A bio-inspired adaptive model for search and selection in the Internet of Things environment

Soukaina Bouarourou[1], Abdelhak Boulaalam[2] and El Habib Nfaoui[1]

[1] Computer Science Department, Faculty of Sciences Dhar EL Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[2] Computer Science Department, National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez, Morocco

ABSTRACT

The Internet of Things (IoT) is a paradigm that can connect an enormous number of intelligent objects, share large amounts of data, and produce new services. However, it is a challenge to select the proper sensors for a given request due to the number of devices in use, the available resources, the restrictions on resource utilization, the nature of IoT networks, and the number of similar services. Previous studies have suggested how to best address this challenge, but suffer from low accuracy and high execution times. We propose a new distributed model to efficiently deal with heterogeneous sensors and select accurate ones in a dynamic IoT environment. The model's server uses and manages multiple gateways to respond to the request requirements. First, sensors were grouped into three semantic categories and several semantic sensor network types in order to define the space of interest. Second, each type's sensors were clustered using the Whale-based Sensor Clustering (WhaleCLUST) algorithm according to the context properties. Finally, the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) was improved to search and select the most adequate sensor matching users' requirements. Experimental results from real data sets demonstrate that our proposal outperforms state-of-the-art approaches in terms of accuracy (96%), execution time, quality of clustering, and scalability of clustering.

Corresponding author
Soukaina Bouarourou,
soukaina.bouarourou@usmba.ac.ma

INTRODUCTION

The Internet of Things (IoT) is a technology used to connect people and physical devices (sensors, actuators, Radio-Frequency Identification (RFID), *etc.*) *via* the Internet, while continuously collecting and sharing data (Fig. 1) (*Perera et al., 2013b*). This interconnection defines three types of interactions: people to people, people to things/machine, and things/machine to machine/things (*Guillemin & Friess, 2009*).

The integration of the IoT and the web (Web of Things) had led to the creation of new kinds of services and applications in different domains: remote healthcare (*Kadhim et al., 2020*), indoor air quality (*Saini & Dutta, 2020*), vehicular traffic management (*Lalitha & Pounambal, 2020*), and air pollution monitoring (*Arora et al., 2019*).
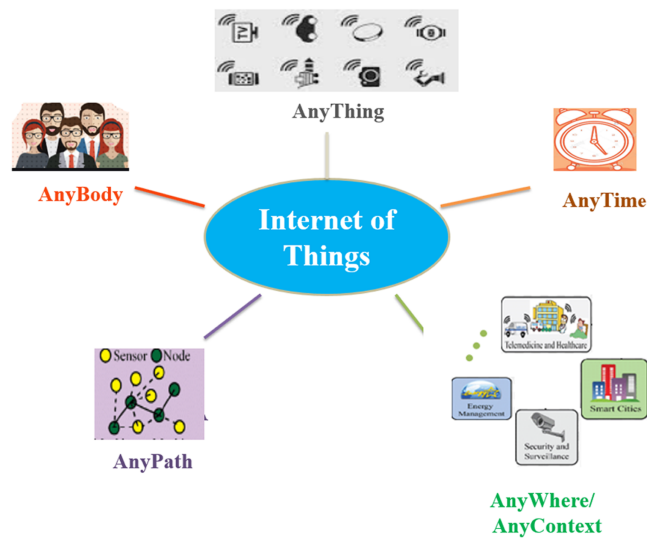
Figure 1 The definition of IoT.
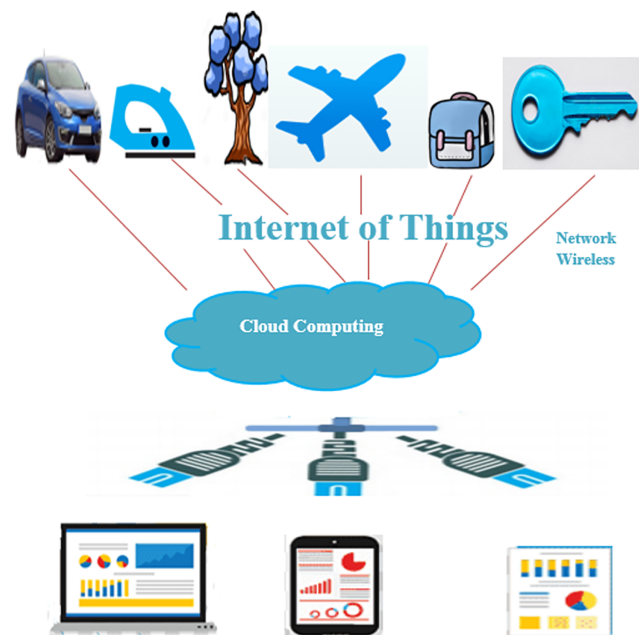
Figure 2 The integration of IoT and web services.

Figure 2 illustrates this integration. These applications have sought to create a smart environment between the real and the virtual that makes energy, transport, cities, and other areas more intelligent in order to enhance human life (*Patel & Patel, 2016*).

A market analysis company group predicted that there will be over 100 billion devices connected the IoT by 2025 (Global Sensors in Internet of Things (IoT) Devices Market, 2016–2022, https://www.bisresearch.com/industry-report/global-sensors-in-iot-devices-market-report-forecast.html). There will be a significant impact worldwide as the number of connected IoT devices that communicate, sense, and share information grows.

These interconnected devices generate a large amount of data, which must then be collected, analyzed, and used; this demonstrates we have an enormously searching space (*Zhang, Yang & Huang, 2011*).

Different interconnected sensors may be especially useful for monitoring and recording environmental phenomena such as temperature, sound, pollution levels, humidity, and wind (*Abdmeziem, Tandjaoui & Romdhani, 2016*). Therefore, it is important to determine which sensors/devices should be selected to retrieve the desired data. This wireless sensor network suffers from various limitations, including unbalanced energy consumption, hardware malfunction, deliberate attacks, physical localization, and weak security (*Laouid et al., 2017*, *2019*; *Mahmood, Seah & Welch, 2015*; *Saoudi et al., 2017*). It is impossible to collect all data from all of the existing sensors in the network due to the increasing number of networks. Thus, our main goal is to determine which sensors should be selected, taking into consideration of required processing time and overall accuracy.

We sought to overcome these limitations with an efficient approach to searching for and selecting the appropriate sensors for a query with minimal time consumption and a high quality. The proposed architecture consists of categorizing all current sensors in the network into three semantic categories (SCs); Society, Industry, and Environment. In parallel, for each semantic category, several semantic sensor network types (SSNTs) were created, and then the Whale-based Sensor Clustering Algorithm was applied to cluster the sensors using their contextual information. The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) was then used to select the appropriate sensors for the user query.

The remainder of this article is organized as follows: "Related works" is devoted to a description and review of the most studies focused on sensor search and selection techniques. The "Preliminaries" section includes a definition of the problem, the WhaleCLUST Algorithm, and the TOPSIS Algorithm. "Proposal" presents the proposed model on suitable way formulation. "Experiments" discusses the main implementation results. Finally, "Conclusion" provides a concluding summary and future work.

RELATED WORKS

Existing methods used to address sensor search issues can be categorized into content-based methods that look for sensors that generate such data, and context-based methods that use the sensors properties.

Content-based approaches

Content-based sensor searches can be conducted by processing and clustering the data generated by the different sensors, respectively. The aim is to obtain the optimal subset of devices that provide the desired data at any instance of time.

Bu (2018) proposed a fuzzy c-means approach to process and analyze data handling in IoT by which the data are grouped into clusters using the canonical decomposition method, which reduces the attributes of each object before loading the dataset into the memory. This approach enhances clustering efficiency and optimizes execution time,

although it employs only a limited number of attributes to evaluate data (*Tang & Fong, 2018*).

A meta-heuristic algorithm for clustering and automatic clustering on big IoT data was proposed by *Tang & Fong (2018)*. This partitioned clustering method optimizes the mini-batch and parallel partition-based Dynamic Group Clustering (DGC) for the IoT big data environment. The method is comprised of three activities: first, a sample of the dataset is partitioned into mini-batches; then the centroids of the mini-batches of data are adjusted; lastly, the mini-batches are collated to form clusters. This method maximizes the quality of the clusters related to the choice of the optimal batch size.

Another meta-heuristic algorithm for clustering and automatic clustering on big IoT data was suggested by *Jabeur et al. (2020)*. This approach has highlighted the impact of spatial events on IoT clustering. The performed clustering for IoT depends on the following six main features: location, energy, connectivity, users' requirements, communication, and semantics based on the Firefly Algorithm (FICA). This approach is inconsistent in its efficient selection of CH; however, it has registered more energy consumption and a high complexity.

Elahi et al. (2009) introduced a predictive model by calculating the estimation probability. Their model can find sensors satisfying a user query by predicting the current resources' real data. However, this model is limited in that it can be used only on sensors that present periodic patterns and, therefore, is not highly accurate.

