accuracy and response time of our proposed model increase with adding the number of parameter n, but the rangeability is different. The rangeability of accuracy is 15.77% while the rangeability of response time is 0.1 seconds. In addition, The main purpose of our model is to improve the precision of sensor recommendation model, so we choose the case with high accuracy first when choosing the parameter n. we set n as 900 when the accuracy is the highest, and the corresponding response time is 0.40 seconds. To sum up, we set the experiment parameter n to 900, and the following experiments are all conducted with that n equals to 900.

2) RESPONSE TIME COMPARISON OF SENSOR SELECTION ALGORITHM

As an interactive system, it needs to consider the satisfaction of users. We require the time interval as small as possible, so the response time is taken as an indicator to measure the sensor selection model. In order to test the performance of the sensor selection model in the area of response time, we conduct experiments under two conditions: "different group of the same dataset" and "same group of different datasets".

TABLE 4. **Comparison of response times for the different group of dataset1.**

group	1	2	3	4	5
TOPSIS	0.220	0.214	0.211	0.217	0.229
ES	12.936	13.031	13.029	12.168	12.810
Dynamic Skyline	3.766	4.093	3.931	4.026	4.010
Our	0.219	0.214	0.218	0.217	0.217

In the premise of the same dataset, user's experimental records are divided into uneven groups and the mean response time of each group are calculated. The stability of the four sensor selection algorithms of TOPSIS, ES, dynamic skyline algorithm and our model was measured by comparing the difference between the highest group and the lowest group. In same dataset, user' s experimental records is divided into 5 groups, the first group contains 5 experimental records, the second group consists of 10 experimental records, the third group consists of 15 experimental records, the fourth group contains 20 experimental records, and the last group contains 25 experimental records. The specific experimental results are shown in table 4, table 5, table 6 and table 7. In the experimental results of dataset1, the response time differences of the four methods among different groups are 0.001 seconds, 0.04 seconds, 0.266 seconds and 0.003 seconds respectively. In the experimental results of dataset 2, the response time differences of the four methods among different groups are 0.018 seconds, 0.863 seconds,

TABLE 5. Comparison of response times for the different group of dataset2.

group	1	2	3	4	5
TOPSIS	0.220	0.214	0.211	0.217	0.229
ES	12.936	13.031	13.029	12.168	12.810
Dynamic Skyline	3.766	4.093	3.931	4.026	4.010
Our	0.219	0.214	0.218	0.217	0.217

TABLE 6. Comparison of response times for the different group of dataset3.

group	1	2	3	4	5
TOPSIS	0.640	0.526	0.534	0.543	0.541
ES	13.087	13.243	13.210	13.291	13.102
Dynamic Skyline	9.596	8.779	8.942	9.012	8.921
Our	0.390	0.391	0.409	0.445	0.452

TABLE 7. Comparison of response times for the different group of dataset4.

group	1	2	3	4	5
TOPSIS	1.071	1.056	1.126	1.117	1.122
ES	14.046	16.794	15.287	14.418	13.920
Dynamic Skyline	19.881	22.283	21.024	20.148	20.277
Our	0.859	0.916	0.985	0.986	0.972

0.327 seconds and 0.005 seconds, respectively. In the experimental results of dataset 3, the response time difference of the four methods among different groups are 0.014 seconds, 0.204 seconds, 0.817 seconds and 0.062 seconds, respectively. In the experimental results of dataset 4, the difference in response time of the four methods among different groups are 0.07 seconds, 2.748 seconds, 2.402 seconds and 0.113 seconds respectively. According to the experimental results, TOPSIS and the proposed model have the best stability in response time, and the general trend of ES algorithm and dynamic skyline algorithm in response time is steady with the capacity of datasets increasing.

TABLE 8. Comparison of response times for the same group of different datasets.

