Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review, and Future Directions

Santosh Pattar, *Student Member, IEEE*, Rajkumar Buyya, *Fellow, IEEE*, K. R. Venugopal, *Fellow, IEEE*, S. S. Iyengar, *Fellow, IEEE*, and L. M. Patnaik, *Fellow, IEEE*

***Abstract*—Internet of Things (IoT) paradigm links physical objects in the real world to cyber world and enables the creation of smart environments and applications. A physical object is the fundamental building block of the IoT, known as a *Smart Device*, that can monitor the environment. These devices can communicate with each other and have data processing abilities. When deployed, smart devices collect real-time data and publish the gathered data on the Web. The functionality of smart devices can be abstracted as a *service* and an IoT application can be built by combining the smart devices with these services that help to address challenges of day-to-day activities. The IoT comprises billions of these intelligent communicating devices that generate enormous amount of data, and hence performing analysis on this data is a significant task. Using search techniques, the size and extent of data can be reduced and limited, so that an application can choose just the most important and valuable data items as per its necessities. It is, however, a tedious task to effectively seek and select a proper device and/or its data among a large number of available devices for a specific application. Search techniques are fundamental to IoT and poses various challenges like a large number of devices, dynamic availability, restrictions on resource utilization, real time data in various types and formats, past and historical monitoring. In the recent past, various methods and techniques have been developed by the research community to address these issues. In this paper, we present a review of the state-of-the-art search methods for the IoT, classifying them according to their design principle and search approaches as: IoT data and IoT object-based techniques. Under each classification, we describe the method adopted, their advantages and disadvantages. Finally, we identify and discuss key challenges and future research directions that will allow the next generation search techniques to recognize and respond to user queries and satisfy the information needs of users.**

***Index Terms*—Internet of Things, ranking and indexing, resource discovery, search and selection, search challenges, search requirements, service discovery.**

I. Introduction

INTERNET of Things (IoT) is a paradigm that connects real-world objects to the Internet, allowing objects to collect, process and communicate data without human intervention. The IoT's vision is to create a better world for humans, where objects (refers to physical objects, the terms object, device, entity, and things are used interchangeably) around us can comprehend our preferences and likeness to act appropriately without explicit instructions [1], [2]. The rapid advancements in low-cost sensor manufacturing, communication protocols, embedded systems, actuators and hardware miniaturization have contributed to the exponential growth of the IoT. Physical objects in the real-world are embedded with these technologies to make them smart. The functionality of the smart devices can be abstracted as a software service and an IoT application can be built by combining the smart devices with services that helps to address challenges of day-to-day activities. Atrozi *et al.* [3] have classified different applications that can be built with the help of IoT into three different domains, *i.e.,* Society, Industry, and Environment, as illustrated in the Figure 1.

As the society is moving towards IoT, the number of sensors deployed around the world is increasing at a rapid pace and these sensors continuously generate huge amount of data. However, not all of this data provide knowledge that helps in decision making process. Nonetheless, through device search functionality provided by an IoT application, the size and scope of data gathered can be reduced. Thus, search is an essential service, that enables to efficiently look for smart devices based on the real-world attributes gathered by the sensors. Perera *et al.* [1] and Barnaghi and Sheth [4] have emphasized the importance of search and discovery functionalities in IoT. However, due to large number of available sensors to choose from and resource limitations of an IoT application, designing such a search service is difficult.

Sensing as a service layer is envisioned to be built on top of the IoT infrastructure, where middleware solutions connect sensor devices to software systems and their related services [5]. Such sensors and services are made available to

Manuscript received July 25, 2017; revised January 27, 2018; accepted March 21, 2018. Date of publication April 10, 2018; date of current version August 21, 2018. *(Corresponding author: Santosh Pattar.)*

S. Pattar and K. R. Venugopal are with the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru 560-001, India (e-mail: santoshpattar01@gmail.com; venugopalkr@gmail.com).

R. Buyya is with the Cloud Computing and Distributed Systems Laboratory, School of Computing and Information Systems, University of Melbourne, Melbourne, VIC 3053, Australia (e-mail: rbuyya@unimelb.edu.au).

S. S. Iyengar is with the Department of Computer Science and Engineering, Florida International University, Miami, FL 33199 USA (e-mail: iyengar@csc.lsu.edu).

L. M. Patnaik is with the Consciousness Studies Program, National Institute of Advanced Studies, Indian Institute of Science, Bengaluru 560-012, India (e-mail: lalitblr@gmail.com).

Digital Object Identifier 10.1109/COMST.2018.2825231

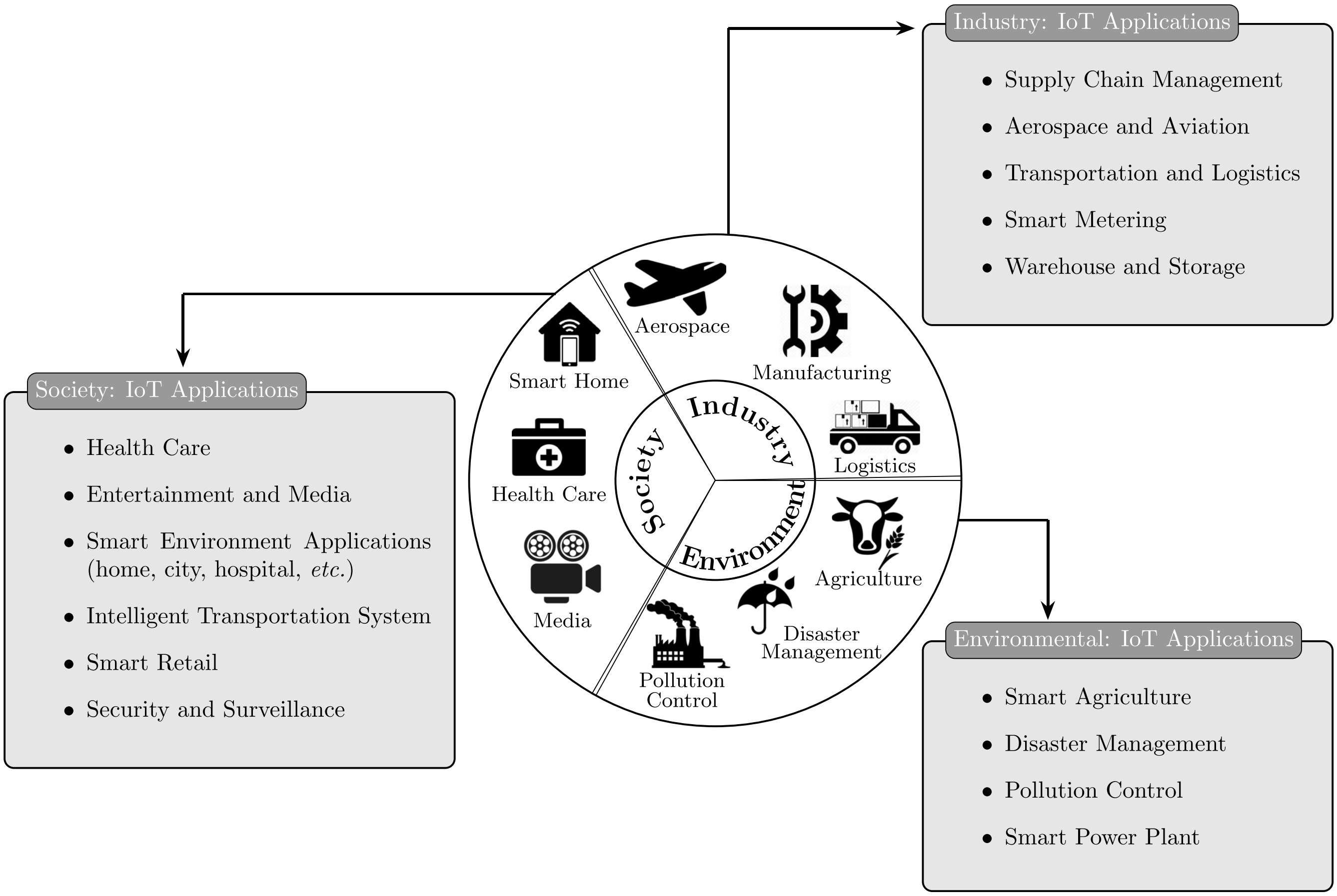


Fig. 1. IoT Applications under Different Domains.

potential users over the Internet either for free or by paying a fee through middleware solutions. The search facility for IoT is expected to be an integral part of this layer. Middlewares like Xively [6] and OpenIoT [7] provide Application Programming Interfaces (APIs) to connect, process and publish sensor data on the Web and thus inspire development of search and discovery service for IoT. At present, there are several search engines for IoT *viz.*, Shodan [8], Thingful [9], Censys [10], and Reposify [11] that offer search and discovery service for the IoT resources. However, their scope is limited to provide search for only those IoT resources that are subscribed or connected to their centralized servers.

In the past, several studies were conducted to review search techniques in IoT. Based on design space of different approaches, Romer *et al.* [12] presented a bird's eye view on search techniques in IoT. However, sensor measurement data has not been addressed in this survey. Zhang *et al.* [13] compared search techniques in IoT with other domains *viz.* ubiquitous computing, information retrieval and mobile computing. Architectural design, real-time, scalability, and locality of the search are identified as major research directions. However, this review does not address the entire gamut of the investigation. Search techniques in Web of Things (WoT) are reviewed by Zeng *et al.* [14] according to aggregation approaches (i.e., either pull-based or push-based). But, different facets of search techniques are not considered in the work. Zhou *et al.* [15] surveyed on search approaches in WoT domain, focusing on techniques applied, types of targeted search results and data representations. This survey does not detail on the challenges faced by the search systems. Table I summarizes the survey efforts of the above mentioned authors.

TABLE I
COMPARISON OF PAST SURVEYS

Authors	Year	Approach
Römer *et al.* [12]	2010	Based on design space.
Zhang *et al.* [13]	2011	Comparison of search techniques in IoT with its predecessors domains.
Zeng *et al.* [14]	2011	Based on data aggregation type.
Zhou *et al.* [15]	2016	Based on search principles.

Our survey is different from the above mentioned surveys, we have selected a large number of research works (106) and classified them into the taxonomy based on their search approaches and design principle. The advantages, disadvantages, and challenging issues of each classification and their applicability in different domain are summarized in this paper. The **contributions** of this paper are as follows:

- *Search and discovery fundamentals in IoT:* A comprehensive tutorial on search and discovery system for IoT along with the integral components, design strategies and search principles are presented.

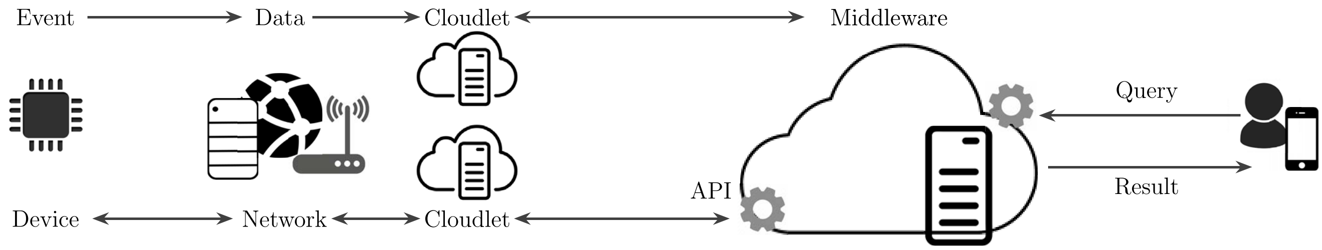


Fig. 2. Fundamental Components of the Search System for the IoT.

- *Identification of search requirements and challenges:* We delineate and categorize various requirements and challenges of search techniques across different application domains of the IoT.
- *Review and analysis of current trends in search and discovery approaches:* We have reviewed the existing literature on search algorithms for IoT extensively and presented an analysis of their potentials and limitations that are propitious for developers and researchers to get familiar with the current search techniques for IoT.
- *Future research guidelines on search techniques:* We have listed a few promising techniques as future research directions that address the research community to overcome the challenges designing an efficient search system for IoT.

The rest of the paper is organized as follows. Section II provides a brief outline of search and discovery fundamentals in IoT. A classification model has been presented in Section III. It also lists applicability of different search techniques in different application domain of IoT under various use case scenarios. In Section IV, a review of the state-of-the-art research that addresses the search and discovery problems in IoT has been discussed. They are categorized into two groups as: (i) IoT Data-based, and (ii) IoT Object-based techniques. These groups are further sub-categorized based on the design principles and solutions employed. Section V contains a comprehensive comparison of the reviewed publications, critical discussion on each category along with their advantages, disadvantages, and challenges. Section VI highlights the future research directions in search and discovery techniques for IoT. Finally, concluding remarks are presented in Section VII.

II. Search and Discovery Fundamentals

A. High-Level Overview of the Search Process in IoT

In this section we will attempt to present the fundamentals and principles of the search and discovery process in the IoT. A search function is utilized by both humans and machines alike in IoT. Figure 2 depicts the overall process of the search operation. Sensors are embedded into the IoT objects that collect real-time data about the surrounding environment. In the real-world, they detect events and then generate data about the detected events. These objects are networked at various levels (*e.g.,* local or global), and thus a middleware is used to manage them. The objects register with the middleware through a subscription process, the APIs are provided by it for application development and to perform management operations. Users/Machines submit their query to middleware either by using an interface provided by the application or through the API. The search algorithm is then initiated by the middleware and results are returned back to the query requester.

Algorithm 1 Query Resolution Process in the IoT

1: *Query capturing:* Collection of query from the user/machine.
2: *Query Analysis:* Preprocess the query by subjecting to transformation, filtering or normalization to identify its intent.
3: *Query Matching:* Retrieve matching IoT resources from the database, ontology or network through the middleware.
4: *Query Result Analysis:* Index and Rank the matching IoT resources based on a scoring method.
5: *Result Generation:* Select appropriate number of the IoT resources from the ranked list and return it as result.

Once a query has been received by the search system, it can be processed (through techniques like transformation, filtering, normalization, *etc.,*) and divided into sub-queries. The system then contacts the nodes (cloudlets) in IoT network to retrieve a list of matching resources, that is further indexed and ranked based on some scoring method employed by the search system. Based on query request, the required number of matching resources are selected at the final stage to resolve the query. These steps (Query Capturing, Query Analysis, Query Matching, Query Result Analysis, and Result Generation) are outlined in Algorithm 1.

B. Fundamental Components of the Search System

A search system in the IoT is designed to look for a specific set of *IoT Resources* that match with the requested query. These IoT resources are composed either of the *IoT Objects* and *IoT Data* or a combination of both. Figure 2 illustrates the essential components of an IoT search system, they are described as follows:

1) IoT Objects: In IoT, the real-world objects are smart devices that are embedded with enabling technologies (like sensors, actuators, storage-device, processors, communicating-devices, *etc.,*) to monitor the environment, gather data, process it, communicate with other IoT objects, and take some action. Search systems try to discover these IoT objects based on their state information and/or data generated by them. Some search systems may differentiate between the IoT objects and its enabling technologies embedded in them, while others may not. Section IV reviews the former search systems, while the latter search systems are described in the Section IV-B3b.