Ostermaier et al. (2010) developed a real-time search engine for the Web of Things named DYSER. This predictive engine assists in finding real-world entities using statistical models to make predictions based on their real-time state and assists in the discovery of resources with a limited number of sensor data retrievals.

Truong, Römer & Chen (2012) used the fuzzy set to calculate a similarity score between two candidate sensors, which were then used to obtain a ranked list of matching sensors. This method becomes time-consuming when the number of sensors within the network is increased.

Tables 1 and 2 summarize and compare different content-based and context-based search techniques with the following parameters: number of used sensors, processing time, attributes or properties, accuracy, and dynamicity (*e.g.*, removal or connection of new sensors in the sensor network).

Context-aware approaches

The content-based method summarizes the processing of all collected data obtained from an enormous number of connected sensors in an IoT network; it is not practical to treat all the collected data, leading many researchers to choose context-based method approach. Context-awareness is an efficient technique for selecting data that requires additional processing and attention.

Context is the characterization of an object in an environment at any place and time; this is extracted from the expression of one or more IoT resources (*Abowd et al., 1999*). Below, we present an overview of works focused on a sensor search based on context properties.

Table 1 Functional and non-functional comparison of content-based search approaches.

Researches	Number of sensors	Properties	Execution time	Accuracy	Dynamicity
Elahi et al. (2009)	250	Unlimited	–	–	Unconsidered
Ostermaier et al. (2010)	385	Unlimited	–	–	Unconsidered
Truong, Römer & Chen (2012)	42	Unlimited	–	–	Unconsidered
Bu (2018)	Unlimited	Limited	650 min/64 Clusters	0.94/64 Clusters	Considered
Tang & Fong (2018)	12,000	100	1.20E+00	–	Considered
Jabeur et al. (2020)	800	Unlimited	–	–	Unconsidered

Table 2 Functional and non-functional comparison of context-based search approaches.

Researches	Number of sensors	Number of properties	Execution Time(s)	Accuracy	Dynamicity
Aberer, Hauswirth & Salehi (2007)	100,000	2	–	–	Unconsidered
Nath, Liu & Zhao (2007)	100,000	2	–	–	Unconsidered
Le-Phuoc et al. (2011)	100,000	2	–	–	Unconsidered
Perera et al. (2013a)	1,000,000	6	9.5	98%	Unconsidered
Hsu, Lin & Chen (2014)	50	1	600	–	Unconsidered
Ebrahimi et al. (2017)	1,000,000	6	8	92%	Considered
Nunes et al. (2017)	10,000	6	–	–	Unconsidered
Nunes et al. (2018)	209,555	6	785	–	Unconsidered
Kertiou et al. (2018)	1,000,000	6	9	95%	Unconsidered
Babu, Prathap & Samuel (2019)	20	4	2,009	91.99%	Unconsidered
Singh, Baranwal & Tripathi (2020)	100	9	2.85	–	Unconsidered
Zannou, Boulaalam & Nfaoui (2021)	3,000	4	–	–	Unconsidered

The Global Sensor Networks (GSN) (*Aberer, Hauswirth & Salehi, 2007*) is a data-processing engine aimed at providing flexible middleware to address sensor data integration and distributed query processing. This approach allows registering sensors with their meta-data in an XML structure. The query is a text-based search; however, this engine features ambiguous descriptions of sensors that users add. Similar to GSN model, the Microsoft SensorMap (*Nath, Liu & Zhao, 2007*) enables the user to choose appropriate sensors based on their location, type, and keywords.

Ebrahimi et al. (2017), *Kertiou et al. (2018)*, *Le-Phuoc et al. (2011)*, *Perera et al. (2013a)*, *(2013b)* addressed the issue of using the Semantic Sensor Network (SSN) Ontology (*Compton et al., 2012*) as the basis for conceptual modeling in the IoT domain (*Janowicz et al., 2010*). Among these works, Linked Sensor Middleware (LSM) (*Le-Phuoc et al., 2011*) is a platform that pairs real-world data with the Semantic Web in a unified model, where raw data are saved in relational databases, and the database schema is mapped to the ontology for improvement. This platform, however, offers limited functionalities for searching based on logical queries. For example, querying the resources relies on selecting

services based only on the sensor location and its type, assuming that the obtained data are static and not subject to frequent changes.

A context-aware sensor search selection and ranking model called CASSARAM (*Perera et al., 2014*; *Perera et al., 2013a*, *2013c*) suggests collecting point-based and proximity-based demands from users before plotting them in a multidimensional space. This method employs a distributed search by executing parallel processing over different server nodes to collect local high-ranking sensors. This collection and management of context-properties is a challenging approach for a large number of sensors due to its disregard for system performance in dealing with changes (add or delete) in the sensor network.

Ebrahimi et al. (2017) was inspired by the behavior of ants when developing the Ant-Clustering (AntCLUST) algorithm to cluster sensors based on their contextual information in the form of Sensor Semantic Overlay Networks (SSONs). The search queries are transmitted to each SSON to select the clusters that contain the most appropriate sensors. Although this strategy is meant to maintain its performance against dynamicity in the IoT, the system suffers from vulnerability to dynamicity issues and a time-consuming off-line computing phase.

Nunes et al. (2017) proposed a resource discovery and selection process based on the required parameters. The authors evaluated and compared the overall quality of this selection among three multi-objective decision methods: the Simple Additive Weight method (SAW), TOPSIS, and VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) with the Pareto optimal solutions.

The Elimination-Selection (ES) model (*Nunes et al., 2018*) combined the Fast-Non-Dominated sort algorithm and the multiple-criteria decision method TOPSIS to enhance quality and response time. The selections given by TOPSIS were reordered and chosen as the prominent best option by using a Fast-Non-Dominated sort, which was applied in an agricultural case study (*Nunes et al., 2016*) using a real dataset. The ES model increased the proportion of the non-dominated selections and increased the processing time by tens of times.

Hsu, Lin & Chen (2014) proposed an architecture allowing sensor search and selection based on identification of the sensors' characteristics, such as sensing range, accuracy, or residual energy among a large set of available sensors. SSN ontology was used to represent the properties of the sensor and this architecture has three components: server, gateway, and sensor. The user posts a request to the server that is connected to different gateways to offer a response. In the simulation study, however, they focused only on the network lifetime.

Kertiou et al. (2018) based their proposed architecture on work by *Hsu, Lin & Chen (2014)*, which employed the context information of sensors with dynamic skylines operators to decrease the search space and select the best sensors depending on the user request. This requires use of distributed gateways connected to a server, where each gateway responds to the users' local requests. There is lower time complexity of the dynamic skyline algorithm, but it does require users to input the ideal values of sensor properties (*Zheng et al., 2019*).

Table 3 Comparison of different search techniques approaches.

Approach	Search publication	Search approach	Architecture	Data used	Algorithm/ontology used	Prototype/simulation
Content-based sensor	Elahi et al. (2009)	Time-related search	Centralized	ETH & MERL	Single and multi-period prediction model	Simulation
	Ostermaier et al. (2010)	Real-time search	Centralized	Bicing	Aggregated prediction model	Web-based prototype
	Truong, Römer & Chen (2012)	Event-based search	Centralized	NOAA, IntelLab , MavHome	Fuzzy logic	Web-based prototype
	Bu (2018)	Time-related search	Distributed	eGSAD, sWSN (Zhang, Chen & Leng, 2015)	Fuzzy c-means algorithm	Simulation
	Jabeur et al. (2020)	Spatiotemporal-based search	Distributed	Randomly generated	Bio-inspired algorithms	Java-based prototype
Context-aware sensor	Perera et al. (2013a)	Context-based search	Distributed	Phenonet project, LSM project, and Bureau of Meteorology	SSN Ontology and Apache Jena SDB & TDB	Java-based prototype
	Ebrahimi et al. (2017)	Context-based search	Centralized	LSM project, Bureau of Meteorology, MesoWes and AirQuality sensor dataset	SSN Ontology (Compton et al., 2012)	Simulation
	Nunes et al. (2017, 2018)	Context-based search	Centralized	Open Weather Map	–	Simulation
	Hsu, Lin & Chen (2014)	Event-based search	Centralized	ViSIoT	SSN Ontology (Compton et al., 2012)	Web-based prototype
	Kertiou et al. (2018)	Context-based search	Centralized	LSM Project, Bureau of Meteorology, and Phenonet project	SSN Ontology (Compton et al., 2012)	Simulation

A comparison of the above-mentioned search techniques (context-based and content-based approaches) is summarized in Table 3, including the search approach, system architecture, data, algorithm/ontology, and prototype/simulation.

Babu, Prathap & Samuel (2019) presented a different context-aware reliable sensor selection method in the IoT environment. The proposed method senses secure data from a sensor that fulfills the user requirements. It also utilizes methods including user addition, context specification, location sensing, user context counting, reliable information storage in the cloud, and the ability to select the provider's best results. However, this method requires a more accurate sensor selection model based on context information and user priority for cyber-physical systems (CPS) such as smart farming, and smart cities.