Dataset	10000	50000	100000	200000
TOPSIS	0.042	0.219	0.553	1.086
ES	1.752	12.701	13.327	14.254
Dynamic Skyline	1.087	3.989	8.916	19.932
Our	0.076	0.216	0.421	0.965

Assuming that the number of user experimental results recorded is the same, we calculate the average response time of different methods with the same dataset, and finally measure the four sensor selection algorithms in response time by comparing the values. As shown in table 8, the response time of TOPSIS, ES, dynamic skyline and our proposed model increases on the rise of the size of dataset, but the increase rate is different. The response time of TOPSIS algorithm, dynamic skyline algorithm and our model in dataset1 as a benchmark, while the increasing ratio of three datasets comparing with dataset1 are 5, 10 and 20, TOPSIS algorithm corresponding response time is 0.219 seconds, 0.553 seconds, 1.086 seconds, dynamic skyline algorithm corresponding response time is 3.9989 seconds, 8.914 seconds, and 19.932 seconds, our model corresponding response time are 0.216 seconds, 0.421 seconds and 0.965 seconds. The response time of TOPSIS algorithm, dynamic skyline algorithm and our model increases at roughly the same rate as that the size of dataset increase. The response time of ES algorithm in the dataset2, dataset3 and dataset4 are 12.701 seconds, 13.327 seconds, and 14.254 seconds, because of the fast nondominated sorting algorithm takes up most processing time in ES algorithm, and that ES algorithm sets a threshold and SR limits the size of date to be inputed in the fast nondominated sorting algorithm, So there is little difference in response time.

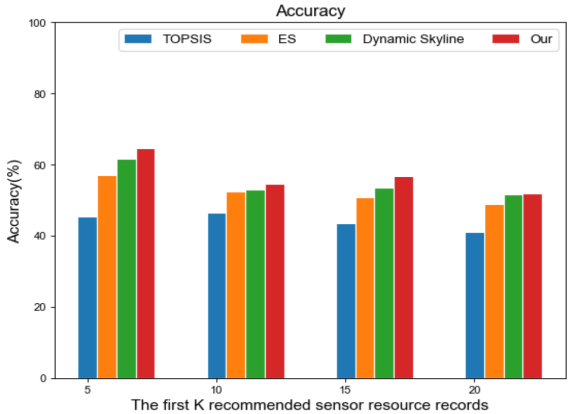
In terms of time complexity, TOPSIS algorithm has time complexity of $O((n)^2)$, ES algorithm has time complexity of $O((mn)^2)$ (n <= 2095), dynamic skyline algorithm has time complexity of $O((mn)^2)$, and our model has time complexity of $O((n)^2)$ (n <= 900), Where n represents the number of sensor records and m represents the number of sensor attributes. According to the time complexity, the advantages of the four sensor recommended models in response time are explained, and the response time of our proposed model is the minimum.

3) ACCURACY COMPARISON OF SENSOR SELECTION ALGORITHM

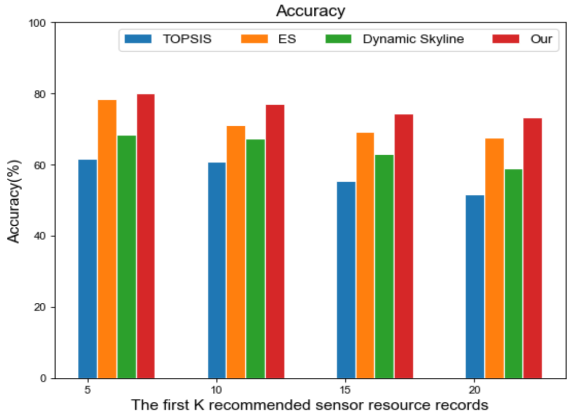
As a sensor selection model, it is also necessary to consider the accuracy of sensor resources that model fed back to users. In this section, the ratio between the number of sensor resources that users are satisfied with and the number of sensor resources that the model recommends to users is taken as the accuracy to measure the performance of the sensor selection model. In different datasets, we need to calculate the accuracy of the top-k options recommended by TOPSIS, ES, dynamic skyline and our model respectively. In this experiment, we set the average value of 120 experimental records of users as accuracy and set k as 5, 10, 15 and 20. the statistical results are shown in the figure 4.