2) IoT Data: The data generated by the IoT objects can be classified into two types, based on its relationship to the data source (*i.e.,* IoT object). (i) Observation and Measurement data, that is, data generated by sensors stimulus to an event in real-world (*e.g.,* temperature sensor generates temperature readings of the surrounding environment in some unit, say Celsius), and (ii) Context-data, that gives description about the IoT object's state and its operating condition (for example, battery-life, availability, error-rate, latency, *etc.,*). Section IV-A1 reviews the search systems that consider observation and measurement data in their search parameters, as content-based search techniques, while search systems that consider context-data are discussed in Section IV-A2.

3) Search Space: A search space is a set of the IoT resources, their related properties and data items over which a search algorithm locates matching resources based on some requirements [16]. Although, search space in the IoT is characterized by a large number of resources and the huge size of their data, it can be structured based on the social relationships of the IoT resources with each other to reduce the size of search space; such search techniques are discussed in Section IV-B2a.

4) Query: A search query is a question asked by user of the search system to get some information; in IoT, a search query can be submitted by either an human user or an IoT object itself. Search requirements are specified in a query as parameters and matched with items in the search space.

5) Middleware: An IoT middleware acts as an interface between the user/application and the IoT network. It establishes connections between heterogeneous IoT resources and thus offers a single platform on which the search systems can be built. The APIs are provided by it that eases application development process. Middlewares are implemented on clouds to support scalability, heterogeneity and interoperability [17].

6) Cloudlets: The cloudlets deliver the computing facilities of the cloud to the edge of network, *i.e.,* closer to the devices. The IoT objects can thus rely on the cloudlets to perform computationally demanding tasks [18]. In a search system, these cloudlets are used to crawl and maintain an updated index of the search space.

C. Classification of Search Techniques

The devices and their services can be searched through various techniques [19]. In this subsection, different search principles are presented and classified according to their approach.

1) Functional Viewpoint: Based on the applicability and usage in different scenarios, search systems are classified as follows:

(i) *Event-based Searching:* The IoT resources are located based on certain real-world events. For example, searching an IoT object that senses temperature.

(ii) *Location-based Searching:* The IoT resources are located based on geographical/relative location of the devices. For example, searching for free parking spots.

(iii) *Time-related Searching*: The IoT resources are located based on the data generated at particular time or period. An example of such an approach is, looking for device that is active from last two hours.

(iv) *Content-based Searching:* The IoT resources are located based on observation and measurement data generated. For example, finding air pollution monitoring sensors that has recorded carbon monoxide level higher than the given threshold.

(v) *Spatiotemporal-based Searching:* The IoT resources are located based on their location, events that they monitor and the time at which the event was generated. For example, locating a free parking spot at a particular area at a given instance of time.

(vi) *Context-based Searching:* The IoT resources are located based on their status and operational parameters. For example, locating sensors that have maximum battery life.

(vii) *Real-Time Search System:* In time-critical applications real-time results are provided by the search system without any latency.

(viii) *User Interactive Searching:* A user interface is provided that allows the user to manually select devices, from which they want to collect the data. For example, a map-based graphical user interface is provided through which the user manually selects the devices of his choice.

2) Implementation Viewpoint: Search systems are implemented on various technologies and approaches, that are based on following:

(i) *Text-based Approach:* The textual descriptions of IoT resources are maintained at centralized/distributed servers and keywords-based queries are used to perform search operations.

(ii) *Metadata-based Approach:* The IoT resources are annotated with metadata that describe operational and status information about them. They can either be stored locally at the IoT object itself or at middleware. The annotation process can either be manual or automatic.

(iii) *Ontology-based Approach:* The IoT resources, their properties and characters are described through well formed rules (i.e., ontology). Semantics of resources are well modeled in this approach; range and description-based queries are supported.

D. Design Strategies

In this subsection, we present fundamental strategies that are to be considered when developing and implementing a search system for the IoT.

1) Architecture: A search system in IoT is designed and implemented through two different architectural styles.

(i) *Distributed Approach:* The Middlewares that store indexes, data items of the IoT resources are distributed geographically and the search algorithm is run locally on them. Results are then aggregated at the global level

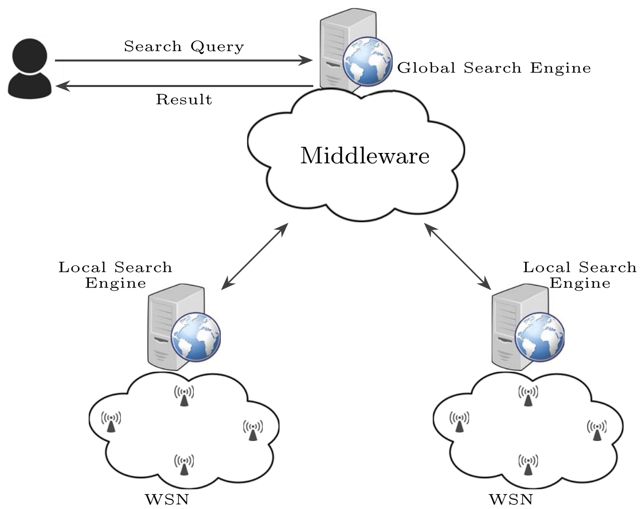


Fig. 3. Architecture of Distributed Search Engine for the IoT.

by combining local search results. Figure 3 illustrates this approach.

(ii) *Centralized Approach:* A central middleware/server is responsible for the search system as opposed to distributed approach.

2) Data Acquisition Methods: The data from IoT resources can be gathered by middleware in the following different ways:

(i) *Publish/Subscribe:* In the IoT network, resources are loosely coupled with each other and the middleware. An explicit relationship between an IoT resource and middleware is established in this method, where the IoT resource acts as data publisher and middleware as data subscriber.

(ii) *Request/Response:* In this method, IoT resources and middleware are tightly coupled with each other. A resource request for some kind of service to the middleware. and the middleware responds to the request by providing the service.

3) Search Space Structuring: Characteristics of the search space in an IoT network determines the type of search algorithm employed. It can be structured to effectively retrieve the query matching resources. Following design strategies are used to construct the search space:

(i) *Indexing:* The data collected from the search space in the IoT network is stored and indexed at middleware for fast and efficient look-up. Indexers in the IoT domain can utilize features like context and content data.

(ii) *Crawling:* The updated information about search space at middleware is maintained by the crawler. It visits every object in the IoT network and fetches its data and gives it to the indexer.

(iii) *Scoring and Ranking:* The relevance of a resources matching a given query is determined by the scoring and ranking algorithm employed. Quantitative scores are given to matching IoT resources based on their fitness to context of query and then sorted to retrieve the top most results.

4) Prediction and Recommendation Models: Due to dynamic and large size of the IoT network, where the IoT resources are computationally constrained, frequent communication between them and the search system can be reduced by constructing appropriate prediction and recommendation models.

E. Requirements and Challenges

The search and discovery techniques for the IoT are faced with numerous challenges that reduce their performance quality. They need to support certain requirements to enhance their applicability and usability across different IoT application domains. We shall discuss some of the requirements and challenges for the search and discovery techniques here.

1) Search Requirements: An IoT application developed for a specific domain performs tasks related to that domain and thus has a definite purpose which is modeled based on the requirements arising through the objectives of the domain. Search techniques for the IoT applications should address these objectives by incorporating the requirements and provide enhanced solutions. We categorizes the search requirements into three groups as: (i) User-oriented, (ii) Solution-oriented, and (iii) Quality of Service (QoS) based, according to their significance across various components of the search system. The User-oriented requirements define the functionalities to be provided by the search system to enhance their usability. We identify four such requirements as: Personalization (R_1), Multifaceted Query Results (R_2), Search Intent Identification (R_3), and Search Engine Experience (R_4). The definitions of these requirements are as follows:

R_1) *Personalization:* Search systems must automatically tune the search results according to the user's preferences.

R_2) *Multifaceted Query Results:* Search systems must categorize the search results into configurable groups based on the search query.

R_3) *Search Intent Identification:* Search systems must identify the intent of the search query through the search requirements.

R_4) *Search Engine Experience:* User interface of the search engine should be simple and search results must provide enhanced information (*e.g.,* latest updates, device status, events generated.)

Solution-oriented requirements capture the essential features of the search technique to be employed by the search system. We list five such requirements as: Real-Time (R_5), Low Latency (R_6), Filtering (R_7), Energy Efficient (R_8), and Resistant to Noisy Data (R_9). The definitions of these requirements are as follows:

R_5) *Real-Time:* Search system should extract the IoT object's data in real time.

R_6) *Low Latency:* Waiting time between different stages of the search algorithm should be at minimum.

R_7) *Filtering:* Search techniques should support filtering of candidate matching devices through the data, context and other properties.

R_8) *Energy Efficient:* Search technique should account for the amount of energy consumption in the IoT resources.

TABLE II
LIST OF REQUIREMENTS

Category	Notation	Criteria	Definition
User-oriented	R_1	Personalization	Search results are to be fine tunned to the user based on his preferences.
	R_2	Multifaceted Query Results	Results should be categorized into configurable groups that are based on the search query.
	R_3	Search Intent Identification	Search technique should identify the intent of query through search requirements.
	R_4	Search Engine Experience	User interface of the search engine should be simple and search results must provide enhanced information (*e.g.* latest updates, device status, events generated).
Solution-oriented	R_5	Real-Time	It must be possible to extract required information at real time.
	R_6	Low Latency	Waiting time between different stages of the search algorithm should be at minimum.
	R_7	Filtering	Search technique should support filtering of candidate matching devices through the data, context and other properties.
	R_8	Energy Efficient	Search technique should account for the amount of energy consumption in IoT object.
	R_9	Resistant to Noisy Data	Noisy data hinders fast query execution and thus search algorithm should be able to filter out them.
Quality of Service based	R_{10}	Accurate Results	Query results must match to exact solutions.
	R_{11}	Security Guarantee	Search technique must account for the security level as desired by the user and IoT device.
	R_{12}	Privacy Protection	Only trusted results should be listed and sensitive information of the IoT device must be hidden.

R_9) *Resistant to Noisy Data:* Noisy data hinders fast query execution and thus search algorithm should be able to filter them out.

Meeting the QoS-based requirements elevates the performance and applicability of the search system, we consider three most important requirements as: Accurate Results (R_{10}), Security Guarantee (R_{11}), and Privacy Protection (R_{12}). The definitions of these requirements are as follows:

R_{10}) *Accurate Results:* Search results must match to the exact requirements as specified in the search query.

R_{11}) *Security Guarantee:* Search technique must account for the security level as desired by the user and the IoT device.

R_{12}) *Privacy Protection:* Only trusted results should be listed and sensitive information of the IoT resources must be hidden.

Table II lists these requirements along with their notations and definitions.

2) Search Challenges: The IoT network is inherently dynamic and its huge size not only generates data at high velocity but also consist of heterogeneous types. These and other challenges impede the search system from better performance which urge the search and discovery solutions to incorporate methods and approaches to overcome them. We list few challenges faced by a search system in IoT domain in Table III along with their notations and definitions. These challenges are grouped into three classes according to the components of the search system that they effect the most as: (i) IoT object oriented, (ii) Search and Discovery Solution-oriented, and (iii) QoS-based challenges. Devices and sensors in the IoT network presents diverse challenges due to their dynamic characteristic features like frequent change in their data properties, position and connectivity. We associate the following four challenges with the IoT resources: Dynamicity (C_1), Scalability (C_2), Mobility (C_3), and Opportunistic Presence (C_4). The definitions of these challenges are as follows:

C_1) *Dynamicity:* Search system should consider constant change in the network topology, sensing data and properties of the IoT resources.

C_2) *Scalability:* Search system should manage large magnitude of the sensors and devices connected to the IoT.

C_3) *Mobility:* Search system must handle frequent change in the location of the IoT resources.

C_4) *Opportunistic Presence:* Search system should take into account the dynamic connection status of the IoT object with the network.

Similarly the search techniques themselves face the challenges with respect to the IoT application domain (*i.e.,* generality (C_5) and specificity (C_6)), and design strategy (*i.e.,* data acquisition (C_7), management (C_8), heterogeneity (C_9), and interoperability (C_{10})). The definitions of these challenges are as follows:

C_5) *Generality:* Search system should not bind to a specific approach, technique or protocol.

C_6) *Specificity:* Search system should perform definitive operation based on the application domain.

C_7) *Data Acquisition:* Search system should retrieve data from the IoT objects, users and applications with minimum communication and in real-time.

C_8) *Management:* Search system must be able to configure, control and manage the IoT devices, middlewares, clouds and applications.

TABLE III
LIST OF CHALLENGES

Category	Notation	Criteria	Definition
IoT object oriented	C_1	Dynamicity	Constant change in network topology, sensing data and properties of IoT objects.
	C_2	Scalability	Large magnitude of sensors and devices connected to the IoT.
	C_3	Mobility	Frequent change in location of the IoT objects.
	C_4	Opportunistic Presence	Dynamic connection status of the IoT object with the network.
Solution-oriented	C_5	Generality	Not to bind to a specific approach, technique or protocol.
	C_6	Specificity	Definitive operation based on application domain.
	C_7	Data Acquisition	Retrieval of data from IoT objects, users and applications.
	C_8	Management	Configure, control and manage IoT devices, middlewares, clouds and applications.
	C_9	Heterogeneity	Diversity in data format, standards, communication protocols and applications.
	C_{10}	Interoperability	Operate among heterogeneous data formats, standards and platforms.
Quality of Service based	C_{11}	Security	Establish secure communication channel and be resistant to attacks.
	C_{12}	Privacy	Gather data from trusted data sources only and not to expose sensitive information of the device.
	C_{13}	Standardization	Well defined and accepted protocols.

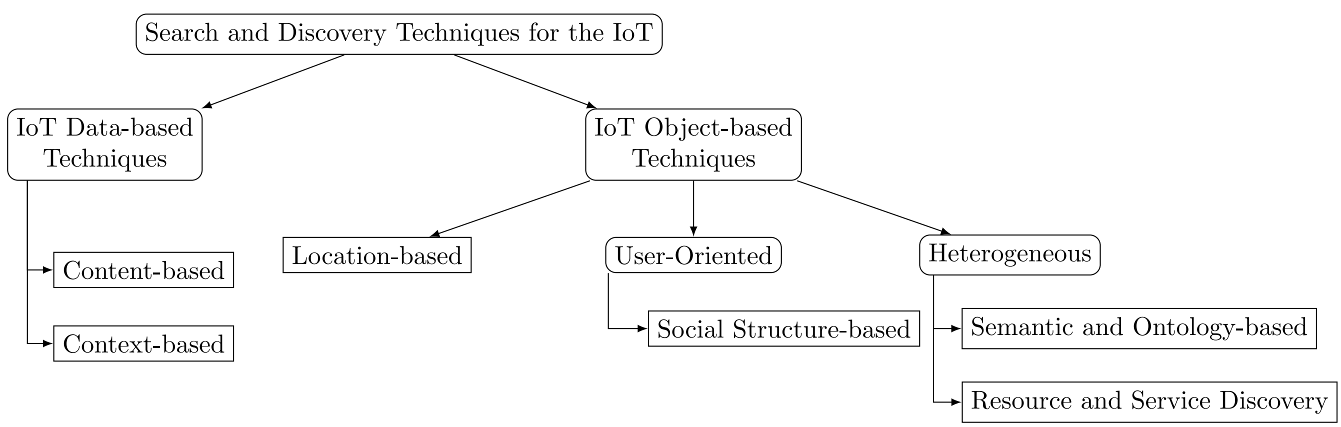


Fig. 4. Classification of Different Search and Discovery Techniques for the IoT.