Singh, Baranwal & Tripathi (2020) proposed an IoT framework for conducting the selection process in QoS-Aware Selection of IoT services, which assumed the sensor, the network, and application layers. The multi-criteria decision making (MCDM) was applied as a combination of two methods under the analytic hierarchy process (AHP) and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). The AHP method was used to measure the weight of the QoS criteria, and TOPSIS ordered the services. These methods define the QoS parameter based on three potential IoT components: things, computing entity, and communication entity. Without regard to illustrating the adaptability framework, they presented only the selection part of the service composition.

With the same aims as content-based and context-aware methods, *Zannou, Boulaalam & Nfaoui (2021)* paired sensor nodes. In this, the best pairing creates a better search for a required service in an optimal path and takes into regard the residual energy. This approach was only applied when the search process for a specific service was demanded by a sensor node in a social network.

There are several architectures and methods that may be used to search for a service based on the content and the context of sensors. However, each study has its restrictions, as mentioned above. The mechanisms to resolve a search task are specific to a given goal. Here, we propose an architecture consisting of several gateways distributed in the network and managed by a server. We applied new methods on this architecture to consider their limitations while keeping an acceptable response time and a low network overload. Furthermore, the deletion or addition of sensors is a crucial challenge that should be considered in the context of a dynamic IoT environment.

Preliminaries

This section includes the necessary background for understanding the context of our study, including the problem definition, WhaleCLUST, and TOPSIS algorithms.

Problem definition

An IoT network is composed of a set of devices (sensors, actuators, *etc.*), gateways, and a server. These devices have a specific communication/transmission, sensing /reception ranges, and different constraints in terms of energy consumption and processing capacities. The gateways should be able to collect and process the data generated from sensors based on their constraints. Each gateway manages its sub-network and the server manages those gateways in the same manner. There are many sensors in each sub-network (gateway) that can be abstracted as services and the combination of these devices within these services is the basis of the IoT application within IoT middleware solutions.

The search task for a requested service in a network is the most important functionality in IoT (*Barnaghi et al., 2012*) due to the huge number of sensors/services, similar services in each sub-network, and the consequently enormous amount of collected data. Thus, sensors may be clustered physically or virtually with minimal and sufficient context information. Figure 3 illustrates this distributed architecture.

When a greater number of available services are higher, the consumption will be greater; therefore, the search space must be minimized without affecting the solution quality.

Table 4 summarizes the main notations used.

WhaleCLUST algorithm

The Whale-based Clustering Algorithm was derived from a desire to improve the Whale Optimization Algorithm (WOA) (*Mirjalili & Lewis, 2016*) with the principle of clustering. We defined a population as a set of search agents, and each search agent
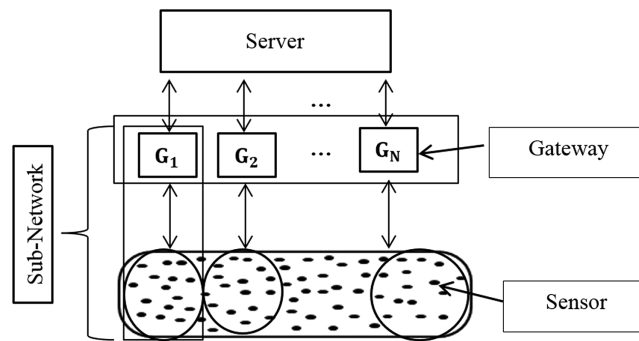
Figure 3 **The distributed environment.**

Table 4 **Notations used in the rest of this paper.**

Symbol	Description
$N_{category}$	The Number of Semantic Category (= 3).
SC_i	The Semantic Category (i = {environment, society, industry}).
$SSNT_{i,j}$	The Sensor Semantic Type Network of the SC_i and the type j.
$P_{h,k}$	The List of index sensor of the hth cluster and kth property.
$P_{h,k} (SSNT_{i,j})$	The existing cluster sensors in $SSNT_{i,j}$.
Q_j	The Matrix of center cluster of each cluster belonging to $SSNT_{i,j}$ for the SC_i.
$C_{h,k}$	Center cluster of the hth cluster and kth property.
N	The number of sensors existing in the hth cluster.
C_j	List of centers of all clusters in $SSNT_{i,j}$.
B_{center}	The best center cluster with required sensor.
$BC_{sensors}$	All sensors existing in BC.
B_{sensor}	The most similar sensor to required sensor.

determines k centers, where k is the number of clusters. Each search agent is defined as follows:

$$S_p = (Z_{p,1}, Z_{p,2}, \ldots, Z_{p,k})$$ (1)

where $Z_{p, j}$ represents the cluster center of the pth search agent in the cluster j.

We can represent the vector properties of a cluster center j named Z_j as follows:

$$Z_j = (z_{j,1}, \ldots, z_{j,m})$$ (2)

where $z_{j, i}$ denotes the value of the ith property of the jth cluster center, and m is the number of properties.

Hence, a swarm (search agent) refers to several candidate centers for a given area. Figure 4 shows how a population that contains S search agents at a certain iteration t is formed.

For each search agent, object X_i is first assigned to the nearest cluster center j that is verified by Eq. (3)
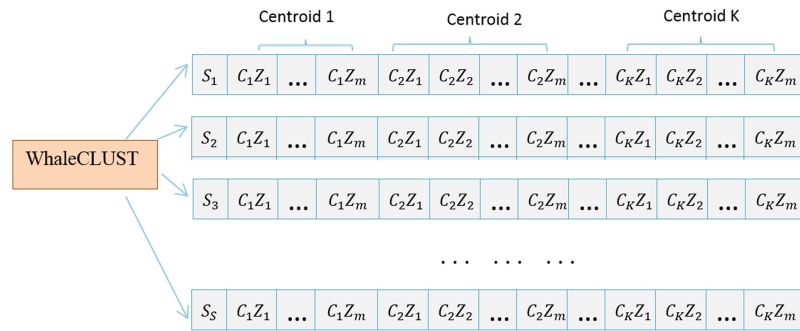
Figure 4 Population of S search agents at specific iterations.

$$d(X_i, Z_p) = \min_{j=1,2,\ldots,k} d(X_i, Z_{pj}) \tag{3}$$

$d(X_i, Z_p)$ corresponds to the squared Euclidean distance between the vector properties of object X_i, and the vector properties of cluster center Z_p, as shown below in Eq. (4):

$$d(X_i, Z_{pj}) = |X_i - Z_{pj}| \tag{4}$$

We computed the fitness function of each search agent S_p, based on the distance between each cluster center j (j belongs to S_p) and the vector properties of object X_i (i belong to cluster j), by minimizing the following Euclidean distance:

$$\text{fitness}(S_p) = \sum_{j=1}^{k} \sum_{i=1}^{N} W_{ij} d(X_i, Z_{pj}) \tag{5}$$

where k is the number of clusters/centers in search agent S_p, and W_{ij} is the association weight of object X_i and cluster j, and is defined by:

$$W_{ij} = \begin{cases} 1 & if \ d(X_i, Z_{pj}) = \min_{1 \leq j \leq k} d(X_i, Z_{pj}) \\ 0 & else \end{cases} \tag{6}$$

Simulating the spiral bubble-net feeding behavior process, the search agents update their position and orient themselves towards the best search agent. The description and the mathematical model for the spiral bubble-net feeding behavior process is broken down into three main aspects: encircling prey, bubble-net attacking method, and search for prey.

Encircling prey

Humpback whales can locate the position of prey and encircle it. The algorithm considers that the existing best search agent position is the target as a prey or close to the optimum point. The other search agents will enhance their position near the best search agent. This behavior is expressed by the following equations:

$$\vec{S}(t + 1) = \vec{S^*}(t) - \vec{A} \cdot \vec{D} \tag{7}$$

$$\vec{A} = 2.\vec{a}.\vec{r} - \vec{a} \tag{8}$$

$$\vec{D} = \vec{C}.\overrightarrow{S^*}(t) - \vec{S}(t) \tag{9}$$

$$\vec{C} = 2.\vec{r} \tag{10}$$

$\vec{S}(t+1)$ is defined as the new position of the search agent; $\overrightarrow{S^*}$ is the best search agent; $\vec{A}$ is the coefficient vector; $\vec{a}$ is linearly reduced within the range of 2 to 0 over the course of iterations; $\vec{r}$ is a random vector that varies in range [0, 1]; $\vec{D}$ is the distance between the vector position of the best search agent $\overrightarrow{S^*}$ and the current position of a search agent $\vec{S}$ at the current iteration t; '| |" is the absolute value; $\vec{C}$ is the coefficient vector; and . is an element-by-element multiplication.