In dataset1, TOPSIS algorithm has the highest accuracy in the top ten sensor selection records, which is 46.53%. ES algorithm, dynamic skyline algorithm and our proposed model all have the highest accuracy in the top five sensor selection records, which are 56.92%, 61.54% and 64.62%. TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our model all have the lowest accuracy in top 20 sensor selection records, which are 40.96%, 49.04%, 51.73% and 51.92%. When K is 5, 10, 15 and 20, the accuracy ranking of the four sensor selection algorithms ranges from large to small: our model, dynamic skyline algorithm, ES algorithm and TOPSIS algorithm.

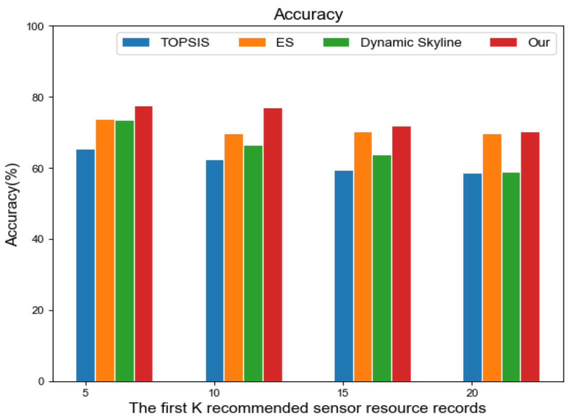
In dataset2, TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our proposed model all have the highest



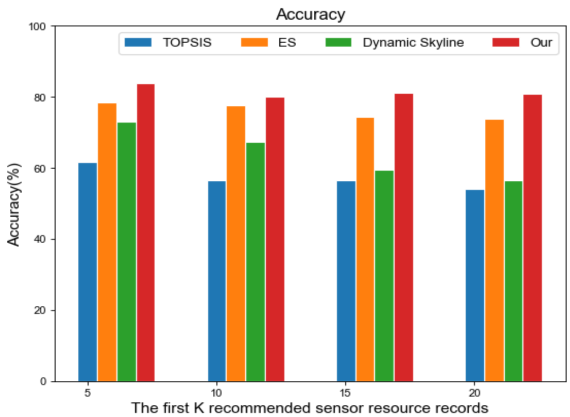
(a) dataset1



(b) dataset2



(c) dataset3



(d) dataset4

FIGURE 4. Accuracy in different datasets.

accuracy in the top five sensor selection records, which are 61.54%, 78.46%, 68.46% and 80.00%. TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our model all have the lowest accuracy in top 20 sensor selection records, which are 51.73%, 67.69%, 58.85% and 73.30%. When K is 5, 10, 15 and 20, the accuracy ranking of the four sensor selection algorithms ranges from large to small: our model, ES algorithm, dynamic skyline algorithm and TOPSIS algorithm.

In dataset3, TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our proposed model all have the highest accuracy in the top five sensor selection records, which are 65.38%, 73.84%, 73.46% and 77.69%. TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our model all have the lowest accuracy in top 20 sensor selection records, which are 58.65%, 69.61%, 59.03% and 70.19%. When K is 5, 10, 15 and 20, the accuracy ranking of the four sensor selection algorithms ranges from large to small: our model, ES algorithm, dynamic skyline algorithm and TOPSIS algorithm.

In dataset4, TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our proposed model all have the highest accuracy in the top five sensor selection records, which are 61.53%, 78.46%, 73.07% and 83.84%. TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our model all have the lowest accuracy in top 20 sensor selection records, which are54.03%, 73.84%, 56.53% and 80.96%. When K is 5, 10, 15 and 20, the accuracy ranking of the four sensor selection algorithms ranges from large to small: our model, ES algorithm, dynamic skyline algorithm and TOPSIS algorithm.