C_9) *Heterogeneity:* Search system should handle diversity in the data format, standards, communication protocols and applications.

C_{10}) *Interoperability:* Search system should operate among the heterogeneous data formats, standards and platforms.

The QoS-based challenges are subservient to the end user of the search system, we identify three such challenges as: Security (C_{11}), Privacy (C_{12}), and Standardization (C_{13}). The definitions of these challenges are as follows:

C_{11}) *Security:* Search system should establish secure communication channel and be resistant to the attacks.

C_{12}) *Privacy:* Search system must gather data from the trusted data sources only and not expose sensitive information of the IoT devices.

C_{13}) *Standardization:* Search system should be implemented on well defined and accepted protocols.

Role played by these requirements and challenges are examined in the next section with a use cases scenario across different IoT application domains.

III. SEARCH AND DISCOVERY IN IoT: CLASSIFICATION AND USE CASE SCENARIO

In this section, we present our outlook on different search and discovery techniques for IoT and classify them into two main categories: (i) IoT Data-based, and (ii) IoT Object-based search techniques, according to the type of IoT resources that the search techniques address. Further, they are categorized based on their approaches, algorithmic design and search principles into various subcategories. Figure 4 represents our classification system. Classification viewpoint is discussed in Section III-A and the applicability of each search technique with a use case scenario are presented in Section III-B.

A. Classification Viewpoint

1) IoT Data-Based Search Techniques: The data originating from devices connected to the IoT network can be classified into one of the following three types as depicted in Figure 5: (i) Inherent (static) data, that do not change (*e.g.,* type of sensor, manufacture details, RFID tag), (ii) Dynamic data, that

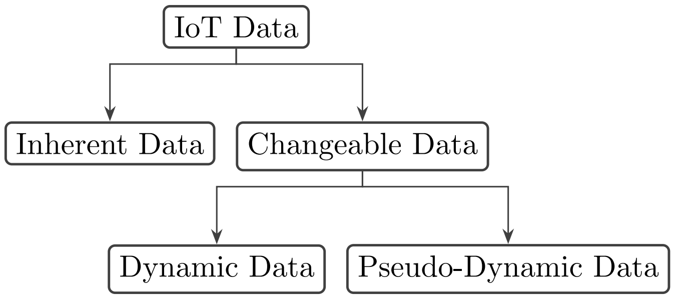


Fig. 5. Data classification of the IoT objects.

changes continuously over time (*e.g.,* data captured by sensors), and (iii) Pseudo-dynamic data, that changes slowly over time (*e.g.,* state of the objects, deployment properties).

This data has various characteristic features such as dynamicity, heterogeneity, huge size, dynamic data generation rate, volatility, *etc*. Due to these features its a challenging task to gather, store, and manage the IoT data. Several protocols and standards are being designed and developed by various organizations to meet these challenges. Some of them are Message Queueing Telemetry Transport (MQTT), Data Distribution Service (DDS) [20], Extensible Messaging and Presence Protocol (XMPP) [21], Advanced Message Queuing Protocol (AMQP) [22], and Constrained Application Protocol (CoAP) [23]. Depending on the application requirements different combination of these protocols can be used for data addressing. We classify the IoT Data-based search techniques into the following types:

- *Content-based searching:* It focuses on readings taken by the sensors, *i.e.,* observation and measurement data.
- *Context-based searching:* It deals with situational and status description data of the IoT objects.

2) IoT Object-Based Search Techniques: These search techniques focus on locating the IoT objects. Various approaches have been employed in this category that make use of object's features like semantic description, location and social relationship. We categorize them further into following subcategories:

a) Location-based searching: It emphasizes on geographical/relative location of the IoT objects in the network.

b) User-oriented search techniques: These techniques correlate user preferences with the IoT object's properties.

- *Social Structure-based searching* It considers social relationships between the IoT objects and users.

c) Heterogeneous search techniques: Heterogeneous search techniques make use of several different types of data representation, status description and other attributes of IoT objects to efficiently resolve a search query. We consider the following two search techniques under Heterogeneous category:

- *Semantic and Ontology-based searching:* It focuses on knowledge representation of the IoT objects and their relationships in terms of well defined rules.
- *Resource and Service Discovery:* It looks for the IoT objects (object is an ensemble of sensors, actuator and other IoT enabling technologies) and their embedded software services.

B. Applicability and Use Case Scenarios

In this subsection, we discuss the applicability of different search and discovery techniques across various disciplines in IoT applications. The use case scenarios and their relevant search techniques as well as requirements and challenges for them are listed in Table IV.

1) IoT Data-Based Search Techniques: Based on the role played by the IoT data in the search system, three use case scenarios from three different domains to highlight the approach, requirements and challenges of the search techniques are discussed.

- *Content-based Search Technique:* In Aerospace and Aviation application domain of IoT, a search use case can be to locate the present altitude of an air borne vehicle. The altitude data of this vehicle is collected through an altimeter sensor attached to the vehicle, for example, Laser Vegetation Imaging Sensor (LVIS) is an air borne altimeter sensor used to perform topography, hydrology, and vegetation studies [24]. However, this data is dynamic and changes frequently, as such during take-off and landing. Requirements for such a search could be to obtain accurate data in real time (R_{10} and R_5), search results may further be subjected to filtering based on different altitudes (R_7) where a range of altitudes can be displayed in different facets (R_2), however these results can be viewed only by authorized personnels (R_{11}). Challenges faced in this search scenario is to acquire frequently changing data (C_1 and C_7) that is specific to Aerospace and Aviation domain (C_6); the search algorithm here has to operate among different data communication protocols (C_{12}).
- *Context-based Search Technique:* One of the most important task to be performed in the agriculture field is detection of pests. Early detection and control of pests can prevent harmful disease from infecting the crops. This scenario requires deployment of sensors that monitor environmental phenomenons to identify the presence of pests, the number of such sensors utilized is enormous due to the size of agricultural field and thus possess scalability challenge (C_2). As such it requires the search algorithm to be energy efficient (R_8) and impervious to the noise (R_9) gathered from the data source. Search system should also be able to identify the intent of the pest detection test (R_3), i.e., to control and activate the sprinkler (C_8).

2) IoT Object-Based Search Techniques:

i) *Location-based Search Technique:* The search and discovery techniques in smart home domain should have effective user interface design (R_4) that provides updated status information about home appliances. An example of a search query that can come forth in such an application domain is to find a misplaced book that was used at some particular instance of the day [25]. The search system on this occasion must be capable of filtering matching candidate items based on context (i.e., time) and location of last usage (R_7). Solution to this search problem must not bind to a particular method

TABLE IV
APPLICABILITY OF DIFFERENT SEARCH TECHNIQUES ACROSS DIFFERENT IoT APPLICATIONS

Category	Search Technique	Applicability	IoT Applications	Use Case Scenario	Search Requirements	Search Challenges
IoT Data based	Content based	For applications that requires to find real-time and historical IoT data.	Aerospace & Aviation	Search for high-velocity and frequently changing altitude data of an air borne vehicle.	R_2, R_5, R_7, R_{10}, & R_{11}	C_1, C_6, C_7, & C_{12}
	Context based	For applications that requires to select, control and activate an IoT object based on certain situational, environmental and other context requirements.	Smart Agriculture	Test whether an agricultural field has been invaded by a pest or not, if so initiate the sprinkling of pesticides.	R_3, R_8, & R_9	C_2, & C_8
IoT Object based	Location based	For applications that need to locate an IoT object at a particular location either geographically or relative to some other object.	Smart Home	Search for a misplaced book in the study room.	R_4, R_7, & R_8	C_4, C_5, & C_7
	Social Structure based	For applications that provides recommendations to user based on their preferences, profile and social links with other users and IoT objects.	Intelligent Transportation System	Find an optimal route in highly congested traffic.	R_5, R_6, & R_{10}	C_4, C_5, & C_{11}
	Semantic and Ontology based	For applications that handle domain specific search requests and where diverse type of IoT data exists, ontologies can be used to unify them on a single platform.	Smart Retail	Find a perfect pair of Jeans.	R_1, R_2, R_7 & R_{12}	C_3, C_6, C_9, & C_{13}
	Resource and Service Discovery	For applications that address entity search, where IoT objects are embedded with sensors, actuators, and their functionality are abstracted as services.	Entertainment & Media	Locate for a candy vending machine in an amusement park.	R_1, R_4, R_5, & R_{12}	C_3, & C_6

Requirements - R_1: Personalization, R_2: Multifaceted Query Results, R_3: Search Intent Identification, R_4: Search Engine Experience, R_5: Real-Time, R_6: Low Latency, R_8: Energy Efficient, R_9: Resistant to Noisy Data, R_{10}: Accurate Results, R_{11}: Security, and R_{12}: Privacy Protection.
Challenges - C_1: Dynamicity, C_2: Scalability, C_3: Mobility, C_4: Opportunistic Presence, C_5: Generality, C_6: Specificity, C_7: Data Acquisition, C_8: Management, C_9: Heterogeneity, C_{10}: Interoperability, C_{11}: Security, C_{12}: Privacy, and C_{13}: Standardization.

(C_5) of data retrieval (C_7) as the device will be in dormant state to effectively utilize its energy level (R_8) and thus presents the detection challenge (C_4).

ii) *User-oriented Search Techniques:* To emphasise the properties of user (like social links, preferences, *etc.*) that are to be considered to meet the search requirements, we describe a use case scenario related to Intelligent Transportation Systems (ITS).

- *Social Structure-based Search Technique:* The ITS is one of the most thriving application domain of IoT. In such an application, the user in heavily congested traffic is assisted by his cars navigation system to find an optimal route, by contacting near by cameras, traffic management system and his peers in the same location to reach the destination in stipulated time [26]. In this circumstance, the search system should retrieve information on volume of traffic on road network of the commuting area in real time (R_5) and the latency between subsequent queries about the route information should be kept at minimum (R_6). However, the most critical requirement is that of the accuracy of query result (R_{10}). To design such a search system, the connectivity of vehicle with the ITS infrastructure is of major concern (C_4) and also the myriad of communication standards used by the ITS system (C_5 and C_{11}) pose a challenge for effective design of a search solution.

iii) *Heterogeneous Search Techniques:* To demonstrate the suitability of ontology based solutions, we have described a retail store use case and similarly an amusement park based scenario is construed for service discovery problem.

- *Semantic and Ontology-based Search Technique:* In a smart retail based application of an IoT, the user would like to locate a specific pair of Jeans based on his preferences (R_1 and R_2) and the search system should respond accordingly by filtering the results (R_7). A major challenge is to keep track of the mobility of the user (C_3) along with his specific requirements (C_6) and maintain his privacy (C_{13}) [27].

- *Resource and Service Discovery:* An amusement park has several outlets that offer different kinds of recreational activities. During a stay in an amusement park, an user would like to locate a nearby candy vending machine that suits his preferences (R_1). The search system for such a scenario should operate on the specific domain (C_6) and must consider dynamic location of the user in park (C_3). The search results should also intimate the most recent updates about the vending machine like, availability of a particular candy, price and offers (R_4 and R_5). To enhance the query results, the search system can display the most popular and highly rated vending machines (R_{12}).

IV. A Review of Research Efforts

In this section, we review the state-of-the art search and discovery techniques based on the classification system proposed in the previous section. For each one of the research publications, we describe the concept and highlight its significant contributions and shortcomings.

A. IoT Data-Based Search Techniques

1) Content-Based Search Techniques: In this subsection, we review search techniques that take into account measurement data of the IoT objects. The content-based search techniques make use of sensor output to search for a particular set of sensors that match the user queries. They look for a specific data item produced by sensors based on the predefined user requirements. These approaches can be applied to obtain real-time and historical data. Applications that address user queries like room occupancy at any given instance, or finding an available parking spot, can be modeled by content-based approaches. Hamann [28] highlighted the use of statistical and prediction models in analyzing sensor data, and some of the research works that make use of such analytic techniques are discussed below.

Elahi *et al.* [29] have constructed a predictive model that ranks sensors based on their current output state with respect to the user queries. The probability scores are calculated based on the output of sensor at any given particular time. The advantage is that, if queries are executed more often, then there is a substantial improvement in prediction rates. However, the model is susceptible to faulty sensors, and prediction model suffers from latency, to incorporate the changes introduced by it. Ostermaier *et al.* [30] proposed a search engine called *Dyser* where statistical methods are used to recognize the sensors that match the requested query. Using prediction model, the matching sensors are ranked according to their similarity score in descending order, to retrieve the top-*k* matches. *Dyser* reduces the overhead communication by predicting periodic patterns, but this finding is based on simulated data and thus cannot be generalized.

Mietz and Römer [31] used Bayesian Networks to exploit correlation among sensors that produce similar outputs. It performs optimally when the query seeks only a small number of sensor results. However, as the data collection and processing is centralized; the model is not scalable for a large number of sensors. Truong *et al.* [32], [33] have employed search-by-example technique, where a sensor similarity search is employed through a fuzzy logic approach. A fuzzy based similarity score selects a matching sensor that outputs the same value as that of a given sensor. The similarity score here is computed incrementally that increases accuracy of the prediction model with reduced computation overhead. Although, the search results are presented quickly to the user because of approximate and incremental computation, the users who need highly accurate query results need to wait for longer durations.

Zhou *et al.* [34] utilized spatial properties of sensors that are embedded into sensor measurement data and are stored in time-series database on a cloud. The system supports search for historical and real-time data. The challenge however, is to collect data from mobile and static entities that are large in number. The Matching State Estimation (MSE) technique is proposed in [35] for device discovery in WoT domain. Two types of search modes, reliable MSE search (R-MSE) and proactive MSE search (P-MSE), are implemented to satisfy the search needs, that present strong randomness and uncertainty with respect to query time and number of returned search results. The advantage of these systems is that communication overhead decreases with increase in the query range. However, only prototype design was presented and practical applicability of MSE in the device search service is low.

Bijarbooneh *et al.* [36] formulated a sensor selection problem and solved it using greedy strategy and constraint programming approach. Belief propagation protocol with multiphase adaptive sensing is used to infer sensor output. The scheme is able to reduce energy consumption by 80% through belief strategy that turns on only a small number of required sensors, as compared to a situation when all sensors are used. The system does not support distributed environment. Zhang *et al.* [37] have proposed a system to accurately estimate the present sensor state based on a multi-step prediction model. A sensor ranking model is developed that reduces communication overhead of the entire process by taking into account the matching probabilities of sensors. It estimates the reference value of sensor output at the query time and achieves high-precision in matching the probability prediction, so as to reduce the storage and energy cost. However, while reducing the communication overhead, the system leads to increasingly high computational complexity.