Bubble-net attacking method (exploitation phase)

The bubble-net behavior of the humpback whale is characterized by two main mechanisms: shrinking encircling and spiral updating position. The two mechanisms have the same probabilities p (0.5 for each), where p is a random variable generated in the range [0, 1]. In the shrinking encircling mechanism, the value of $\vec{a}$ is decreased from two to zero over the total number of iterations in Eq. (8). The value of $|A|$ is also decreased between -a and a. The spiral updating position mechanism is applied between prey (the best search agent) and the position of whale (the current search agent) to simulate the helix-shaped movement of humpback whale. Setting random values for $|A|$ in the range [−1, 1], the search agent's new position can be defined as anywhere between the original position of the current search agent and the position of the best search agent as shown below:

$$\overrightarrow{D'} = \overrightarrow{S^*}(t) - \vec{S}(t) \tag{11}$$

where $\overrightarrow{D'}$ is the new distance between the current search agent and the best search agent.

The new position of the current search agent was updated as follows:

$$\vec{S}(t+1) = \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{S^*}(t) \tag{12}$$

where b is a constant that defines the logarithmic shape, l is a random number in [−1, 1], and '.' is an element-by-element multiplication.

The value of p is adopted to make the decision on the Equation used for updating the position of the current search agent; it is given as follows:

$$\vec{S}(t+1) = \begin{cases} \overrightarrow{S^*}(t) - \vec{A}.\vec{D} & p < 0.5 \\ \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{S^*}(t) & p \geq 0.5 \end{cases} \tag{13}$$

Search for prey (exploration phase)

Most meta-heuristic algorithms search for the optimum solution using a random selection. In the bubble-net method, the humpback whales randomly search for the best search agent when the values of $|A|$ are greater than 1 or less than −1. With this consideration, the

search agent moves far away from a reference whale (the best search agent chosen at first); it is expressed as follows:

$$\vec{S}(t+1) = \vec{S}_{rand} - \vec{A}.\vec{D} \tag{14}$$

$$\vec{D} = \vec{C}.\vec{S}_{rand} - \vec{S} \tag{15}$$

where $\vec{S}_{rand}$ is a random position vector.

The WhaleCLUST algorithm (Algorithm 1) initiates the search agent from a set of random solutions. The search agents update their position according to the above details.

The TOPSIS algorithm

The TOPSIS algorithm is a classical MCDA method and is of great interest to researchers in the subject. The TOPSIS algorithm is based on finding the best solution, where the best option is nearest to the ideal solution and farthest to the inferior anti-ideal solutions (*Tzeng & Huang, 2011*). The MCDA problem is typically defined by an M×N matrix Q_j called an analysis matrix. The element q_i matches the performance value of the i option regarding the decision criteria c_j, such as represented by Eq. (16).

$$Q_j = \begin{matrix} C_1 .... C_n \\ \begin{bmatrix} q_{11} & \cdots & q_{n1} \\ \cdots & \cdots & \cdots \\ q_{1m} & \cdots & q_{nm} \end{bmatrix} \end{matrix} \tag{16}$$

The TOPSIS algorithm can be summarized as:

The first step: the normalization of the analysis matrix Q to Q', the normalized value $r_{i,j}$ of each performance $c_{i,j}$ is calculated as follows:

$$r_{i,j} = \frac{q_{i,j}}{\sqrt{\sum_{j=1}^{m} q_{i,j}^2}} \tag{17}$$

The second step: the determination of the Ideal p_{+d} and Anti-ideal p_{-d} solutions using the decision matrix for each criterion. The formulas to compute p_{+d} and Anti-ideal p_{-d} for a maximization problem are:

$$\begin{cases} p_{+d} = max(r_{i,j}) \\ p_{-d} = max(r_{i,j}) \end{cases} \tag{18}$$

The third step: the calculation of the distances of each alternative (the normalized value $r_{i,j}$) from the Ideal and Anti-ideal solutions, according to the equation below:

$$\begin{cases} S_{+i} = \sqrt{\sum_{j=1}^{m} (r_{i,j} - p_{+d})^2} \\ S_{-i} = \sqrt{\sum_{j=1}^{m} (r_{i,j} - p_{-d})^2} \end{cases} \tag{19}$$

Algorithm 1 The Whale-based clustering algorithm.

Input: S_p (i = 1, 2, …, s): The population set of search agent.

Output: S′: The best search agent

1. **Procedure** Cluster _Formulation (Si):

2. Initialize data X_i (i = 1, 2, …, N)

3. Choose S search agent S_p (i = 1, 2, …, s) to contain k randomly cluster centers

4. **for** each search agent Si

5.   **for** each data vector X_m

6.   Calculate the distance using Eq. (4)

7.   Assign sensors to the nearest cluster using Eq. (3)

8.   **End for**

9. **End for**

10. S′: The best search agent

11. **While** t < Maximum_iteration

12.   **for** each search agent

13.   Update $\vec{a}$, $\vec{A}$, $\vec{C}$, l and p

14.   **if** p<0.5 **then:**

15.   **if** |A| < 1 **then:**

16.   Update search agents position using Eq. (7)

17.   **end if**

18.   **Else if** |A| ≥ 1 **then:**

19.   Select random search agent

20.   Update the position of current search agent by Eq. (14)

21.   **End if**

22.   **Else if** p < 0.5 **then:**

23.   Use Eq.(12)

24.   **End if**

25.   **End for**

26.   Update $\vec{a}$, $\vec{A}$, $\vec{C}$, l and p

27.   Update $\vec{S'}$ if there is a better solution

28. **End while**

29.   **return** S′

30. **End procedure**

The fourth step: the computation of the relative closeness C_{i+} of each solution to the ideal solution according to the following equation:

$$C_{i+} = \frac{S_{-i}}{S_{+i} + S_{-i}} \tag{20}$$

The fifth step: Sort the options in ascending order according to the C_{i+} value.

Our proposal

Multiple factors may impact the search for and selection of appropriate sensors that match the users' requirements in the Internet of Things environment. These include finding accurate sensors, optimizing time-consuming tasks for users, and managing system performance against the dynamicity of IoT. Therefore, we suggested a new method to search for and select an appropriate service considering the above-mentioned criteria simultaneously.

System architecture

To improve the performance of sensor search in IoT; including time-consumption and dynamicity, we propose classifying the current sensors in the network into three semantic categories and create several SSNTs (temperature, pressure, vibration, light, weather, *etc.*) in each category. The server extracts the semantic category, type, and values of the different properties from the user query. Next, the server sends them to the gateways. In each SSNT, sensors are clustered by the Whale-based Sensor Clustering Algorithm based on their context properties (accuracy, reliability, cost, availability, energy, *etc.*) before forwarding the queries to those related SSNTs. Consequently, the relevant cluster to the requested sensor properties is selected.

In order to ensure the parallel nature of IoT architecture and to minimize the network traffic caused by the dynamic context properties and number of sensors we chose a distributed architecture. Context properties are then stored in the local gateway and they are not globally updated.

The proposed architecture consists of three components: sensor, gateway, and server.

The sensors: these contain information and context properties and are registered within the gateways.

The gateway: this is responsible for managing a local network composed of three semantic categories of sensors clustered in the area of interest, and for connecting to the server.

The server: this is capable of processing users' requests, connecting to several gateways to obtain the local response from each gateway (local search and selection phases), and returning the response to the user (global search and selection phases).

Figure 5 depicts an overview of the system architecture and shows the following steps:

Connection: the user connects to the system and expresses its requirements *via* a system interface.

User requirements: the user's system sends the request (the user ID and user requirements) to the server.

Request: the server can extract the semantic category, type of the requested sensor, and users' requirements from a request. Next, it forwards them to the gateways. Figure 6 depicts a decomposed query (request).

Local search and selection: using the clustering process, each gateway determines the corresponding service according to the semantic category, type, and users' requirements from a request. The adequate service to the request is returned to the server. The details of this phase are described in the "Creation of SSNTs" section.
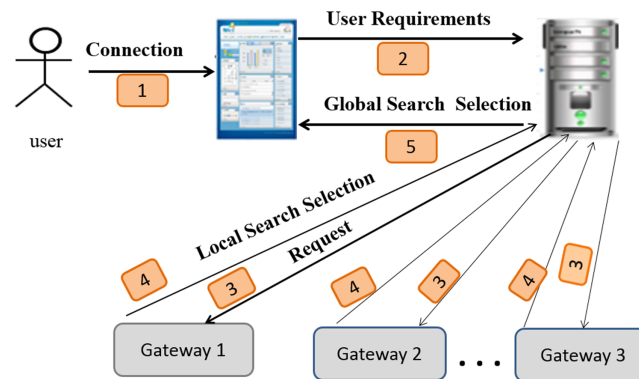
Figure 5 Sensor search architecture.
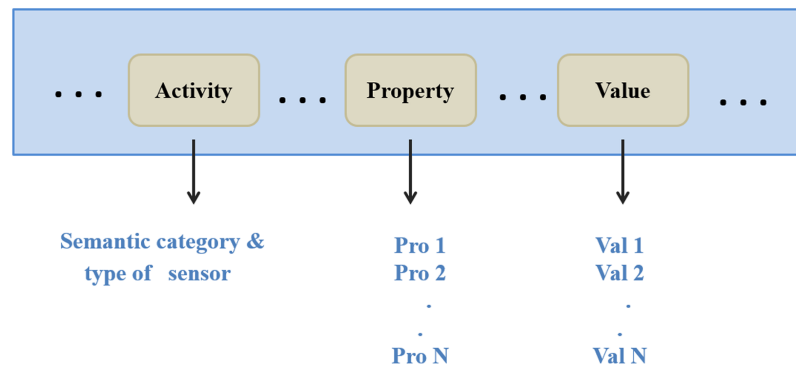
Figure 6 The request decomposition.