According to top five sensors recommended options, the accuracy of the four algorithms increase with the size of dataset rising. The highest accuracy of TOPSIS algorithm is 65% in the dataset3, the highest accuracy of ES algorithm is 78% in the dataset4, the highest accuracy of dynamic skyline algorithm is 73% in the dataset3, and the highest accuracy of our model is 83% in the dataset4. The accuracy growth of TOPSIS, ES, dynamic skyline algorithm and our model for top five sensors recommended options are 20%, 21.54%, 11.54% and 18.85%, among which our model and dynamic skyline algorithm have good stability. Although the accuracy of the four sensor selection algorithms fluctuates greatly, they will eventually reach a stable state. The accuracy of

TOPSIS algorithm, ES algorithm, dynamic skyline algorithm and our model in stable state is 65.38%, 78.46%, 73.46% and 83.84% respectively. In conclusion, our model has the highest accuracy and good stability.

V. CONCLUSION

In this paper, the sensor selection model based on an improved fast nondominated algorithm is proposed, which overcomes the low accuracy of TOPSIS algorithm and the high time complexity of the fast nondominated sorting algorithm. First, the R-tree is built to quickly retrieve sensor resources, which reduces the time of repeated retrieval. We take the user's preference of sensor attribute into account to narrow sensor dataset, and the size of sensor dataset is set to 900 after experiments where the accuracy of our proposed model is the highest. Then, an improved fast nondominated sorting algorithm is used to get the nondominated front to improve the accuracy of the model. Finally, we use TOPSIS algorithm to sort the sensor dataset and recommend them to users. To illustrate the usability of the model, we conducted experiments on four datasets of different sizes. In terms of accuracy, our proposed model has the highest accuracy on four datasets. In terms of the response time, our proposed model has more superiority than others with the size of sensor dataset increasing because our model restricts the number of sensor options dealt with an improved fast nondominated sorting algorithm. Experimental results on four datasets show that our proposed model has high accuracy and low response time in sensor selection problem.

Current sensor data is stored semantically, which provides a good opportunity for cross-platform data processing. In order to improve the compatibility and extensibility of sensor selection model, we plan to extension the sensor selection model in the context of semantic Internet of things.