Ihm *et al.* [38] have used top-*k* queries to address data discovery in IoT applications. Data gathered by IoT objects are partitioned through grid and hyperplane approach to reduce the search space. Indexes are then built for the partitioned data space and matched with the query to obtain top-*k* results. Support Vector Machine (SVM) [39] was employed by Jiang *et al.* [40] to predict the dynamic measurement error of the sensor device. Cuckoo Search (CS) algorithm is developed to optimize the key parameters and avoid local minimum values, that occurs when using traditional method of parameter optimization. However, the performance of SVM depends on setting appropriate parameters that are situation dependent.

TABLE V
COMPARISON OF DIFFERENT CONTENT-BASED SEARCH TECHNIQUES

Authors	Year	Concepts	Advantages	Disadvantages
Truong *et al.* [32] [33]	2013	Light-weight prediction model to select a matching sensor that outputs same value as that of a given sensor.	Noisy data obtained from low cost sensors are addressed.	User who needs highly accurate query results need to wait for longer durations.
Zhang *et al.* [35]	2015	Matching state estimation scheme for device search in WoT domain (MSE).	Ordered verification approach efficiently checks sensors after ranking sought sensors on the top.	Applicability of MSE is rare in practical sensor search service.
Zhang *et al.* [37]	2016	Low-Overhead and High-Precision Prediction Model (LHPM) to assess matching probabilities of sensors.	LHPM saves more storage and computing resources and effectively reduce the communication overhead of the search process.	Computing time of LHPM increases sharply with the degree of polynomial of the approximation method.
Jiang *et al.* [40]	2016	Cuckoo Search-Support Vector Machine (CS-SVM) algorithm to optimize parameters and avoid local minimum value.	System accurately predicts sensor state and is more effective in predicting dynamic measurement errors of sensor output.	Probability parameter in CS-SVM is fixed, which can affect convergence of the algorithm.
Zhang *et al.* [41]	2016	Periodical estimation of sensor output through Content-based Sensor Search with a Matching Estimation Mechanism (CSME).	High accuracy and reduced communication overhead of sensor search system.	Search performance depends on resource consumptions and proposed system is not fault-tolerant.

A time-dependent periodical prediction method is presented by Zhang *et al.* [41] to estimate the sensor output. The periodic feedback prediction fully exploits verification information for enhancing the precision of sensor readings that efficiently serve the needs of sensor search service. Although, efficiency and applicability of sensor search systems are improved by accurate matching estimation mechanism, the search performance depends on resource consumptions; and moreover the system is not fault-tolerant. Anas *et al.* [42] have developed a heuristic framework to capture the IoT object data. Once data collected is from sensors, it undergoes thorough transformation and filtering procedure and is finally fed to a genetic algorithm for searching. However, the system does not consider heterogeneous data collection and management, that is a characteristic feature of data in IoT. Shemshadi *et al.* [43], [44] developed a crawler to collect IoT data automatically from different data sources. The system provides interface for both human users and machines. However, only those data sources were chosen where the sensor data is represented through a map.

Vasilev *et al.* [45] proposed a scalable model based on hyper-graph representation for the evaluation of the cooperation between the sensor nodes for a reliable node discovery. The advantage of the model is that it allows the metric to be computed in parallel between sub-networks. The metric can be computed independently by each sub hyper-tree and efficiently combined at the tree intersections. In addition, if the connected component can be built by keeping its structure to be a hyper-tree, the metric can be computed in a linear time. The model does not perform well in the presence of higher-order dependencies between the metric and the packet acceptance probability.

Table V compares the top-five most recent content based search approaches, along with their advantages and disadvantages. As measurement data is quantitative and dynamic over time, the searching devices in IoT domain based on the content of device measurement is challenging in nature. With a large scale deployment of the low-cost sensors, it is not feasible to monitor the output of each sensor. Thus, most of the current search systems fail to support such an approach and it is the least employed technique by the research community. Web document management and retrieval techniques can be integrated with content-based approach to increase the efficiency of these systems [46].

2) Context-Based Search Techniques: Context-based search approaches allow user to express search parameters in terms of sensor properties that describe the situation and quality of service of the sensor, like location, type, accuracy, battery life, *etc*. Some existing tools provide IoT middleware solutions for sensor search, and through these middlewares, the user submits a query, and search system returns appropriate sensor data to the user without any further interactions [47].

Linked Sensor Middleware (LSM) [48] is one such middleware that provides sensor search and selection service based on sensor location and its type. However, the users have to specify search queries in SPARQL Protocol and RDF Query Language (SPARQL) [49] that is unfriendly to novice users. The Global Sensor Networks (GSN) [50] is a flexible middleware that facilitates distributive query processing and sensor data integration. The user, however, has to manually select appropriate sensors to query its data. Similarly, the Microsoft SensorMap [51] allows the user to select sensors based on their location, type, and keywords. Xively [6] is yet another middleware that is scalable, secure and used for data storage. It acts as an interface between users and devices to provide real-time device control. SenseWeb [52] like Xively, interfaces the sensors and allows to execute the search queries on their data. Considering the dynamic nature of the IoT, these tools and middlewares fail, as they allow sensor selection through only static meta-information. A detailed description and challenges faced by existing IoT middlewares can be found in [53]. Zhang [54] proposed a fuzzy logic based middleware for the IoT called, *FuzWare*. It converts the context data of the IoT objects into fuzzy representation through uncertain reasoning and learning rules. Apart from these tools, several research efforts have been made to use context-properties of devices for search and discovery.

Pfisterer *et al.* [55] have proposed semantic WoT framework, SPITFIRE. It provides search service for entities based on their current state that are inferred by semantic descriptions of sensors embedded into entities. The notable characteristic of SPITFIRE is that it makes use of prediction model to build semantic descriptions of sensors by calculating similarity patterns among them. As IoT is inherently dynamic in nature, the states of entities change within a short interval, that SPITFIRE fails to address. For Web-enabled things, Mayer and Guinard [56] developed a generic semantic discovery engine. Meta-data of these devices describes semantic information like device-related and contextual information are recorded manually by human operators through a Web-based interface. The drawback is that prototype of the system lacks extensive evaluation.

Perera *et al.* [57], [58] have designed Context Aware Sensor Selection and Ranking Model (CASSARAM) that models users requirements to context information of the sensors. The SSN ontology [59] is used to represent sensor context properties, and then search space is indexed and ranked based on comparative priority weights. The efficiency of the search is improvised by using distributed search technique through parallel processing of different server nodes to index and rank the sensors. The disadvantage is that registry and subscription of sensors to server nodes is not addressed in this work. The technique proposed by Buchina [60] allows to express context properties of devices as a set of atomic and independent tags. The context information is used as a search factor by utilizing naming conventions to express the context inside the Domain Name System Service Discovery (DNS-SD). The process of assigning the context information to services has not been addressed in this work.

Ebrahimi *et al.* [61], [62] have addressed the search problem through meta-heuristic algorithm [63], [64], an Ant-Clustering algorithm (AntClust) is developed to create a Sensor Semantic Overlay Networks (SSONs) that group semantically related sensors. The search queries from the user are forwarded in this method to particular SSON only. Disadvantage of the system is a time-consuming off-line computing phase, and it is also vulnerable to dynamicity issues. Michel *et al.* [65] implemented a search engine and middleware, *Gander*, for a pervasive computing environment. Real-time search functionality is provided by *Gander*, through incorporating context of data items. The relationships of data items among themselves and their surrounding environment are taken as context, and the sampling techniques through peer-to-peer methods are applied on these spatio-temporal data. Due to privacy concerns, *Gander* does not store user's private data centrally which increases the need for local data storage at the user device.

Sensor search and selection service architecture proposed by Hsu *et al.* [66] have used context properties of sensors like, battery life, reliability, accuracy, etc., along with user requirements to determine the optimal score for ranking the sensors. However, the network lifetime of the gateways in the proposed architecture vary to a large extent between different scores of context properties and it is very difficult to determine an optimal score. Lunardi *et al.* [67] presented Context-Based Search Engine, COBASEN. It is made up of two modules: context module and a search engine, where former module provides semantic descriptions of the entities and later module uses these descriptions to interact with the desired entity. The advantage of COBASEN is that it employs an effective middleware that manages a large number of entities with overlapping and redundant functions. The disadvantage is that as context information of entities increases, the indexing time of COBASEN also increases resulting in search delays.

A framework for semantic WoT, Context-aware Sensor Search Framework (CASSF) was developed by Gong *et al.* [68]. This framework facilitates the search of sensors by building Resource Description Framework (RDF) graphs to model context properties of sensors. The search efficiency is improved by reducing the targeted search space size through a query approximation technique called Threshold Algorithm for Sensor Information (TASI). Although, the search is scalable considering the dynamic nature of the IoT, CASSF involves a huge computational time overhead as every sensor is contacted to find its relevance in the search result which ensues sluggish response time. Wang and Cao [69] have proposed the use of context information in the event management at the IoT middleware. The fuzzy ontology is used to model the unexpected events, based on which query execution plans are generated. A Notable feature of the system is that context-related queries are remodeled as context independent and targeted data window is segmented based on event patterns and their context information. However, due to a large scale of IoT networks, ontology size increases and thus increases the query execution time.

Arnaboldi *et al.* [70] have developed a context-aware middleware for mobile applications. It considers the socio-economic attributes of the users to form communities with similar interests. A detailed API is provided for mobile application developments to address the opportunistic issue of the IoT objects through context and social attributes. The community formation phase of the system is time consuming and increases with the number of users/devices connecting to the community. Context-aware search system for IoT was presented by Chen *et al.* [71]. It addresses the search issue related to objects in the IoT, in addition to their related information. Hidden Markov model [72] is used to recognize the user requirements with that of context-properties of objects. With this system, the user is able to search for a variety of content information of the sensors, but the system fails to search the predefined context information in the database. Paparrizos *et al.* [73] developed a method to embed the context metadata of sensors along with their semantic meaning to aid sensor search and visualization. The sensor data is tagged dynamically with context metadata by calculating the occurrence frequency of a keyword. The implemented system provides an extensive visualization of results based on types of query. The search engine suffers from real-time delivery of results due to dependency on the ranking method to calculate the scores of newly created metadata.

Zhou *et al.* [34] have addressed the problem of searching in frequently updated time-stamped data generated by the IoT devices. The data sources are virtualized and

TABLE VI
COMPARISON OF DIFFERENT CONTEXT-BASED SEARCH TECHNIQUES

Authors	Year	Concepts	Advantages	Disadvantages
Perera *et al.* [58]	2014	CASSARAM, captures context properties of sensors and then indexes and ranks the search space based on comparative priority weights.	Employed distributed search by implementing parallel processing over different server nodes to gather local high ranking sensors.	Collection and management of context-properties is inherently difficult for large number of sensors.
Michel *et al.* [65]	2014	Gander, RDF graph-based model, to represent context of data item with environment and its neighboring items.	Gander's implementation in RESTful simplifies application development process.	Due to privacy concerns Gander does not stores user's private data centrally, which increases the need for local data storage at user device.
Ebrahimi *et al.* [61]	2015	A meta-heuristic Ant-Clustering algorithm (AntClust) creates a Sensor Semantic Overlay Networks (SSONs) that group semantically related sensors.	Search queries from user are forwarded in this method only to particular SSON.	Time-consuming off-line computing phase, and system is also vulnerable to dynamicity.
Lunardi *et al.* [67]	2015	COBASEN framework, consists of Context Module which stores semantic characteristics of devices and a Search Engine to query semantic properties.	Iot application development is simplified as middleware complexity is hidden.	Restrict number of devices per search session.
Chen *et al.* [71]	2016	Hierarchical context-model based on ontology.	Context recognition model is superior than rudimentary statistical approach.	Context reasoning phase of the system is time consuming, and thus as the search session prolongs query running time increases.

integrated to support heterogeneous devices. It supports different types of queries like range, distance, and time-window. The Geohashing and Uniform Resource Locator (URL) patterns are used to index data that requires a dedicated database and thus suffers from performance degradation due to frequent updates. Dragon, a data discovery engine was conceptualized by Kolcun and McCann [74]. The static information of sensors are stored in a distributed fashion across different data tables in the IoT network. A routing algorithm is implemented to propagate and match the query with the help of routing table that holds destination node address, next hop node address and distance to it. However, a time-consuming setup phase is required to initialize data table for routing. Hu *et al.* [75] proposed a cloud platform for vehicular networks to ease mobile application development. It identifies and classifies the sensors based on their context-information. Optimized deployment of the sensors aids in better management and enhances the retrieved data quality. However, mobility impact of the vehicles has not been addressed in this work.

Table VI lists most recent context based approaches along with their advantages and disadvantages. The widespread development of semantic technologies has attracted a large number of researchers and this area is one of the popular and highly adopted technique in search and discovery of the IoT resources.

B. IoT Object-Based Search Techniques

1) Location-Based Search Techniques: The location of objects in the IoT networks plays an important role in search techniques. The user requirements usually center around location parameter that can refer either to geographical location coordinates, *i.e.,* in terms of longitude and latitude, or logical location, in terms of relative references (an, *e.g.,* book is placed on the second shelf). In this subsection, we discuss search systems that consider location as a search parameter.

Mayer *et al.* [76] considered the logical location of sensors as an attribute in search space construction that is modeled as a tree. A tree-based search technique was used to locate a sensor, either at the local node or at the distant node based on routing mechanism of the query [77]. The system adapts to the large size of the IoT network, by limiting communication between the adjacent nodes. However, it experiences identification problem, as sensor nodes are named statically and are prone to errors. The search service for sensors is developed through P2P architecture by Liang and Huang [78]. It is a location-aware system where the search is performed by applying space filling curve to the location tagged data. The data measurements from the sensors are collected and annotated with location information through space filling curve technique. Though the system supports simple spatiotemporal queries, it fails to answer queries that are constructed by combining two or more requirements.

Frank *et al.* [79] used heuristics of location attribute to develop a search system for sensor networks. A certain probability is assigned to each device based on their location, and queries are resolved through heuristic approach by considering location probability of devices. The implemented work establishes energy reduction by routing the query to only a small number of sensor objects. Although, the scheme is scalable for large sensor networks, it can only be used to locate the objects tagged to users. Yap *et al.* [80] proposed a search system called *MAX*, that takes the logical location of an object with respect to each other and surrounding environment as a search parameter. Pull-based technique is used to retrieve object tags that consists of descriptions of objects in a text form. *MAX* does not maintain indexes of tags, and thus it handles mobility and dynamicity issue appropriately. The implementation of *MAX* as a distributive system does not address the scalability

TABLE VII
COMPARISON OF DIFFERENT LOCATION-BASED SEARCH TECHNIQUES

Authors	Year	Concepts	Advantages	Disadvantages
Mayer *et al.* [76]	2012	Logical location based sensor search system, by organizing search space.	System adapts to large size of IoT network, by limiting communication between adjacent nodes only.	Suffers from identification problem, as sensor nodes are named statically and is prone to errors.
Liang *et al.* [78]	2013	Location-aware search system, space filling curve is applied to location tagged data.	Reduced data-processing cost and in-time notification of a typical event.	Issues like privacy, security policy for sensor web are not studied.
Wang *et al.* [81]	2015	Geographical location based service discovery.	Division of measurement area of sensors, decreases search space size.	Computationally intensive and thus cannot be applied in IoT networks.
Du *et al.* [84]	2015	Multi-fold index model taking into account geographical location.	Supports range value queries.	Susceptible to distance sensitivity problem.

issue as all the substations are contacted for the query resolution.