Global search and selection: the server determines the gateways that can provide the appropriate service (sensor) for each request.

Semantic modeling

We adopted the SSN Ontology (*Compton et al., 2012*) to model the sensor descriptions and context properties, and to publish the information of sensors in a standard format. *Compton et al. (2009)* presented a comparison of different semantic sensor ontologies is presented. The SSN ontology provides the most common context properties, such as accuracy, precision, drift, sensitivity, selectivity, measurement range, detection limit, response time, frequency, and latency. Figure 7 illustrates a segment of the SSN ontology used in our work.

Creation of SSNTs

The application of IoT can be categorized into three semantic categories (SCs) based on their focus (*Sundmaeker et al., 2010*; *Atzori, Iera & Morabito, 2010*): industry, environment, and society. For example: transportation and logistics (*Yuqiang, Jianlan & Xuanzi, 2010*), supply chain management (SCM) (*Chaves & Decker, 2010*), and aerospace, aviation, and automotive are some industry-oriented applications of IoT. Agriculture and breeding (*Shifeng et al., 2011*), disaster alerting, recycling, and environmental
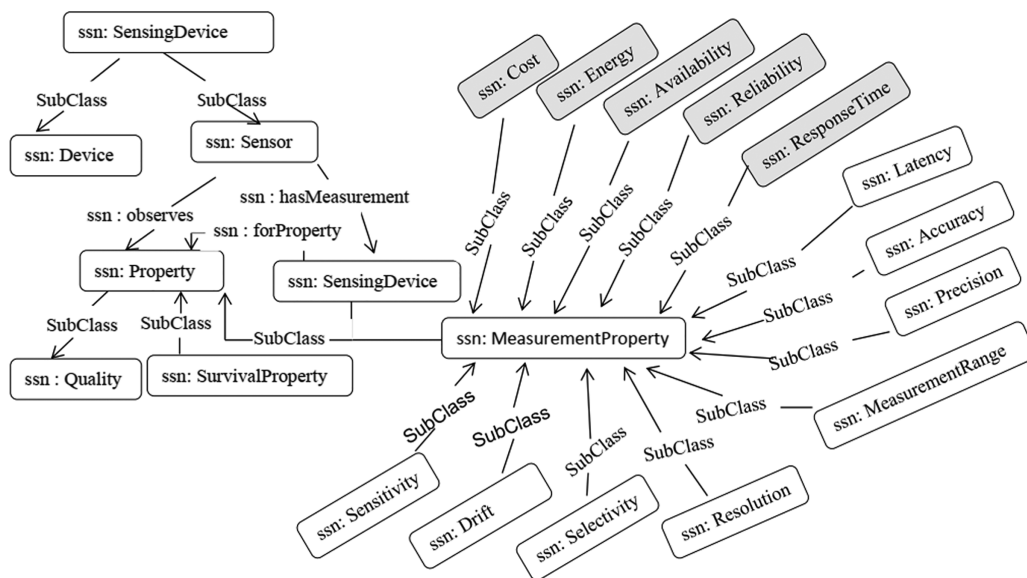
Figure 7 SSN ontology from a sensor perspective.

monitoring (*Jang, Healy & Skibniewski, 2008*) are some examples environment-oriented applications. Smart building (*Minoli, Sohraby & Occhiogrosso, 2017*), telecommunication, medical technology (*Lu & Liu, 2011*), healthcare, media, ticketing, and entertainment are some society-oriented applications.

We propose an unsupervised and a decentralized creation of SSNT to improve the efficiency of routing the queries to the appropriate SSNT (type). The SSNT is based on context properties (accuracy, reliability, energy, availability, cost, *etc.*) for the three SCs (environment, society, industry) in one of the gateways (Fig. 8). For this reason, we recommend a new bio-inspired meta-heuristic that mimics the bubble-net hunting and foraging behavior of a humpback whale (WhaleCLUST-A).

We assume that a given SSNT consists of N sensors, which are described by real-value m dimensional vector properties as follows:

$$X_i = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,m}\right) \tag{21}$$

where X_i is a vector property of sensor i, $x_{i,1}$ denotes the value of the jth property of the ith sensor, and m is the number of the properties.

The WhaleCLUST algorithm has been performed on each $SSNT_{i,j}$ to cluster sensors using their context information (availability, accuracy, reliability, response time, and cost), where i is a SC such that $i = \{1; 2; 3\}$ and j is the index of a SSNT.

The WhaleCLUST is regarded as a global optimizer because it has more abilities than other algorithm. The algorithm has a high exploration capability due to the position updating mechanism of Whales using Eq. (14), and a high emphasized exploitation and convergence, which initiate from Eqs. (7) and (12). These equations clarify that WhaleCLUST can run away from local minima with a quick convergence. WhaleCLUST
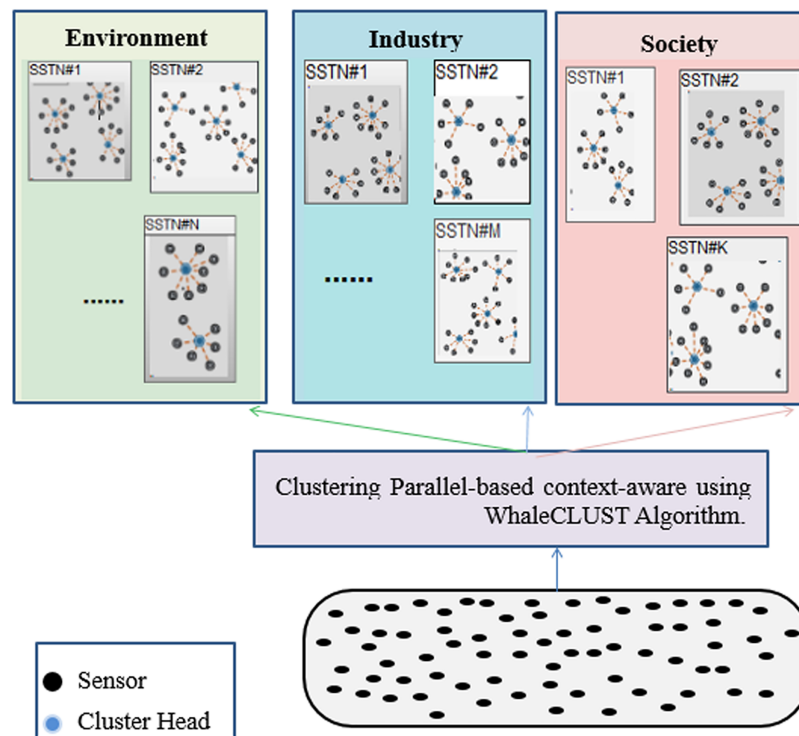
Figure 8 **Parallel-based context-aware clustering using the WhaleCLUST algorithm in one of the gateways.**

solves continuous and convex problems in addition to having a larger search area, simple structure, and is adaptable in dynamic conditions.

In particular, the mobile sensors will move from a gateway to another without any change in their context properties values. The gateway that receives the new joined sensor will apply the same search process by determining the most similar group based on the semantic category, SSNT, and distance similarity of each context property. The SSNT was obtained by the WhaleCLUST algorithm. Posteriorly, we calculated the center of each cluster h noted C_h as follows:

$$C_{h,k} = \frac{1}{n_h} \sum_{i=1, i \in cluster(h)}^{n_h} x_{i,k} \tag{22}$$

where $C_{h,k}$ is the value of the attribute k in the cluster h, n_h is the number of sensors in cluster h, and $x_{i,k}$ is the value of the attribute k in the sensor i. Algorithm 2 illustrates the steps for calculating the center of clusters.