REFERENCES

[1] D. Evans, "The Internet of Things: How the next evolution of the Internet is changing everything," Cisco, San Jose, CA, USA, White Paper, 2011.
[2] L. H. Nunes, J. C. Estrella, L. H. V. Nakamura, R. de Libardi, C. H. G. Ferreira, L. Jorge, C. Perera, and S. Reiff-Marganiec, "A distributed sensor data search platform for Internet of Things environments," *Netw. Internet Archit.*, vol. 4, no. 1, pp. 1–12, 2016.
[3] C. Perera, A. Zaslavsky, P. Christen, M. Compton, and D. Georgakopoulos, "Context-aware sensor search, selection and ranking model for Internet of Things middleware," in *Proc. IEEE 14th Int. Conf. Mobile Data Manage.*, Jun. 2013, pp. 314–322.
[4] H. Gong, Y. Shi, and G. Li, "Context-aware sensor search framework in semantic Web of things," in *Proc. 8th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, Aug. 2016, pp. 94–98.
[5] Y. Zhou, S. De, W. Wang, and K. Moessner, "Search techniques for the Web of things: A taxonomy and survey," *Sensors*, vol. 16, no. 5, p. 600, 2016.
[6] C. Truong, K. Römer, and K. Chen, "Fuzzy-based sensor search in the Web of things," in *Proc. 3rd IEEE Int. Conf. Internet Things*, Oct. 2012, pp. 127–134.
[7] Neha and S. Saxena, "Vector method for ranking of sensors in IoT," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Aug. 2016, pp. 1–5.
[8] D. Carlson and A. Schrader, "Ambient Ocean: A Web search engine for context-aware smart resource discovery," in *Proc. IEEE Int. Conf. Internet Things (iThings)*, Sep. 2014, pp. 177–184.
[9] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," *J. Comput. Syst. Sci.*, vol. 66, no. 4, pp. 614–656, Jun. 2003.
[10] X. Zhang and J. Chomicki, "Semantics and evaluation of top-k queries in probabilistic databases," *Distrib. Parallel Databases*, vol. 26, no. 1, pp. 67–126, 2009.
[11] L. H. Nunes, J. C. Estrella, C. Perera, S. Reiff-Marganiec, and A. C. B. Delbem, "The elimination-selection based algorithm for efficient resource discovery in Internet of Things environments," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf.*, Jan. 2018, pp. 1–7.
[12] I. Kertiou, S. Benharzallah, L. Kahloul, M. Beggas, R. Euler, A. Laouid, and A. Bounceur, "A dynamic skyline technique for a context-aware selection of the best sensors in an IoT architecture," *Ad Hoc Netw.*, vol. 81, pp. 183–196, Dec. 2018.
[13] J. Wu, L. Chen, Qi Yu, L. Kuang, Y. Wang, and Z. Wu, "Selecting skyline services for QoS-aware composition by upgrading MapReduce paradigm," *Cluster Comput.*, vol. 16, no. 4, pp. 693–706, 2013.
[14] S. Pattar, R. Buyya, K. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Searching for the IoT resources: Fundamentals, requirements, comprehensive review, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2101–2132, 3rd Quart., 2018.
[15] B. Ostermaier, K. Römer, F. Mattern, M. Fahrmair, and W. Kellerer, "A real-time search engine for the Web of Things," in *Proc. Internet Things (IOT)*, Nov./Dec. 2010, pp. 1–8.
[16] R. Mietz and K. Römer, "Exploiting correlations for efficient content-based sensor search," in *Proc. IEEE SENSORS*, Oct. 2011, pp. 187–190.
[17] C. Truong and K. Römer, "Content-based sensor search for the Web of things," in *Proc. IEEE Global Commun. Conf.(GLOBECOM)*, Dec. 2013, pp. 2654–2660.
[18] P. Zhang, Y.-A. Liu, and F. Wu, "Matching state estimation scheme for content-based sensor search in the Web of things," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 11, 2015, Art. no. 326780.
[19] V. Vasilev, G. Iliev, V. Poulkov, and A. Mihovska, "Optimization of wireless node discovery in an IoT network," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–5.
[20] P. Zhang, Y.-A. Liu, F. Wu, W. Fan, and B. Tang, "Content-based sensor search with a matching estimation mechanism," *IEICE Trans. Commun.*, vol. E99.B, no. 9, pp. 1949–1957, 2016.
[21] L. H. Nunes, J. C. Estrella, C. Perera, S. Reiff-Marganiec, and A. C. BotazzozDelbem, "Multi-criteria IoT resource discovery: A comparative analysis," *Softw.-Pract. Exper.*, vol. 47, no. 10, pp. 1325–1341, 2016.
[22] S.-Y. Ihm, A. Nasridinov, and Y.-H. Park, "Grid-PPPS: A skyline method for efficiently handling top-*k* queries in Internet of Things," *J. Appl. Math.*, vol. 2014, May 2014, Art. no. 401618.
[23] H. Köhler, J. Yang, and X. Zhou, "Efficient parallel skyline processing using hyperplane projections," in *Proc. Int. Conf. Manage. Data*, Jun. 2011, pp. 85–96.
[24] F. H. Bijarbooneh, W. Du, E. C.-H. Ngai, X. Fu, and J. Liu, "Cloud-assisted data fusion and sensor selection for Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 257–268, Jun. 2016.
[25] P. Zhang, Y. Liu, F. Wu, S. Liu, and B. Tang, "Low-overhead and high-precision prediction model for content-based sensor search in the Internet of Things," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 720–723, Apr. 2016.
[26] M. Jiang, J. Luo, D. Jiang, J. Xiong, H. Song, and J. Shen, "A cuckoo search-support vector machine model for predicting dynamic measurement errors of sensors," *IEEE Access*, vol. 4, pp. 5030–5037, 2016.
[27] G.-H. Tzeng and J.-J. Huang, *Multiple Attribute Decision Making*. London, U.K.: Chapman & Hall, 2011, doi: 10.1201/b11032.
[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[29] K. K. Nithya, M. V. Prathap, and K. R. R. Babu, "Cluster oriented sensor selection for context-aware Internet of Things applications," in *Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things*, 2018, pp. 981–988.
[30] N. Ramachandran, V. Perumal, S. Gopinath, and M. Jothi, "Sensor search using clustering technique in a massive IoT environment," in *Industry Interactive Innovations in Science, Engineering and Technology*. 2018, pp. 271–281.
[31] S. Bharti, K. K. Pattanaik, and P. Bellavista, "Value of information based sensor ranking for efficient sensor service allocation in service oriented wireless sensor networks," *IEEE Trans. Emerg. Topics Comput.*, to be published.
[32] C. Zhang, Y. Liu, F. Wu, W. Fan, J. Tang, and H. Liu, "Multi-dimensional joint prediction model for IoT sensor data search," *IEEE Access*, vol. 7, pp. 90863–90873, 2019.