Wang *et al.* [81] have proposed the use of geographical locations coordinates for sensor services. A distributed architecture is employed by the system consisting of sensor nodes, services, and gateways. The measurement area of sensors and gateways (sensors interconnect with each other through gateways, forming a local network) are bounded by a rectangle and then indexed using R-trees through geographical indexing method. It is computationally expensive and thus is of limited use in IoT networks. Li *et al.* [82] developed a distributed indexing technique for data with multiple dimensions. Based on the geographical location of sensors, the deployment area is divided into a number of zones and event-data are hashed to these zones. The system supports a range queries by constructing the search tree with multiple data dimensions. The drawback of the system is that routing algorithms are used to resolve range queries that are resource intensive and fails to perform in a large scale network.

Du *et al.* [83], [84] have used location attributes of sensors to index and organize sensor networks. A method named Distributed Index for Features in Sensor Networks (DIFS) is developed to assist the range query resolution. Indexes are constructed in a hash table and visualized as a tree. The range values of specific geographical locations are stored in nodes of the hash tables, and a non-root node is allowed to have multiple parents. However, if the parent nodes are located far away from the child node then the system's performance deteriorates due to the distance sensitivity problem. Fathy *et al.* [85] used the unsupervised machine learning algorithm to gather and circulate indexes in a decentralized IoT network. The IoT objects are clustered based on their location and a network gateway is assigned to each cluster for organizational and management related operations. The index are built at gateways and communicated to upper-level discovery servers, where they are aggregated and stored. The drawback is that the number of gateways deployed in the system depend on the number of the clusters formed during search space construction step, and thus is a hindrance in real-time deployment.

Fredj *et al.* [86] have employed geographical location based clustering and aggregation search method to look for IoT devices and their services. The physical objects are clustered and managed by semantic gateways based on their location, and at the global level, there exists a hierarchy of semantic gateways. A routing table is maintained at each gateway that is constructed according to the semantic descriptions of the IoT objects. Queries are forwarded and matched using these routing tables at gateways. The system assumes the static location of objects and thus fails to address mobility parameter and it also does not consider maintenance cost of routing tables during update operation of the semantic descriptions [87].

Abdelwahab *et al.* [88] proposed a sensor discovery and selection technique on a cloud computing platform. A virtual layer is implemented on top of the sensor networks that are grouped according to their geographical locations. These virtual layers form edge nodes that can be scaled up horizontally based on the requirements and thus solves scalability issues. Shemshadi *et al.* [89], [90] employed spatial properties of the sensors to cluster them according to their correlations scores. A graph-based approach is used to measure the similarity of a sensor according to its parental and location relationship with other sensors. However, the system does not address the social relationship attribute.

Relative locations of sensors along with six other parameters concerning sensor state and communication properties were used by Shah and Sardana [25] to search for the sensors in the IoT. A pull based approach is used by query generator node to gather information about all other nodes, this information is then ranked based on Euclidean distance. However, the scheme gets affected from scalability issues as all nodes are contacted to gather current state information. Michel and Julien [91] devised a method to distribute cloud-based solutions for the IoT search and discovery, called *cloudlets*, across physical locations. The location-based services are offered by cloudlets that leverage IoT objects proximity with each other. Although local queries that require information about the immediate neighborhood are resolved quickly, queries that require information about the IoT objects in a remote location takes longer resolution time.

The location-based search techniques for IoT are listed in Table VII along with their potential benefits and drawbacks. A considerable number of applications are being developed today that try to solve everyday challenges. These applications tend to work on preferences and likeliness of users, among which location plays a vital role. Though, some of these applications have constructed location-aware search systems for the IoT, they do not address the user requirements significantly.

2) User-Oriented Search Techniques:

a) Social structure-based search techniques: To facilitate easy task completion, the IoT devices communicate with each other to get a better perspective of their deployment environment. Usually, the devices communicate with known set of other devices thus forming a group of frequently contacted nodes. This concept is leveraged to form Social-IoT that mimics social networks of humans. Several works are carried out in search and discovery process of the IoT objects that utilize the concept of social networks.

Shen *et al.* [92] implemented a search engine for Cyber-Physical System (CPS), by constructing a prediction model based on Bayesian networks. The system is oriented towards humans and utilize their traveling habits. The system leverages social links between objects and the user to build a Distributed Hash Table (DHT) that is used for indexing. Although, the model can predict unusual movements, it fails to address the behavior of the user towards a particular response. Liang and Cao [93] have presented an overview of different social-aware context middleware platforms. Role played by the different social properties in modeling a middleware is discussed in detail. However, integration of the middleware architecture with the IoT application system is missing. Nitti *et al.* [94] described a method for finding the IoT objects using navigability parameters in social networks. A decentralized search system is constructed by computing degree of centrality of a node in the network. Due to the use of traditional graph structures for the search operation, the system fails to address dynamic nature of the IoT.

Jung *et al.* [95] developed a hypergraph-based overlay network model as a discovery mechanism for Social-IoT. They modeled Social-IoT by scrutinizing distinct characteristics and structural facets of human-centric social networks, to fully understand how and to what extent these objects mimic behaviors of the humans. Bhaumik *et al.* [96] devised a method to group sensors based on their social structure. Ownership information of sensor is used to cluster them into social groups that have a common purpose. Deshpande *et al.* [97] conceptualized a Machine-4-Machine (M4M) abstract model that enables sharing of the IoT devices among friends in a Social-IoT network. A notable feature in this work is that sharing can be controlled by the degree of associations of the IoT devices with their peers. The limitation is that system does not consider security threats, like man-in-middle attack, posed during the formation of associations.

A desktop-based search engine for CPS system was designed in [98] by Deng *et al*. The correlation graph of user activity is constructed based on two kinds of memory patterns, explicit and implicit. The former pattern recognizes the events at particular context while the latter memory pattern is used when the context is unknown. The use of virtual window to speed-up correlation graph construction of the user activities is the advantage of the system, yet monitoring the activities of the user leads to privacy and security concerns. *Paraimpu*, a platform for resource and service sharing on the Social-WoT was developed by Pintus *et al*. [99]. It connects Hypertext Transfer Protocol (HTTP) compatible WoT objects, through abstraction of their services, to the Web to support device and service lookup. Though the platform is scalable, it uses a centralized server to store and manage device descriptions that leads to single point of failure.

Luis-Ferreira and Jardim-Goncalves [100] described an approach to integrate human emotions and sensations to objects in the IoT, that is modeled on the Human brain to incorporate sensations associated with objects in IoT. The tags are used to build emotions and sensations databanks from which queries are resolved. The tagging mechanism is not addressed and it is a difficult task to gather emotions of users related with the IoT objects. Wu *et al.* [101] presented a security assessment model for social P2P sensor networks to select a particular service. The chance based discovery theory was implemented with a KeyGraph structure to select an appropriate service [102]. Through the service forwarding scheme, the system is able to reduce communication overhead by forwarding only a few service request packets. Due to use of cache at every node to maintain the service request packets, the real-world deployment of the system is difficult to attain in a resource constrained IoT devices.

The recent works that utilize the social structure of IoT network for device and service look-up are compared in Table VIII. Use of efficient graph-based structure to capture relationship among IoT devices leads to a better discovery technique. However, management of such a structure for large scale IoT network is not an easy task.

3) Heterogeneous Search Techniques:

a) Semantic and ontology-based search techniques: Ontology-based models have also been proposed to address search and discovery problem in IoT. Ontologies are knowledge representations of a specific domain in terms of concepts, types and relationships among them. They imbibe the semantics of the IoT data into their concepts and relations and help to build domain, task and approach specific search systems. This subsection reviews the use of ontologies in the search process for the IoT objects.

Calbimonte *et al.* [103] have used an ontology-based query technique to address search problem in large scale networks. In this approach, sensor generated data and its associated semantic metadata are recorded in ontology that has query capabilities. García-Castro *et al.* [104] have developed a core ontological model, that represents sensor networks as well as services. De *et al.* [105] presented a conceptual architecture on the IoT platform and several semantic ontological based models that capture the sensor's state. The World Wide Web Consortium (W3C)'s Incubator Group proposed a Semantic Sensor Network (SSN) ontology that allows the description of sensors and their characteristics, to address the issue of interoperability of metadata annotations [59].

Mietz *et al.* [106] demonstrated the use of semantic Web technologies, Resource Description Framework (RDF) and SPARQL, in search of data from heterogeneous sources. The prediction models are engaged to determine the similarity score of the search space with the query that is represented in RDF format and specified in SPARQL. Only fundamental attributes of the sensors like spatial and temporal characters are used in the development of prediction model. Nayak and Parhi [107] proposed a semantic based

TABLE VIII
COMPARISON OF DIFFERENT SOCIAL STRUCTURE-BASED SEARCH TECHNIQUES

Authors	**Year**	**Concepts**	**Advantages**	**Disadvantages**
Wu *et al.* [101]	2015	Security assessment model for services selection in social P2P based sensor networks.	Supports integration of sensor network and social networks in a system securely	Use of cache to maintain service request packets, prevents system from real-world deployment in resource constrained IoT devices.
Shen *et al.* [92]	2015	Search engine for Cyber Physical System (CPS), by constructing prediction model based on Bayesian networks.	Oriented towards humans and utilize their traveling habits.	Fails to address behavior of user towards a particular response.
Nitti *et al.* [94]	2016	Decentralized search system to find IoT objects using navigability parameters in social networks.	Reduced computations cost to calculate degree of centrality of a node in network.	Due to use of traditional graph structures for search operation, system fails to address dynamic nature of IoT.
Deshpande *et al.* [97]	2016	Machine-4-Machine (M4M) abstract model, to enable sharing of IoT devices among friends in a Social-IoT network.	Sharing is controlled by degree of associations of IoT devices with their peers.	System does not consider security threats, like man-in-middle attack, posed during formation of associations.
Jung *et al.* [95]	2016	Hypergraph-based overlay network model as discovery mechanism for Social-IoT.	Model adapts itself flexibly to the dynamic behavior of IoT objects.	Does not consider distributed management of IoT interactions.

sensor discovery and selection engine. Universal Description, Discovery, and Integration (UDDI) framework is extended to include the semantic descriptions of services; these descriptions are added at the registration time of the service. Matchmaking algorithm is introduced to find services based on the keyword search. It does not describe the semantic annotation process of services during registry and the cost incurred.

Searching in WoT domain was addressed by Christophe *et al.* [108] by incorporating the WoT objects with semantic profiles. The machine learning algorithms are used to compare the profiles of different objects. Upon arrival of the request, its context is used to select an algorithm for query execution. However, the system is not validated. Another semantic based service discovery approach is presented in [109]. Based on semantic technologies, a middleware was developed to perform search on the sensor data through their context properties. Mayer *et al.* [56] addressed the use of semantic techniques to discover the WoT objects based on multiple mapping scheme. It makes use of compressed data representation models to identify the IoT resources. Although, interoperability is preserved across the heterogeneous devices in IoT, the search system suffers from dynamic issues of identification, as the addresses of devices are to be known in advance.

A Semantic framework for looking up smart objects was developed by Alam and Noll [110]. The resources advertise their services along with other information like location, identification, name, and semantic descriptions to middlewares that capture them and register it in their service directory. The mechanism eases registration process and thus improvises the search phase. Hu *et al.* [111] proposed metadata model to account for observations and context of the data generated by sensors. They also developed a user interface to aid discovery of sensors based on geographical positions. However, the model described is implemented particularly for earth observation sensor deployments and requires extensive modification for generic applications.

Yang *et al.* [112] proposed hierarchical ontological model to improvise search system reasoning. The abstract terminologies are expanded vertically in this system, and a query is decomposed into lower level terminologies and matched with directories containing sensor descriptions. The advantage of the system is that search space drastically decreases in size as queries are decomposed. The disadvantage is that influence of configurational differences in ontologies and their complexity is not considered. Perera and Vasilakos [113] implemented a knowledge driven sensor configuration tool, CASSCOM. It is an IoT middleware that facilitates data search operation without the use of any query representation language. The semantic and context properties are used to drive the search process. However, privacy and security measures of the sensor data in middleware are not addressed.

Chaochaisit *et al.* [114] have developed a location-aware sensor search system by utilizing Human Location Sensor Ontology. Domain knowledge of human positions is modeled as classes in Ontology Web Language (OWL) for automated sensor classification. Using different constructs, the sensor and location parameters can be described in accordance with the user's location context. The advantage of the system is that the IoT features have been considered and knowledge structure of ontology can be extended to different domains. But the ontology is not validated to handle demands from the IoT applications that come at a different scale and varying time requirement over a specific platform. Zhou and Ma [115] presented an ontology focused Web service matching algorithm aimed at the IoT systems. As a proof-of-concept, the vehicular sensors are portrayed into the ontology. The semantic similarity and relativity scores of the sensors are calculated by search algorithm, and then merged together to get the maximum value of Web services. These values along with matching degree are used to look for appropriate Web services.

TABLE IX
COMPARISON OF DIFFERENT ONTOLOGY-BASED SEARCH TECHNIQUES

Authors	Year	Concepts	Advantages	Disadvantages
Garcia-Castro *et al.* [104]	2012	Ontology to model sensor networks and services.	Facilitates reuse of ontologies and data models.	Domain specific ontology evaluation techniques are not used.
Zhou *et al.* [115]	2012	Ontology focused web service matching algorithm aimed at IoT systems.	Highly scalable and generic algorithm.	Score calculation and merging stages are computationally intensive.
Sezer *et al.* [121]	2015	Specialized Smart Home sensor ontology.	Heterogeneous data items and different types of sensors are extensively handled by the ontology.	Results are not hardware and platform independent.
Chaochaisit *et al.* [114]	2016	Human Localization Sensor (HLS) ontology to enable search for sensors.	HLS ontology is compatible with other major ontologies like QUDT, GeoNames, SSN, and DOLCE Ultra Lite.	HLS ontology is not validated to handle demands from IoT applications, which come at large scale and varying time requirement, over a specific platform.
Dey *et al.* [116]	2016	Extended SSN ontology to capture salient features of an energy meter sensor.	Efficiently resolves metadata query.	Semantic smart energy service is specific to energy meter use case.

Dey *et al.* [116] in their work extended SSN ontology to capture salient features of an energy meter sensor in resolving metadata query, and further enhanced its role in semantic smart energy services. However the semantic smart energy service is specifically designed to cater to the needs of energy meter use case and is not generic to all applications. ForwarDS-IoT, a distributed semantic repository was proposed by Gomes *et al.* [117]. Semantic descriptions of an IoT object are handled at three levels: SSN and Semantic Actuator Network (SAN) [118] for IoT object description, Geo Vocabulary [119] for location data and OWL-S [120] for modeling services. SPARQL end-points are provided as the query interface. However, the work is not evaluated and thus its performance is unknown.