Search for the relevant sensor

Here, we present the proposed technique for searching and selecting the relevant sensors in an IoT network to a given user query. The first step classifies the existing sensors into three SCs, the SSNTs within a clustered sensor are created using the WhaleCLUST algorithm in parallel for each SC. Then, Algorithm 2 is applied to the list of the centers of clusters j is elaborated as an analysis matrix with size M×N as follows:

Algorithm 2 Parallel calculation of the center of cluster.	
Input:	$N_{category}$: The number of categories (=3)
	SC_i: The set of categories
	$SSNT_{i,j}$: The type j corresponds to the category i.
	$P_h (SSNT_{i,j})$: The current *h*th cluster sensor in the $SSNT_{i,j}$
Output:	C_j: Centers of all clusters in $SSNT_{i,j}$
1.	**While** ($N_{category}$ < 3) **do**
2.	**For** each $SSNT_{i,j}$, p_h ($SSNT_{i,j}$) **do**
3.	Calculate center cluster of C_h using Eq. (22).
4.	Add C_h to C_j
5.	**End for**
6.	**End while**
7.	return C_j

$$Q_j = \begin{matrix} C_1 .... C_n \\ \begin{bmatrix} c_{11} & \cdots & c_{n1} \\ \cdots & \cdots & \cdots \\ c_{1m} & \cdots & c_{nm} \end{bmatrix} \end{matrix} \qquad (23)$$

Each element $c_{i,k}$ (the performance value) corresponds to the value of the center cluster i of attribute j (j named also criterion).

In terms of quality and response time, we improved the TOPSIS algorithm (*Rahim et al., 2018*). TOPSIS was applied in each center of cluster of a SSNT against all the used sensors, which minimized the time consumed to select and search for the requested services.

After employing the TOPSIS algorithm for each center cluster in $C_{h,k}$, the best center cluster B_{center} was selected; this was the most similar center cluster to the ideal sensors. Then, the list of all sensors belonging to the cluster $BC_{sensors}$ was extracted, where the Euclidean distance between the $BC_{sensors}$ and the B_{center} was computed to select the best sensor B_{sensor} between them. Algorithm 3 is shown in detail below.

The last phase is the global search selection. Here, the server receives the responses (the selected sensors) from the gateways. It chooses the appropriate sensors based on the user requirements by calculating the distance between each suggested solution (selected sensor) and the user requirements using Eq. (3), where a sensor with the minimal Euclidian distance is selected. The final solution provided by the service is then returned.

EXPERIMENTS

Experiments setup and data sets

We used a HP- EliteBook 8440p computer with Intel i5 CPU running at 2.40 GHz, under Windows 7 (64-bit) and 8 GB of RAM to evaluate our proposed method. Accordingly, the simulation software was written in Python3 using the WhaleCLUST algorithm and the TOPSIS search method.

Algorithm 3 Search and selection the best sensors.	
Input:	$N_{category}$: The number of categories (=3)
	SC_i: The set of categories
	C_j: Centers of all clusters in $SSNT_{i,j}$
Output:	B_{sensor}: List of centers of all clusters in $SSNT_{i,j}$

1.   **While** ($N_{category} < 3$) **do**
2.   **For** each $SSNT_{i,j}$ **do**
3.   Apply TOPSIS algorithm to C_j
4.   Add C_h to C_j
5.   Get B_{center}: the best center cluster (higher relative closeness value)
6.   **End for**
7.   Compute the Euclidean distance between each $BC_{sensors}$ and B_{center}.
8.   Get the B_{sensor} that has the minimal Euclidean distance with B_{center}
9.   **End while**
10. **return** B_{sensor}

Table 5 The parameters used in our model.

Parameters	Values
Number of iterations (t)	600
Number of search agents	50
Number of clusters (k)	3

It was difficult to determine some context properties related to the sensors in public datasets therefore we used a combination of real and synthetically generated data to evaluate our proposed method on a large scale. We collected datasets from the Linked Sensor Middleware (LSM) project (*Digital Enterprise Research Institute (2011)*, the Bureau of Meteorology (*Australian Government, Bureau of Meteorology (2012)*, and the Phenonet project (*Patni, Henson & Sheth, 2010*).

In all experiments, we considered five context properties (availability, accuracy, reliability, response time, and cost) and 10,000 sensors. We took the averages of some experimental results and the parameters used in the WhaleCLUST model are shown in Table 5.

We created a model for the proposed architecture and datasets to evaluate the robustness, efficiency, and parameter impact of our proposed approach. We also explained the results obtained from the different experiments. Finally, we compared AntCLUST (*Ebrahimi et al., 2017*) and ParticuleCLUST (*Wang et al., 2017*) as state-of-the-art methods. Our proposed method was performed with the highest accuracy and lowest processing time required to search and select the suitable sensors to the users' requirements.
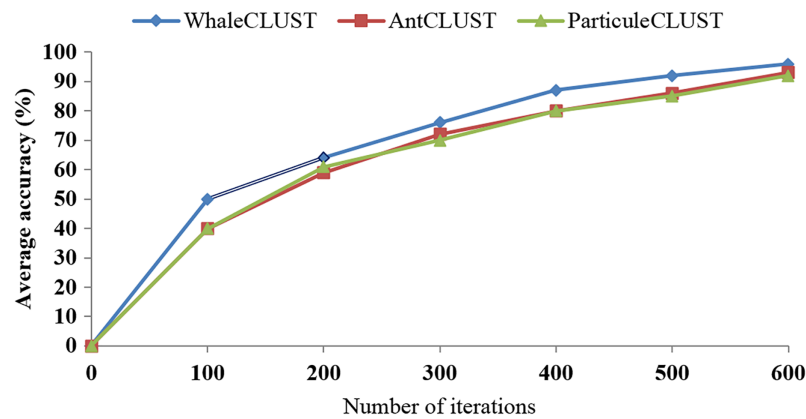
Figure 9 Accuracy comparisons of WhaleCLUST, AntCLUST, and ParticuleCLUST according to the number of iterations.
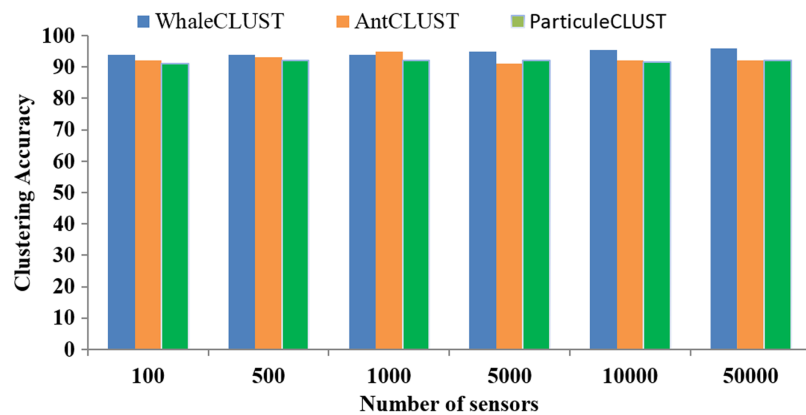
Figure 10 Accuracy comparisons of WhaleCLUST, AntCLUST, and ParticuleCLUST according to the number of sensors.

RESULTS AND DISCUSSION

We evaluated the accuracy of the WhaleCLUST algorithm and compared it with the AntCLUST and ParticuleCLUST algorithms using the number of iterations and the number of sensors. The greater the number of iterations and sensors the higher accuracy of our algorithm (Figs. 9 and 10). When the number of sensors exceeded 5,000, the accuracy of our algorithm was 96%, 92% for the AntCLUST, and 91% for the ParticuleCLUST. As a result, our algorithm was shown to be more efficient and scalable than the two other ones.

We used three trials with the number of search agents set to 10, 15, and 50 according to the number of iterations to evaluate the quality of clustering and the system performance when applying the proposed algorithm. The results show that the best quality of clustering was over 96% when the number of search agents was set to 50 (Fig. 11). This demonstrates the impact of change in the number of search agents on the quality of clustering.
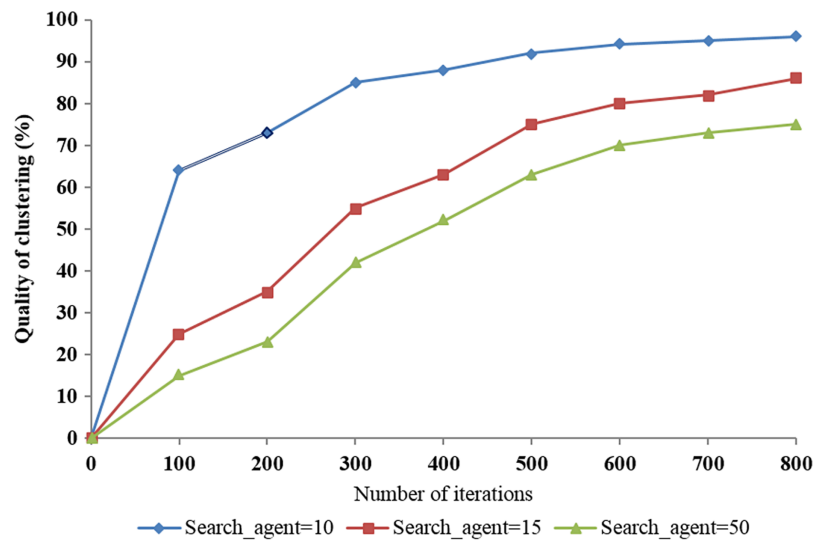
Figure 11 The impact of the number of search agents on the quality of clustering according to the number of iterations.
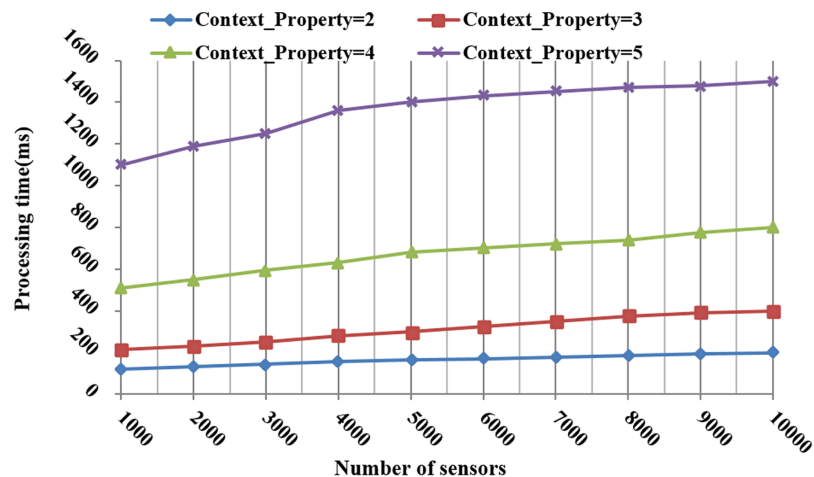
Figure 12 The processing time during the search and selection phases according to the number of context properties and sensors.