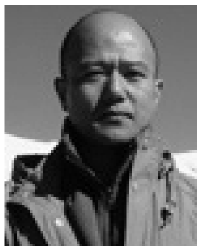
[33] M. Ebrahimi, E. Shafieibavani, R. K. Wong, and C.-H. Chi, "A new meta-heuristic approach for efficient search in the Internet of Things," in *Proc. IEEE Int. Conf. Services Comput.*, Jun./Jul. 2015, pp. 264–270.
[34] V. Deshpande, "An efficient clustering of sensors using a meta heuristic algorithm for IoT," *Int. J. Comput. Appl.*, vol. 173, no. 8, pp. 30–35, 2017.
[35] V. Sharma, R. Kumar, and R. Kaur, "UAV-assisted content-based sensor search in IoTs," *Electron. Lett.*, vol. 53, no. 11, pp. 724–726, May 2017.
[36] Q. Wang, S. Guo, J. Hu, and Y. Yang, "Spectral partitioning and fuzzy C-means based clustering algorithm for big data wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, p. 54, Dec. 2018.

ZENGWEI ZHENG received the B.S and M. Ec degrees in Computer Science and Western Economics from Hangzhou University, China in 1991 and 1998, respectively. He received the Ph. D degree in Computer Science and Technology from Zhejiang University, China in 2005. He is currently a professor of the Department of Computer Science and Engineering at Zhejiang University City College. His research interests include wireless sensor network, location-based service, internet of things, digital agriculture and pervasive computing. He is a member of the ACM and the CCF.

YANYUN TAO received her BS degree in Computer Science and Technology from Shandong university of science and technology in 2018. She is now a master candidate at College of Computer Science and Technology, Zhejiang University and is studying at Zhejiang University City College. Her research topic includes the Internet of Things and sensor recommendation.

YUANYI CHEN received his BSc degree from Sichuan University in 2010, the MSc degree from Zhejiang University in 2013 and PhD degree from Shanghai Jiao Tong University in 2017. From September 2014 to September 2015, he was a jointly supervised PhD candidate at Hong Kong Polytechnic University. He is currently a distinguished research fellow at the Department of Computer Science and Engineering at Zhejiang University City College, and a visiting scholar at University of California, Irvine. His research interest includes the Internet of Things, mobile computing and ubiquitous computing. He published more than 20 technical papers in major international journals and conference proceedings.

FENGLE ZHU received the PhD degree from Zhejiang University in 2014. From September 2014 to September 2016, she was a postdoctor in Mississippi State University. She is currently a lecturer at the Department of Computer Science and Computing at Zhejiang University City College. Her research interests include digital agriculture, image processing and machine learning.



DAN CHEN received the B.S. degree in electrical engineering and automation from Tianjin Polytechnic University in 1996 and the M.E. degree in computer Science from Zhejiang University in 2004. Since 2008, he has been an Associate Professor of computer science with the Zhejiang University City College. His research interests include computer network, mobile computing, IoT, and computer architecture.