Sezer *et al.* [121] have simulated smart home use case to propose an ontology on Smart Home. SSN ontology was extended in this work to handle heterogeneous data items and different types of sensors. The performance of the ontology application was gauged for different scenarios. The results are however dependent on the hardware used and vary significantly on different hardware configurations [122]. Cabral *et al.* [123] also extended the SSN ontology to develop sensor cloud ontology that includes time series data and location coordinates in semantic representations. A fitness function was modeled to rank and index the sensors. This model outperforms prediction models described in [29] and [33]. However, in this search system query resolution time is increased and thus the system is not applicable to time-critical search applications.

Table IX examines different ontology-based approaches for IoT. As the size of the IoT networks continues to grow, the sensor generated data is also expected to grow in exponential size as compared to metadata descriptions of objects and sensors. Efficient management of such a situation by an ontological model still remains a challenge.

b) Resource and service discovery: The resources in the IoT networks are embedded with sensors and actuators to monitor a specific event in real-world and act upon it when occurred. Resources offer some kind of services which are abstracted as software components and are presented as APIs through middleware for the application development. For an application to address day-to-day activities, it is a difficult task to discover resource and its associated services due to the opportunistic presence, dynamicity, and largeness of the IoT network. This subsection presents a review of research efforts that address these challenges.

Ruta *et al.* [124] have presented a solution to resource discovery, allotment and sharing in swarm intelligence scenarios. A framework is designed that allows novel and advanced retrieval of resources in highly dense contexts based on semantics of the annotation. However, the system is not flexible as compared with standard approaches. It also suffers from limited computational resources and service volatility due to unpredictable device mobility and network link unreliability. Datta *et al.* [125] developed a resource discovery framework for the IoT that supports different communication protocols. A notable component of this framework is a proxy layer that contains drivers to facilitate integration of various communication techniques. Although, the framework is generic with respect to the connectivity issue, it lacks a common syntax to describe heterogeneous IoT resources.

Maekawa *et al.* [126] used the sensors embedded in objects to detect activities of users and then retrieve and display the relevant Web pages to show additional information about them. Objects are clustered according to their use in performing an activity, and queries are generated by the system automatically without any intervention from the user. A lightweight Web browser is implemented and deployed across various daily usage objects to retrieve and display the information fetched from the queries. Though, context of the IoT objects are well modeled, inter-object relationships are not considered in the design of the system and leads to ambiguous description of the Smart Home application domain. Giang *et al.* [127] proposed Browsing as a Service for IoT. *SeedHTML*, an HTML protocol is developed that allows direct access to IoT objects without having to pass through a middleware. Each IoT device hosts a unique SeedHTML URL that points to service offered by it. A cloud server, known as *BuddyThing*, is included in back-end to manage SeedHTML URLs. The clusters of devices offering the same services are formed in the cloud to reduce the

IoT search space. This approach requires a dedicated naming system to manage tags in the cloud and thus requires conversions between traditional DNS that increases complexity of the system.

Nunes *et al.* [128] implemented *ViSIoT*, a visual search engine for the IoT object discovery. It is a pull based approach where the sensor data is gathered by a central repository, stored in a generic data format through a conversion operation called marshal, and thus supports heterogeneous devices. However, as with other centralized techniques, it suffers from scalability issue and the marshal operation is time-consuming. IoT Search Engine (ISE) is another search engine developed by Jin *et al.* [129], consisting of three components to index, update and query the IoT search space. ISE supports only RFID-based search and fails to offer service for other IoT objects. Fortino *et al.* [130] have developed a resource discovery framework using two standard Web protocols (Representational State Transfer (REST) and JavaScript Object Notation (JSON)). Indexes of the IoT objects are built using the category of the domain to which the objects belong. The advantage of the framework is it eases the query resolution process through a centralized repository. The disadvantage is indexing of the IoT objects based on their application domain which fails to account for the objects that find their usage across multiple domains.

Leu *et al.* [131] have presented Distributed Resource Discovery (DRD) architecture for the IoT. P2P communication model is implemented to ease the discovery operation, where the IoT objects register with each other through Machine-to-Machine (M2M) technique. The IoT objects are identified uniquely by hashing the Media Access Control (MAC) address. However, these MAC addresses can be spoofed which leads to identification issue. IoT-SVK [132] supports the search for the IoT objects based on their content, location and keyword descriptions. The data values are retrieved from sensors, sampled, and stored in repositories as keywords. Further, they are indexed through B^+ and R-tree structures. The real-time service offered by IoT-SVK engine increases transmission delays and is prone to communication failures.

Wang *et al.* [133] have constructed, *Snoogle*, a search engine to lookup physical objects. Textual descriptions of objects are represented as keywords, and a query can be constructed to use them. A three-layer architecture is constructed made up of the physical objects, an IP (Index Points) and a KeyIP (Key Index Point). The descriptions of objects/sensors are collected and indexed by the IP, while KeyIP acts as a manager, controlling and coordinating the IPs. *Snoogle* fails to address scalability and dynamicity demand of the IoT networks, due to the use of KeyIP as a centralized server. It provides searches through pseudo-static metadata and thus generated results are approximate. *Microsearch* is yet another search system that enables discovery of the sensor embedded physical objects [134]. It stores textual descriptions of sensors in a centralized repository. It assumes that sensors are available to access *via* a reliable network connectivity and thus falls short in the IoT scenario where the device availability is dynamic.

Federica and David [135] have developed a system for distributed service discovery using the Distributed Hash Table (DHT) in the Peer-to-Peer (P2P) networks for look-up service. The system supports multi-attribute and range queries. The Radio-frequency identification (RFID)-based scenarios are experimented in this work. Only exact matches are supported in this system as no scoring method is employed. Li *et al.* [136] conceptualized resource discovery in Social Internet of Things (SIoT) by implementing a 3-dimensional Cartesian coordinate system. The user preferences and motion are observed to extract recurring patterns and clustered based on a similarity score. The sub-communities are constructed based on the computed score in a 3-dimensional Cartesian coordinate system. Later, similar sub-communities are clustered into a virtual global community. The communication overhead is reduced in this approach by adjusting search radius. The system suffers from computational complexity, as the similarity is calculated at the various level of community formation. Human behavior was also not considered as a parameter in the subcommunity construction [137].

Noguchi *et al.* [138] adopted ubiquitous computing technique to address the issue of device discovery in a Smart Environment. Two ontologies are defined to incorporate intensional and extensional knowledge of the domain. The advantage of the system is on-demand matching of semantic annotations in ontologies that reduces computational overhead. The system is complex due to inclusion of two domain dependent ontologies. Yachir *et al.* [139] implemented a user-centric, event aware, service-oriented framework to monitor events in ambient environments. Dynamic service discovery and selection algorithms are developed that guarantees continuous service through self-adaption to unexpected changes in the environment. It uses service replacement and replanning in the case of failure. The impact of different parameters like number of ambient services class, number of detected events, and events monitoring strategy is not addressed in this work.

Li *et al.* [140] formulated a scheduling problem to crawl newly captured events from periodically sleeping sensors. Constrained optimization strategy was used to crawl and index the generated events in time. A sleep-aware schedule method, named *EasiCrawl*, is implemented for achieving a near-optimal expected latency in receiving events. *EasiCrawl* fails to address the situation when sensor's sleep plans are unknown. Jara *et al.* [141] presented a mechanism for resource discovery across several communication technologies. *Digcovery* a search system made up of a centralized registry server (called *Digrectory*) was developed. A device has to register itself with this registry so as to publish its data. Different *Digrectory* servers are deployed to handle specific types of communication protocol. The advantage of the system is that it supports the legacy objects search.

Kamilaris *et al.* [142] extended DNS system capabilities to include discovery mechanism for services. It includes a top-level domain for services offered by devices in the URL. Device and their service registration are maintained in a central repository, may leads to single point of failure. Georgakopoulos *et al.* [143] proposed a service-based IoT architecture where every IoT component (including the IoT devices, cloud resources, and application components) is exposed a service, allowing dynamic discovery, composition, and integration with other services. The challenge, however is unification of all the components into the architecture that requires protocols that are yet to be designed.

Rykowski [144] proposed a method for management of REST-based services acting as proxies for IoT devices. It monitors REST resources by a hierarchical set of directories, with the possibility of smart searching for *the best device* according to at-the-place device's availability and functionality, overall context (including geo-location) and personal preferences. The system is resistant to changes in network addresses of the devices and their services, as well as core system points such as directories. But subscription and management of devices are unanswered in this work. Namatame *et al.* [145] discussed WoT resource management. Layered architecture is proposed to enable abstraction of physical objects and their network connections. Management labor is divided into two levels, at the local level a stand-alone server called *uBox* is deployed and at the global level, a middleware interconnects these uBoxes. Privacy of local applications deployed in local *uBox* is maintained through this approach. However, overlay networks are not addressed in this work.

Bastani *et al.* [146] addressed the sensor selection and classification problem by using a sparse estimation technique to develop a classification algorithm that select the appropriate sensors in human material handling tasks. Use of approximation method speeds up the classification stage and thus makes the system suitable for online decision-making process. The use of weighing scheme to represent sensors on a different scale is computationally intensive. Qian and Che [147] developed a security enabled search framework. An identity-based cryptosystem is used to preserve the privacy of the IoT objects while security and authentication are managed by encryption and decryption of valuable data in the IoT object through Elliptic Curve Cryptography (ECC) algorithm. Although ECC has been designed for computationally hindered IoT objects, it increases processing time and thus real-time search response of the system is affected.

Faqeeh *et al.* [148] developed a specialized search engine to retrieve information related to the IoT domain. It complements addition of URLs related to the IoT field by the user, but its ranking process depends on tradition frequency count of the keywords and thus is returns results that are almost similar to generic search engines. Service composition method on Web was proposed by Tang *et al.* [149]. It is based on logical inference of Petri nets and Horn clauses. A forward-chaining algorithm is used to transform service composition problem into a Horn clause inference problem, then Petri nets are used to retrieve composite services. As there are a large number of services present in the repository that generates an even larger number of operating rules, candidate clauses are selected only at the arrival of new queries. Though this approach addressed scalability issue the major set back is that services cannot be composed on-the-fly and needs to wait for the query arrival.

El Haddad *et al.* [150] proposed Web service selection and composition. Transactions are extracted from the user defined tasks and combined with QoS parameters to compose services that fulfill the given complex task. In an IoT application, although QoS parameters are given high priorities, transaction management does not play a pivotal role, as the IoT resources allow retrieval/stimulation operations and do not support undo, commit and other transactional operations. Wu and Khoury [151] have addressed the issue of service composition where a tree-based search algorithm is implemented on a cloud computing platform [152]. The user requirements are considered to build a tree, whose branches contains all possible solutions; then a pruning algorithm is applied to remove all illegal branches. In the final stage, a heuristic search algorithm is applied to obtain the optimal solution. The performance of the system is improvised by reducing response time through pruning [153]. However, scalability remains a challenge, with the increase in tree size.

Scalable and self-configurable service discovery architecture for P2P networks is proposed by Cirani *et al.* [154]. Utilization of P2P technologies enables deployment of distributed and large-scale infrastructure for service discovery. The backbone of service discovery architecture is a IoT gateway, that keeps track of any things joining or leaving its network and updates the list maintained at its CoAP server. This server is used in service discovery phase where the required information of the connected resources is collected through a GET request. Leitner *et al.* [155] proposed a service composition method in the WoT domain. Agreements between the end-user and service provider like QoS, delivery time, costs, etc., are used to form rules based on the rules the services offered by various providers are composed to meet the requirements of the users. The drawback, however, is that prior knowledge of business constraints are required to build rules.

Trendy was implemented by Butt *et al.* [156] to provide context aware, registry-based service discovery. An interoperable interface is provided by this system making use of CoAP-based RESTful Web services. An adaptive timer is introduced to control energy consumption. A grouping mechanism is used to localize query traffic based on location tag of a query generator.

Analysis of different Service Discovery approaches, along with their advantages and disadvantages is listed in Table X. The penetration of Internet into the physical objects and their surrounding environment has resulted in the creation of various applications that support our day-to-day activities. A complex requirement from the user has to be addressed by integrating different services, and an IoT application relies on selection, composition, and management of services provided by the IoT objects. However, these needs cannot be fulfilled easily with ever growing size and dynamic nature of the IoT. Novel approaches are needed for service discovery and selection to support easy application development for the IoT.

V. Evaluation and Insights Gained

In previous section we reviewed different techniques that are applied to develop and implement a search system for the IoT. In this section we perform a comparative study of the reviewed research publications and outline our views on them.

TABLE X
COMPARISON OF DIFFERENT SERVICE DISCOVERY APPROACHES

Authors	Year	Concepts	Advantages	Disadvantages
Rykowski [144]	2014	Smart searching for devices according to at-the-place device's availability and functionality, over all context and personal preferences.	Resistant to changes in network addresses of devices and their services, as well as core system points such as directories.	No support for frequent disconnections of devices and their states.
Yachir *et al.* [139]	2015	Context aware event monitoring for ambient environments.	Supports self-adaptation to unpredicted changes using services replacement and replanning in case of failure.	Impact of parameters like number of ambient services class, number of detected events, etc., on performance of the proposed system are not studied.
Li *et al.* [136]	2016	Search mechanism based on preference and movement pattern similarity for SIoT.	Improvement in average delay.	Day-to-day human social behavior is not considered in similarity measurement.
Datta *et al.* [125]	2016	Resource discovery framework	Central registry is designed to store configurations of resources and indexes them based on configuration parameters.	Framework suffers from interoperability as a common format or syntax to describe the resources, units and domains are not yet to be standardized.
Ruta *et al.* [124]	2016	Swarm intelligence solution to resource discovery, allotment and sharing.	Advanced retrieval of resources in highly dense contexts, based on semantics of annotation.	Suffers from limited computational resources and service volatility due to unpredictable device mobility and network link unreliability.

A. Comparative Study

As seen in the previous section, each paper focused on a particular aspect of the search process. To gain a through understanding and deeper perspectives of the different kinds of search techniques, we perform a comparative study of existing works. These works are evaluated according to the following metrics:

i) *Search Approach:* It refers to the class of search technique. These classifications are explained in detail in the Section II-C.
ii) *Design Principle:* It indicates the fundamental design strategy adopted by the work to handle search operation in the IoT (like, Indexing, Crawling, Ranking, Search Space Structuring, Query Processing, Recommendation, *etc.,*).
iii) *Data Model:* It points out the data format and mechanism used by the search technique to process and store the IoT Data.
iv) *Architecture:* It specifies the architectural design implemented by the search technique. It can be either centralized, distributed, or P2P.
v) *Data Type:* It mentions the type of data handled by the search system. We have compared different search techniques with different data type parameters as:
   a) *IoT Data Type:* In Content-based Search Technique, IoT Data Type refers to the nature of data generated by the IoT object (like Single-valued, Multi-valued, Time-series, *etc.,*).
   b) *Context Data Type:* In Context-based Search Technique, it indicates type of the context data used by the search system.
   c) *Location Data:* In Location-based Search Technique, it mentions the nature of location data (*e.g.,* geographical coordinates, logical coordinate, *etc.,*) considered by the search technique.
vi) *Dataset Used:* It indicates the dataset used by the search technique along with the reference.
vii) *Prototype/Simulation:* It specifies whether the search system was implemented as a simulation model or as a prototype (*e.g.,* Web-based, java-based, mobile application, *etc.,*).