Figure 12 shows the processing time to respond to the request requirements with a different number of sensors and context properties. More time was consumed as the number of context properties and sensors grew. More precisely, when the number of sensors increased to 5,000, the consumed time slowly increased. This means that the search and selection tasks are efficient in a large space of interest.

To demonstrate the efficiency of the search and selection phases of our approach compared to the AntCLUST and the ParticuleCLUST approaches, we measured the required processing time (Fig. 13). When the number of sensors was more than 5,000, the consumed time is semi-equal for all approaches. However, when the number of sensors
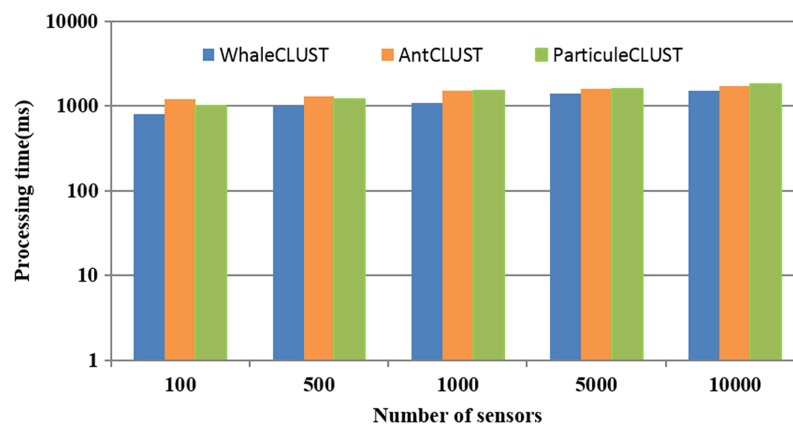
Figure 13 The performance of our proposed algorithm (WhaleCLUST) compared to the AntCLUST and ParticuleCLUSTalgorithms.

was more than 5,000, the consumed was less with our approach compared to the other approaches due to the distributed model used in our approach that processes the request on a parallelized way.

CONCLUSION

The advanced technology in the IoT environment allows access to multiple sensors that have similar utility. Searching and selecting the relevant and suitable sensors for user requirements is challenging. We studied the accuracy, time complexity, and scalability of the search and selection tasks to satisfy the requirements based on the context properties of sensors. We categorized the sensors into three semantic categories named environment, society, and industry. Within each SC, the SSNTs are created, where the improved Whale clustering is applied to cluster the sensors of SSNT according to the context properties. The distributed architecture consists of gateways that are connected to sensors and managed by a server, where the response is performed jointly and simultaneously among the server and the gateways. The experimental results showed that our proposed solution was promising in terms of accuracy, quality clustering, scalability, and execution time when compared to other approaches.

Furthermore, the current work can provide many benefits for practitioners and researchers who want to develop or integrate IoT applications to exploit the services provided by smart city, industry, agriculture, and healthcare systems. Our work is designed to support new technologies such as 5th generation and edge computing to efficiently and quickly respond to complex IoT requests.

We plan to apply our model in specific domains including pollution control based on storage capacity, energy consumption, and processing power. We will compare our study with works in the same context and domain and will consider the user request processing phase and the method used to extract the user requirements in future studies.

Funding
The authors received no funding for this work.

Competing Interests
The authors declare that they have no competing interests.

Author Contributions
- Soukaina Bouarourou conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Abdelhak Boulaalam conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- El Habib Nfaoui conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability
Our code for clustering and searching in IoT is available in the Supplemental File.

Supplemental Information
Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.762#supplemental-information.

REFERENCES

Abdmeziem MR, Tandjaoui D, Romdhani I. 2016. Architecting the Internet of Things: state of the art. *Robots and Sensor Clouds* **36**:55–75 DOI 10.1007/978-3-319-22168-7.

Aberer K, Hauswirth M, Salehi A. 2007. Infrastructure for data processing in large-scale interconnected sensor networks. In: *2007 International Conference on Mobile Data Management*. 198–205.

Abowd GD, Dey AK, Brown PJ, Davies N, Smith M, Steggles P. 1999. Towards a better understanding of context and context-awareness. In: Gellersen HW, ed. *Handheld and Ubiquitous Computing*. Berlin: Springer, 304–307.

Arora J, Pandya U, Shah S, Doshi N. 2019. Survey-pollution monitoring using IoT. *Procedia Computer Science* **155(4)**:710–715 DOI 10.1016/j.procs.2019.08.102.

Atzori L, Iera A, Morabito G. 2010. The Internet of Things: a survey. *Computer Networks* **54(15)**:2787–2805 DOI 10.1016/j.comnet.2010.05.010.

Australian Government, Bureau of Meteorology. 2012. List of station locations and sites. *Available at https://data.gov.au/dataset/ds-dga-900143f6-6582-49c5-bfd4-0838901d99c8/distribution/dist-dga-48261212-1415-4a28-a4d0-3677514d2bcf/details?q=.*

Babu KR, Prathap MV, Samuel P. 2019. Context aware reliable sensor selection in IoT. *International Journal of Intelligent Systems Technologies and Applications* **18(1–2)**:34–51 DOI 10.1504/IJISTA.2019.097746.

Barnaghi P, Wang W, Henson C, Taylor K. 2012. Semantics for the Internet of Things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)* **8(1)**:1–21 DOI 10.4018/IJSWIS.

Bu F. 2018. An efficient fuzzy c-means approach based on canonical polyadic decomposition for clustering big data in IoT. *Future Generation Computer Systems* **88(4)**:675–682 DOI 10.1016/j.future.2018.04.045.

Chaves LWF, Decker C. 2010. A survey on organic smart labels for the Internet-of-Things. In: *Seventh International Conference on Networked Sensing Systems (INSS)*. 161–164 2010.

Compton M, Barnaghi P, Bermudez L, Garcia-Castro R, Corcho O, Cox S, Graybeal J, Hauswirth M, Henson C, Herzog A. 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics* **17(4)**:25–32 DOI 10.1016/j.websem.2012.05.003.

Compton M, Henson CA, Lefort L, Neuhaus H, Sheth AP. 2009. A survey of the semantic specification of sensors. In: *2nd International Workshop on Semantic Sensor Networks, at 8th International Semantic Web Conference.*

Digital Enterprise Research Institute. 2011. Linked Sensor Middleware (LSM) Galway, Ireland. *Available at http://lsm.deri.ie/.*

Ebrahimi M, ShafieiBavani E, Wong RK, Fong S, Fiaidhi J. 2017. An adaptive meta-heuristic search for the Internet of Things. *Future Generation Computer Systems* **76(5)**:486–494 DOI 10.1016/j.future.2015.12.006.

Elahi BM, Romer K, Ostermaier B, Fahrmair M, Kellerer W. 2009. Sensor ranking: a primitive for efficient content-based sensor search. In: *2009 International Conference on Information Processing in Sensor Networks*. 217–228.

Guillemin P, Friess P. 2009. Internet of Things strategic research roadmap. The Cluster of European Research Projects, Tech. Rep.

Hsu Y-C, Lin C-H, Chen W-T. 2014. Design of a sensing service architecture for Internet of Things with semantic sensor selection. In: *2014 IEEE 11th International Conference on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conference on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conference on Scalable Computing and Communications and Its Associated Workshops*. Piscataway: IEEE, 290–298.

Jabeur N, Yasar AU-H, Shakshuki E, Haddad H. 2020. Toward a bio-inspired adaptive spatial clustering approach for IoT applications. *Future Generation Computer Systems* **107(4)**:736–744 DOI 10.1016/j.future.2017.05.013.