Apart form these metrics, Social Structure-based Search Techniques are evaluated on an additional parameter, Social Relationship, that describes the nature of social links between the IoT objects and the user. Also, to compare Semantic and Ontology-based Search Techniques, we consider the following three additional metrics:

i) *Ontology Language:* It refers to the type of language on which the ontology was developed (*e.g.,* Resource Description Framework (RDF) [157], Web Ontology Language (OWL) [158], *etc.,*).
ii) *Support for Queries:* It specifies whether the search system supports for semantic queries or not, if it support then the query language is included (*e.g.,* SPARQL Protocol and RDF Query Language (SPARQL) [49]).
iii) *IoT Domain:* It indicates the IoT application domain on which the ontological model was developed (like, Smart Home, Smart City, *etc.,*).

However, we do not consider architecture metric for Semantic and Ontology-based Search Techniques, as most of the techniques are implemented on centralized repositories (except Mietz *et al.* [106] which is implemented on a Peer-to-Peer model).

The Tables XI to XV provide a summary of evaluations based on the above metrics for different kinds of the search and discovery techniques reviewed in Section IV.

B. Discussions and Insights Gained

In this section, we outline the insights gained from the review, and comparative study. The comparison of all the classes of the search techniques is summarized with their advantages, disadvantages, and challenges in the Table XVI.

1) Content-Based Search Techniques: Content-based search approaches are used to identify patterns in data generated by the IoT devices and thus help in the decision-making

TABLE XI
EVALUATION OF REVIEWED RESEARCH PUBLICATIONS: CONTENT-BASED SEARCH TECHNIQUES

Research Publication	Search Approach	Design Principle	Data Model	Architecture	IoT Data Type	Dataset Used	Prototype/ Simulation
Elahi *et al.* [29]	Time-related Search	Ranking	Single and Multi-period Prediction Model	Centralized	Time-series occupancy data	ETH [159] & MERL [160]	Simulation
Ostermaier *et al.* [30]	Real-time Search	Indexing	Aggregated Prediction Model	Centralized	Time-series availability data	Bicing [161]	Web-based prototype
Mietz *et al.* [31]	Event-based Search	-	Bayesian Network	Centralized	Time-series availability data	Bicing [161]	Simulation
Truong *et al.* [32] [33]	Event-based Search	Crawling	Fuzzy Logic	Centralized	Time-series multivalued data	NOAA [162] IntelLab [163] MavHome [164]	Web-based prototype
Zhou *et al.* [34]	Spatiotemporal-based Search	Middleware	-	Distributed	Frequently updated spatiotemporal data	SmartSantander [165]	Web-based prototype
Zhang *et al.* [35]	Event-based Search	Ranking	Grey Recurrence Dynamical Model	Distributed	Time-series multivalued data	NOAA [162] IntelLab [163]	Simulation
Bijarbooneh *et al.* [36]	Event-based Search	Middleware	Belief Propagation Estimation Model	Centralized	Time-series multivalued data	IntelLab [163]	Simulation
Zhang *et al.* [37], [41]	Event-based Search	Ranking	Multi-step Prediction Model	Distributed	Time-series multivalued data	IntelLab [163]	Simulation
Ihm *et al.* [38]	-	Indexing	Grid-based Partitioning	Centralized	-	-	Simulation
Jiang *et al.* [40]	-	-	Cuckoo Search-SVM Model	Centralized	Single-valued data	Goniometer Sensor Data	Simulation
Anas *et al.* [42]	Event-based Search	-	Genetic Algorithm	-	Time Stamped Trajectory data	Taxi Dataset	Simulation
Shemshadi *et al.* [43], [44]	Spatiotemporal-based Search	Crawler	Restructuring, Merging and Enriching	Centralized	Time-series multivalued data	Google Maps and Xively [6]	Web-based prototype

process. They allow access to historical and real-time data and various mathematical and statistical models find their applications here to detect the missing values, outliers, future readings, *etc.*, and aid in detection of the event generated in physical world. These search techniques give reasonable accuracy of the search results and can be employed easily in crawling and indexing phase of the search operation in the IoT. Most of the search techniques reviewed in this approach implement a centralized repository to catalog the IoT data which might lead to a single point of failure and cannot accommodate to the exponential growth of number of the IoT objects connected to the IoT. Also, these techniques have to deal with the raw data generated by the IoT objects due to which additional inference modules have to be developed to obtain high level and meaningful results. They suffer from high bandwidth utilization due to frequently changing IoT data. The cache techniques can be developed to reduce frequent communications between the search system and the IoT network. It is noted that most of the reviewed works in this category of search techniques employ simulation models and use datasets to measure the performance of the developed search systems. There is a need for prototype implementation here that deal with real-world IoT applications. The challenges faced by content-based search techniques are mainly due the dynamic nature of the IoT data, where data is generated in high volume, velocity and varieties.

2) Context-Based Search Techniques: Context-based search methods provide an efficient management to control the IoT objects, as they make use of status and operational properties of the IoT objects. High-level meaningful results are generated, that can be represented in natural-language. Use of semantics to describe the IoT objects aid in easy query resolution and

TABLE XII
EVALUATION OF REVIEWED RESEARCH PUBLICATIONS: CONTEXT-BASED SEARCH TECHNIQUES

Research Publication	**Search Approach**	**Design Principle**	**Data Model**	**Architecture**	**Context Data Used**	**Dataset Used**	**Prototype/ Simulation**
Pfisterer *et al.* [55]	Event-based Search	Crawling	RDF Triple Store, and Periodic & Correlation Prediction Model	Centralized	High-level Object Status Information	-	-
Mayer *et al.* [56]	Metadata-based Search	Semantic Representation and Resolution	Microdata [166] & Microformats [167] Markup Language, and JavaScript Object Notation (JSON)	Centralized	-	-	Simulation
Perera *et al.* [57], [58]	Context-based Search	Ranking	SSN Ontology [59], and Apache Jena SDB & TDB [168]	Distributed	QoS-based Context Data	Phenonet Project [169], LSM Project [170], and Bureau of Meteorology [171]	Java-based Prototype
Ebrahimi *et al.* [61], [62]	Context-based Search	Search Space Restructuring	SSN Ontology [59]	Centralized	QoS-based Context Data	LSM Project [170], Bureau of Meteorology [171], and Air Quality Sensor Dataset [172]	Simulation
Michel *et al.* [65]	Event-based Search	Middleware and Query Processing	LIGHTS Tuple Space Framework [173]	Distributed	User Behavior and Activity Data	Time-stamped User Activity Data	Android Application
Hsu *et al.* [66]	Event-based Search	Middleware	SSN Ontology [59]	Centralized	QoS-based Context Data	-	Web-based Prototype
Lunardi *et al.* [67]	-	Middleware and Indexing	PostgreSQL database	Centralized	QoS-based Context Data	-	Java-based Prototype
Gong *et al.* [68]	Context-based Search	Query Processing	RDF Triple Store	Centralized	QoS-based Context Data	-	Simulation
Wang *et al.* [69]	Event-based Search	Query Processing	Fuzzy Ontology	Centralized	-	Traffic Simulation Data	Simulation
Chen *et al.* [71]	Context-based Search	Hidden Markov Model for User Activity Prediction	RDF Triple Store	Centralized	User Activity Data	Case Study Data	-
Paparrizos *et al.* [73]	Context-based Search	Ranking	Sensor Metadata Repository	Centralized	Real-time Environmental Observation Data	Swiss Experiment Platform [174]	Web-based Prototype
Zhou *et al.* [34]	Context-based Search	Middleware	RDF Triple Store	Centralized	Frequently Updated Spatiotemporal Data	SmartSantander [165]	Web-based Prototype
Kolcun *et al.* [74]	Context-based Search	Query Routing	Distributed Data Table	Peer-to-Peer	Logical Place Identifiers	-	Simulation

thus provide an effective method to rank the query results in the search operation. Search systems reviewed in this category have mostly used a distributed repository to index the context-data and thus scale efficiently to a large number of the IoT objects. They have also developed a prototype implementations to validate their search approaches and thus establish the use of these techniques in real-world deployments. But most of the reviewed works in this category use QoS-based context-data of the IoT objects and concentrate less on the user and environmental context information. These techniques require a dedicated context-aware middleware/server for managing the context related information that incurs additional burden on the operational costs and increases the complexity of the search system implementation. Further data acquisition

TABLE XIII
EVALUATION OF REVIEWED RESEARCH PUBLICATIONS: LOCATION-BASED SEARCH TECHNIQUES

Research Publication	Search Approach	Design Principle	Data Model	Architecture	Location Data	Dataset Used	Prototype/ Simulation
Mayer *et al.* [56]	Location-based Search	Indexing	Microdata [166] & Microformats [167] Markup Language and Hierarchal Tree Structure	Centralized	Logical Place Identifiers	600 Simulated Sensors	Simulation
Liang *et al* [78]	Location-based Search	Data Acquisition and Tagging	Semantic Model	Peer-to-Peer	Geographical Location Data	-	Web-based Prototype
Frank *et al.* [79]	Event-based Search	Query Resolution	-	Centralized	Physical Location and Observed Network Cells	Real-world Data (within office setup) and Momentum Project Simulation Data [175]	Mobile Application
Yap *et al.* [80]	Location-based Search	Query Resolution	-	Distributed	Textual Descriptions and Tags	-	Web-based Prototype
Wang *et al.* [81]	Ontology-based Search	Spatial Indexing	Semantic Service Description Model [176]	Distributed	Latitude and Longitude Data	Experimental Dataset with 10,000 Sensors across 22 Locations	Simulation
Li *et al.* [82]	Event-based Search	Query Resolution	-	Distributed	Binary Encoded Zone Data	Event Information and their Geographic Zone Data	Simulation
Du *et al.* [83], [84]	Location-based Search	Indexing	Spatial Index Tree	Centralized	Spatiotemporal Data	-	Simulation
Fathy *et al.* [85]	Spatiotemporal-based Search	Indexing	Information Repositories	Distributed	Geographical Co-ordinates	Automated Surface Observing System [177]	Simulation
Fredj *et al.* [86]	Location-based Search	Indexing	Hierarchal Ontology-based Data Model	Distributed	Geographical Location	OWL-S Service Retrieval Test Collection [178]	Java-based Prototype
Shemshadi *et al.* [89], [90]	Event-based Search	Data Correlation	Object Relationship Model	Centralized	Latitude and Longitude	Xively [6]	Simulation
Shah *et al.* [25]	Location-based Search	Ranking	-	Centralized	Virtual Coordinates	-	Simulation
Michel *et al.* [91]	Location-based Search	Indexing	-	Distributed	Geographical Region Data	Real-world and Simulated Mobility Trace Data Sets	Simulation

from the IoT objects also remain a challenge. Machine learning algorithms can be used to cluster related IoT objects based on their context-information and thus reduce the search space size and address the interoperability issue.

3) Location-Based Search Techniques: Location of the IoT object is strongly associated with the user preferences and thus plays a pivotal role in query resolution phase of the search techniques. Research publications reviewed in this category of search techniques effectively utilize indexing structures to speed up query execution time and produce real-time results. They also develop distributed data models to store the location related information of the IoT object which enhances solving of the local queries that enquire about the IoT objects in the immediate vicinity. Use of different kinds of the location data (*e.g.,* geographical coordinates, virtual coordinates, place identifiers, tags, *etc.,*) to describe the property of the IoT objects is well established across different works and various prototypes have also been developed to carter the real-world task in the IoT applications. But, most of the works in this category do not address the spatiotemporal-data and thus suffers from mobility issue; recent advancements in the field of edge computing can be used here to address it. In addition, most of the works fail to identify the co-location problem (*i.e.,* IoT objects that offer same or different services are located at same physical location) and they implicate security treats and hence efficient encryption and decryption techniques are to be developed for the resource constrained IoT objects.

4) Social Structure-Based Search Techniques: The IoT devices communicate with a known set of other devices and the user thus forming a group of frequently contacted nodes;

TABLE XIV
EVALUATION OF REVIEWED RESEARCH PUBLICATIONS: SOCIAL STRUCTURE-BASED SEARCH TECHNIQUES

Research Publication	**Search Approach**	**Design Principle**	**Data Model**	**Architecture**	**Social Relationship**	**Dataset Used**	**Prototype/ Simulation**
Shen *et al.* [92]	Event-based Search	Indexing	Distributed Hash Table	Distributed	User Behavior and Movement Patterns	MIT Reality Dataset [179]	Simulation
Nitti *et al.* [94]	Spatiotemporal-based Search	Crawling	Social Relationship Graph	Distributed	Friendship	Stanford Large Network Dataset [180]	Simulation
Jung *et al.* [95]	Event-based Search	Search Space Structuring	-	Centralized	Inter-object Social Relationships	CASAS Dataset [181]	Smart Home Prototype
Bhaumik *et al.* [96]	Real-Time Search	Search Space Structuring	Social Relationship Graph	Centralized	IoT object and User Relationship	-	-
Deshpande *et al.* [97]	Event-based Search	Friendship Recommen-dations	Social Relationship Graph	Centralized	Friendship	-	Mobile Application
Deng *et al.* [98]	Event-based Search	Indexing	Correlation Graph	Centralized	Event Relation-ships	User Study Dataset	Desktop Application
Pintus *et. al.* [99]	Event-based Search	Friendship Recommen-dations	-	Centralized	IoT object and User Relationship	-	Web-based Prototype
Luis-Ferreira *et al.* [100]	Context-based Search	Indexing	Emotions & Sensations Databank	Centralized	IoT object and User Relationship	-	-
Wu *et al.* [101]	Context-based Search	Crawling	Intuition Graph [182]	Peer-to-Peer	IoT object and User Relationship	Open Sourced Vulnerability Database [183]	Simulation

this concept is leveraged to associate social links among the IoT objects and the user. Social Structure-based search techniques utilize graph-based methods to model relationships and provide recommendations to search system users. The research publications reviewed in this category of search techniques have modeled the user behavior and likeliness to associate him/her with the IoT objects that offer the required services. Most of the works provide a prototype implementations of the proof-of-concept developed and thus ascertain their usage in the real-world deployment. Future Social Structure-based search techniques should orient the application development with the mobile applications and provide mashups and plug-ins to integrate with the already thriving social applications to enhance user experience. However, due to the existence of multiple data sources from a large number of IoT objects, data ownership is a concern and needs to be addressed through well established standards and regulations. These search techniques are also prone to identity thefts and suffers from the traffic congestion and scalability issues due to the large scale of social links present in the IoT network.

5) Semantic and Ontology-Based Search Techniques: These search techniques effectively manage data of the IoT objects through the use of a well defined rule set that describes relationships among the IoT objects, users, applications and services. They support the use of crawlers to build indexes of the IoT objects and thus are compatible with Web-search techniques. Research publications reviewed in this category of the search techniques allows representation of the complex real-world events in the machine readable format by developing the semantic and ontological models across different IoT application domains (like, Smart Home, Smart Space, Smart Viticulture, *etc.,*). These works support efficient query management by employing techniques such as similarity computations, search intent identifications, correlations computations, *etc.,* to produce accurate search results. However, the raw IoT data has to be transformed into semantic descriptions so as to be consumed by the semantic search systems that incurs additional costs and suffers from performance issues if the semantic rules are ill formed and thus requires expertise in the IoT application domain for which the IoT search systems are implemented. Designing a particular ontology that fulfills the huge range of applications that are expected to appear with the future IoT remains a challenge.