Jang W-S, Healy WM, Skibniewski MJ. 2008. Wireless sensor networks as part of a web-based building environmental monitoring system. *Automation in Construction* **17(6)**:729–736 DOI 10.1016/j.autcon.2008.02.001.

Janowicz K, Schade S, Bröring A, Keßler C, Maué P, Stasch C. 2010. Semantic enablement for spatial data infrastructures. *Transactions in GIS* **14(2)**:111–129 DOI 10.1111/(ISSN)1467-9671.

Kadhim KT, Alsahlany AM, Wadi SM, Kadhum HT. 2020. An overview of patient's health status monitoring system based on Internet of Things (IoT). *Wireless Personal Communications* **114(3)**:2235–2262 DOI 10.1007/s11277-020-07474-0.

Kertiou I, Benharzallah S, Kahloul L, Beggas M, Euler R, Laouid A, Bounceur A. 2018. A dynamic skyline technique for a context-aware selection of the best sensors in an IoT architecture. *Ad Hoc Networks* **81(2)**:183–196 DOI 10.1016/j.adhoc.2018.08.011.

Lalitha K, Pounambal M. 2020. IoT-based traffic management. In: *Emerging Research in Data Engineering Systems and Computer Communications*. Springer, 155–161.

Laouid A, Dahmani A, Bounceur A, Euler R, Lalem F, Tari A. 2017. A distributed multi-path routing algorithm to balance energy consumption in wireless sensor networks. *Ad Hoc Networks* **64(4)**:53–64 DOI 10.1016/j.adhoc.2017.06.006.

Laouid A, Dahmani A, Hassen HR, Bounceur A, Euler R, Lalem F, Tari A. 2019. A self-managing volatile key scheme for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing* **10(9)**:3349–3364 DOI 10.1007/s12652-018-0772-9.

Le-Phuoc D, Quoc HNM, Parreira JX, Hauswirth M. 2011. The linked sensor middleware-connecting the real world and the semantic web. *Proceedings of the Semantic Web Challenge* **152**:22–23.

Lu D, Liu T. 2011. The application of IOT in medical system. In: *2011 IEEE International Symposium on IT in Medicine and Education*. Piscataway: IEEE, 272–275.

Mahmood MA, Seah WKG, Welch I. 2015. Reliability in wireless sensor networks: a survey and challenges ahead. *Computer Networks* **79(4)**:166–187 DOI 10.1016/j.comnet.2014.12.016.

Minoli D, Sohraby K, Occhiogrosso B. 2017. IoT considerations, requirements, and architectures for smart buildings—energy optimization and next-generation building management systems. *IEEE Internet of Things Journal* **4(1)**:269–283 DOI 10.1109/JIOT.2017.2647881.

Mirjalili S, Lewis A. 2016. The Whale optimization algorithm. *Advances in Engineering Software* **95(12)**:51–67 DOI 10.1016/j.advengsoft.2016.01.008.

Nath S, Liu J, Zhao F. 2007. Sensormap for wide-area sensor webs. *Computer* **40(7)**:90–93 DOI 10.1109/MC.2007.250.

Nunes L, Estrella J, Nakamura L, de Libardi R, Ferreira C, Jorge L, Perera C, Reiff-Marganiec S. 2016. A distributed sensor data search platform for Internet of Things environments. *Available at http://arxiv.org/abs/1606.07932*.

Nunes LH, Estrella JC, Perera C, Reiff-Marganiec S, Botazzo Delbem AC. 2017. Multi-criteria IoT resource discovery: a comparative analysis. *Software: Practice and Experience* **47(10)**:1325–1341 DOI 10.1002/spe.2469.

Nunes LH, Estrella JC, Perera C, Reiff-Marganiec S, Delbem AC. 2018. The elimination-selection based algorithm for efficient resource discovery in Internet of Things environments. In: *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. Piscataway: IEEE, 1–7.

Ostermaier B, Römer K, Mattern F, Fahrmair M, Kellerer W. 2010. A real-time search engine for the Web of Things. In: *2010 Internet of Things (IOT)*. 1–8.

Patel KK, Patel SM. 2016. Internet of Things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *International Journal of Engineering Science and Computing* **6(5)**:1482 DOI 10.4010/2016.1482.

Patni H, Henson C, Sheth A. 2010. Linked sensor data. In: *2010 International Symposium on Collaborative Technologies and Systems*. 362–370.

Perera C, Zaslavsky A, Christen P, Compton M, Georgakopoulos D. 2013a. Context-aware sensor search, selection and ranking model for Internet of Things middleware. In: *2013 IEEE 14th International Conference on Mobile Data Management*. Piscataway: IEEE, 314–322.

Perera C, Zaslavsky A, Christen P, Georgakopoulos D. 2013b. Context aware computing for the Internet of Things: a survey. *IEEE Communications Surveys & Tutorials* **16(1)**:414–454 DOI 10.1109/SURV.2013.042313.00197.

Perera C, Zaslavsky A, Christen P, Georgakopoulos D. 2014. Context aware computing for the Internet of Things: a survey. *IEEE Communications Surveys Tutorials* **16(1)**:414–454 DOI 10.1109/SURV.2013.042313.00197.

Perera C, Zaslavsky A, Liu CH, Compton M, Christen P, Georgakopoulos D. 2013c. Sensor search techniques for sensing as a service architecture for the Internet of Things. *IEEE Sensors Journal* **14(2)**:406–420 DOI 10.1109/JSEN.2013.2282292.

Rahim R, Siahaan APU, Wijaya RF, Hantono H, Aswan N, Thamrin S, Sari DAP, Agustina S, Santosa RB, Muttaqin WM. 2018. Technique for order of preference by similarity to ideal solution (TOPSIS) method for decision support system in top management. *International Journal of Engineering and Technology* **7(3.4)**:290–293.

Saini J, Dutta M. 2020. Applications of IoT in indoor air quality monitoring systems. In: Raj P, Chatterjee J, Kumar A, Balamurugan B, eds. *Internet of Things Use Cases for the Healthcare Industry*. Cham: Springer DOI 10.1007/978-3-030-37526-3_4.

Saoudi M, Lalem F, Bounceur A, Euler R, Kechadi M-T, Laouid A, Bezoui M, Sevaux M. 2017. D-LPCN: a distributed least polar-angle connected node algorithm for finding the boundary of a wireless sensor network. *Ad Hoc Networks* **56(4)**:56–71 DOI 10.1016/j.adhoc.2016.11.010.

Shifeng Y, Chungui F, Yuanyuan H, Shiping Z. 2011. Application of IOT in agriculture. *Journal of Agricultural Mechanization Research* **7**:190–193.

Singh M, Baranwal G, Tripathi AK. 2020. QoS-aware selection of IoT-based service. *Arabian Journal for Science & Engineering* **45(12)**:10033–10050 DOI 10.1007/s13369-020-04601-8.

Sundmaeker H, Guillemin P, Friess P, Woelfflé S. 2010. Vision and challenges for realising the Internet of Things. *Cluster of European Research Projects on the Internet of Things, European Commission* **3(3)**:34–36 DOI 10.2759/26127.

Tang R, Fong S. 2018. Clustering big IoT data by metaheuristic optimized mini-batch and parallel partition-based DGC in Hadoop. *Future Generation Computer Systems* **86**:1395–1412 DOI 10.1016/j.future.2018.03.006.

Truong C, Römer K, Chen K. 2012. Fuzzy-based sensor search in the web of things. In: *3rd IEEE International Conference on the Internet of Things*. Piscataway: IEEE, 127–134.

Tzeng G-H, Huang J-J. 2011. *Multiple attribute decision making: methods and applications.* Boca Raton: CRC Press.

Wang J, Cao Y, Li B, Kim H, Lee S. 2017. Particle swarm optimization based clustering algorithm with mobile sink for WSNs. *Future Generation Computer Systems* **76(12)**:452–457 DOI 10.1016/j.future.2016.08.004.

Yuqiang C, Jianlan G, Xuanzi H. 2010. The research of Internet of Things' supporting technologies which face the logistics industry. In: *International Conference on Computational Intelligence and Security*. 659–663.

Zannou A, Boulaalam A, Nfaoui EH. 2021. SIoT: a new strategy to improve the network lifetime with an efficient search process. *Future Internet* **13(1)**:4 DOI 10.3390/fi13010004.

Zhang Q, Chen Z, Leng Y. 2015. Distributed fuzzy c-means algorithms for big sensor data based on cloud computing. *International Journal of Sensor Networks* **18(1–2)**:32–39 DOI 10.1504/IJSNET.2015.069871.

Zhang D, Yang LT, Huang H. 2011. Searching in Internet of Things: vision and challenges. In: *2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications*. Piscataway: IEEE, 201–206.

Zheng Z, Tao Y, Chen Y, Zhu F, Chen D. 2019. An efficient preference-based sensor selection method in Internet of Things. *IEEE Access* **7**:168536–168547 DOI 10.1109/ACCESS.2019.2953045.