6) Resource and Service Discovery Techniques: These search and discovery techniques differentiate between the IoT enabling entities (like sensors, actuators, *etc.,*) and objects, where an object is embedded with sensors and actuators. Most of the research publication reviewed in this category of search techniques solve complex relationships among the IoT objects and support for QoS-based parameters by employing distributed architecture. They also support the service look-up facility provided by different IoT objects through

TABLE XV
EVALUATION OF REVIEWED RESEARCH PUBLICATIONS: SEMANTIC AND ONTOLOGY-BASED SEARCH TECHNIQUES

Research Publication	**Search Approach**	**Design Principle**	**Model**	**Ontology Language**	**Support for Queries**	**IoT Domain**
Mietz *et al.* [106]	Time-related Search	Indexing	Aggregated Prediction Model	RDF	Yes (SPARQL)	Smart City (Smart-Santander Dataset [165])
Nayak *et al.* [107]	Context-based Search	Matching & Ranking	Semantics Repository	-	No	-
Christophe *et al.* [108]	Context-based Search	Search Intent	Joint Probability Distribution	OWL	No	Smart Space
Yun *et al.* [109]	Context-based Search	Middleware	Semantics Repository	-	No	-
Alam *et al.* [110]	Ontology-based Approach	Matching	Microformat based Service Description	OWL	Yes (SPARQL)	Intelligent Transportation System
Hu *et al.,* [111]	Ontology-based Approach	Semantic Modeling	Spatiotemporal Description of Geo-events	SensorML [184]	Yes (Web-interface)	Earth Observation
Yang *et al.* [112]	Ontology-based Approach	Query Processing	Sensor Descriptions Model	-	No	Smart Home
Perera *et al.,* [113]	Context-based Search	Ranking	Task Description Model	OWL	Yes (SPARQL)	Smart Agriculture and Smart Environment
Chaochaisit *et al.* [114]	Location-based Search	Semantic Modeling	Location Descriptions	OWL	No	Smart City
Zhou *et al.* [115]	Context-based Search	Similarity Computation	Semantic Similarity and Relativity Model	OWL	No	Smart Vehicles
Dey *et al.* [116]	Ontology-based Approach	Semantic Modeling	Spatiotemporal Description of Energy Meters	RDF	Yes (SPARQL)	Smart Meter
Gomes *et al.* [117]	Ontology-based Approach	Middleware	Semantic Repository	OWL	Yes (SPARQL)	-
Sezer *et al.* [121]	Ontology-based Approach	Semantic Modeling	Publish/subscribe Model	RDF	No	Smart Home
Cabral *et al.* [123]	Ontology-based Approach	Ranking	Sensor Fitness Model	RDF	Yes (SPARQL)	Smart Viticulture

service identification, selection, and composition capabilities. However, these methods require additional middlewares to manage the index registry and thus incur additional operational costs. They also suffer from opportunistic presence and dynamicity problems of the IoT, where availability of the IoT devices and their services are frequently changing and leads to increased latency for real-time service composition and discovery. These search techniques are specific to a application domain, as a particular IoT entity is designed for a specific purpose and a dedicated task (*e.g.,* pollution monitoring device in Smart City domain, smoke detector in Smart Home domain) and suffers from generality issue due to lack of a upper semantic model that represent all kind of the IoT objects.

VI. FUTURE RESEARCH DIRECTIONS

With the proliferation of IoT resources in the physical world, search needs for humans will gradually shift from the cyber world to physical world. An ideal search system's vision in IoT will be to discover any kind of IoT object or its data at anytime with minimal inputs from the user by learning through his historical searches and present needs. The previous section laid out critical shortcomings of the search techniques in IoT, and in this section we present some important directions for future research that allows to develop an efficient and effective search system that satisfy the information needs of the user.

1) *Indexing Techniques:* Due to heterogeneous and multimodal nature of the IoT data that is generated at high-velocity and high-volume it encounters the scalability challenge to the solutions for search and discovery operations. Efficient indexing mechanism that sort and rank the IoT data are needed. The indexed data is to be stored and processed in distributed fashion to accomplish the real-time query matching. The Geohash encoding of the spatiotemporal data is a promising direction to this end [185], [186]. Fathy *et al.* [187] have elaborated on

TABLE XVI
COMPARISON OF DIFFERENT IoT SEARCH TECHNIQUES

Category	Approach	Advantages	Disadvantages	Challenges
IoT Data based	Content based	• Allows access to historical and real-time data. • Statistical and prediction models can be employed. • Reasonable accuracy of search results. • Missing values, future outcomes, and error detection is relatively easy.	• Have to deal with low-level sensor data. • Frequent communication between sensors and search system leads to wastage of IoT device's energy. • Do not provide information of operational costs to manage IoT objects. • Meaningful and high-level results cannot be achieved.	• *Dynamicity*: prone to errors between two successive readings in short interval of time. • *Scalability*: has to send data request to all the nodes. • *Heterogeneity*: have to deal with data in various formats.
	Context based	• Information about the deployment environment and IoT object's state are available. • Do not need to deal with low-level sensor data. • Provides efficient control to manage IoT objects through context-information. • Generates high-level and meaningful results.	• Hard to acquire context-information. • IoT objects have to be intelligent, to gather information about their state. • Predicting missing values is difficult. • Complex due to many levels of interlinked properties.	• *Management*: requires dedicated context-aware server for management. • *Data Acquisition*: hard to retrieve context-information from IoT objects. • *Generality*: strongly integrated with specific application requirements.
IoT Object based	Location based	• Supports indexing and ranking methods to reduce search space. • Strongly coupled with query routing techniques.	• Utilizes large storage space, when index of mobile IoT objects are built. • Queries that require information about remote locations take more time to resolve.	• *Identification & Naming*: difficulty to uniquely identify IoT objects when they are co-located. • *IoT object safety and Security*: prone to physical damage and operational condition when location is revealed. • *Mobility & Dynamicity*: has to deal with mobile IoT objects, that frequently change their availability status.
	Social Structure based	• Leverage social networks formed by humans. • User behavior and likeliness are well modeled.	• Data ownership in Social-IoT is not regulated. • Suffers from traffic congestion and scalability issues due to large scale of social network.	• *Privacy*: prone to identity thefts as personal and private informations are collected.
	Semantic and Ontology based	• Allows representation of complex real-world events. • Results can be modeled in natural language. • Crawlers and Indexer can be utilized effectively. • Preprocessing and query subdivision are supported.	• Domain knowledge is required. • IoT data is to be transformed to compatible format. • Suffers from performance issues, due to ill formed rules and huge size of data. • Requires integration of middleware for management.	• *Interoperability*: various ontologies across different domains have to be integrated. • *Standardization*: no well defined and accepted ontology for all applications.
	Resource and Service Discovery	• Support for QoS parameters is well established. • Complex relationship are managed.	• Requires additional publish/subscription management. • Real-time service composition and discovery suffers from latency.	• *Opportunistic presence*: devices and their services are not available all the time. • *Specificity*: limited to application domain, due to lack of general semantic model to represent all kind of IoT objects.

the directions for future research work to be undertaken for indexing the IoT Data.

2) *Prediction Models:* Large scale of IoT network pose an immediate challenge to the crawlers of the search system, as they cannot contact every node in the network to retrieve the data. Further, due to the volatile nature of the IoT data and opportunistic presence of devices, it becomes a tedious task to crawl the IoT network. To overcome this challenge, prediction models can be used to sketch the future readings of the sensors, and analyze data streams to infer patterns and correlations. Regression technique can be considered to predict the future data streams in real-time [188]. Deep Learning techniques (Convolutional Neural Networks,

Recurrent Neural Networks, Long Short Term Memory, Autoencoders, Variational Autoencoders, Generative Adversarial Networks, Restricted Boltzmann Machine, Deep Belief Network, and Ladder Networks) are detailed in [189] that can be used for predictive analytics on streaming data in IoT.

3) *Progressive Search:* Search space in the IoT is largely unstructured due to heterogeneous devices and thus onerous efforts have to be put by search system to find a query matching device owing to which there is increase in latency between successive search steps. The progressive search algorithms decrease the time taken during each subsequent steps by performing incremental search. Ma and Liu [190] have delineated three strategies, listed below, to reduce the size of search space:
   i) *Coarse-to-fine:* Properties of the IoT objects are categorized based on their discriminative power and then used to filter elements from the search space. Initially, rough features are utilized to drastically reduce the size of search space and latter subjected to fine filtering to increase search accuracy.
   ii) *Near-to-distant:* The spatiotemporal data of IoT objects is used to structure the search space. This strategy is used to re-rank the matching results.
   iii) *Low-to-High:* The search space is further subjected to filtering based on device access permission, relevance to user, rating, *etc.*

   Other such search strategies are to be designed to achieve real-time, highly-accurate, multi-faceted results.
4) *Domain Specific Knowledge:* The IoT supports multitudinous applications where the tasks performed vary significantly across different domains. To generalize and operate under many different scenarios and applications, a search system should be developed in a modular fashion where the modules works independently of each other. To perform domain specific search operations, an ontology module is to be designed that embodies domain knowledge in the form of concepts and relationships. In the recent past, several domain specific ontologies have been designed *viz.* Smart Home [191], Smart Agriculture [192], Smart Water Management [193], *etc.* that provide promising direction here. Some of the existing domain specific and generic ontologies for the IoT are listed in [194].
5) *Tackle Mobility Issue through Edge Computing:* Edge computing paradigm brings the computing facilities of cloud to the edge of network, *i.e.,* IoT objects. As devices in IoT network are mobile and change their location frequently, data acquisition, device management, access control and other issues are challenging to handle. Edge computing eases these tasks as the computation is closer to the device, and it provides a promising direction with following benefits: fault tolerance, scalability, load distribution, low latency, local processing, better security and privacy. Frameworks proposed in [195]–[197] are quite useful in this context.

VII. Conclusion

Advancements in disciplines like sensor networks, cloud computing, middlewares, communication protocols, *etc.*, has led to the proliferation of a large number of Internet-connected objects around us. This means we have huge range of choices to select devices/services that offer similar functions, that lead to lookup and discovery operation. In this paper, we have presented fundamentals of search and discovery procedure for the IoT. The State-of-the-art research works which have addressed these challenges, are reviewed based on their solutions and design principles. A comparative study of the research works along with critical discussions, lessons learned, future research directions, and challenges are presented. Addressing these challenges will allow next generation search techniques to recognize and respond to user queries and, as a result, to a great extent satisfy the information needs of the users.

Acknowledgment

The authors thank anonymous reviewers for their insightful comments on improving the paper.

References

[1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.
[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.
[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.
[4] P. Barnaghi and A. Sheth, "On searching the Internet of Things: Requirements and challenges," *IEEE Intell. Syst.*, vol. 31, no. 6, pp. 71–75, Nov./Dec. 2016.
[5] A. Zaslavsky, C. Perera, and D. Georgakopoulos, "Sensing as a service and big data," in *Proc. Int. Conf. Adv. Cloud Comput. (ACC)*, 2012, pp. 21–29.
[6] *Xively*. Accessed: Jul. 21, 2017. [Online]. Available: https://www.xively.com/
[7] *OpenIoT*. Accessed: Jul. 21, 2017. [Online]. Available: https://www.openiot.eu/
[8] *Shodan*. Accessed: Jul. 21, 2017. [Online]. Available: https://www.shodan.io/
[9] *Thingful*. Accessed: Jul. 21, 2017. [Online]. Available: https://www.thingful.net/
[10] *Censys*. Accessed: Jul. 21, 2017. [Online]. Available: https://censys.io/
[11] *Reposify*. Accessed: Jul. 21, 2017. [Online]. Available: https://www.reposify.com/
[12] K. Romer, B. Ostermaier, F. Mattern, M. Fahrmair, and W. Kellerer, "Real-time search for real-world entities: A survey," *Proc. IEEE*, vol. 98, no. 11, pp. 1887–1902, Nov. 2010.
[13] D. Zhang, L. T. Yang, and H. Huang, "Searching in Internet of Things: Vision and challenges," *Proc. 9th IEEE Int. Symp. Parallel Distrib. Process. Appl. (ISPA)*, Busan, South Korea, 2011, pp. 201–206.
[14] D. Zeng, S. Guo, and Z. Cheng, "The Web of Things: A survey," *J. Commun.*, vol. 6, no. 6, pp. 424–438, 2011.
[15] 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, pp. 1–29, 2016.
[16] C. D. Truong, "Routing and sensor search in the Internet of Things," Ph.D. dissertations, Dept. Comput. Sci. Eng., Univ. Lübeck, Lübeck, Germany, 2014.
[17] M. García-Valls, P. Bellavista, and A. Gokhale, "Reliable software technologies and communication middleware: A perspective and evolution directions for cyber-physical systems, mobility, and cloud computing," *Future Gener. Comput. Syst.*, vol. 7, pp. 171–176, Jun. 2017.

[18] C. Mouradian *et al.*, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.
[19] C. Perera, C. H. Liu, and P. Christen, "Device search and selection for CPS," in *Cyber Physical Systems*. Boca Raton, FL, USA: Taylor & Francis, 2015, pp. 67–99.
[20] P. Saint-Andre, "Extensible messaging and presence protocol (XMPP): Core," Internet Eng. Task Force, Fremont, CA, USA, RFC 3920, 2011.
[21] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for wireless sensor networks," in *Proc. 3rd Int. Conf. Commun. Syst. Softw. Middleware Workshops (COMSWARE)*, Bengaluru, India, 2008, pp. 791–798.
[22] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Comput.*, vol. 10, no. 6, pp. 87–89, Nov./Dec. 2006.
[23] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," Internet Eng. Task Force, Fremont, CA, USA, RFC 7252, 2014.
[24] J. B. Blair, D. L. Rabine, and M. A. Hofton, "The laser vegetation imaging sensor: A medium-altitude, digitisation-only, airborne laser altimeter for mapping vegetation and topography," *ISPRS J. Photogrammetry Remote Sens.*, vol. 54, nos. 2–3, pp. 115–122, 1999.
[25] M. Shah and A. Sardana, "Searching in Internet of Things using VCS," in *Proc. 1st Int. Conf. Security Internet Things*, 2012, pp. 63–67.
[26] I.-Y. Ko, H.-G. Ko, A. J. Molina, and J.-H. Kwon, "SoIoT: Toward a user-centric IoT-based service framework," *ACM Trans. Internet Technol.*, vol. 16, no. 2, pp. 1–21, 2016.
[27] *The IoT and the Blue Jeans: A Retail Story*. Accessed: Jan. 19, 2018. [Online]. Available: https://www.intel.com/content/www/us/en/internet-of-things/smart-retail-solutions.html
[28] H. F. Hamann, "From sensors to smarter solutions with physical analytics," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2015, pp. 1–3.
[29] B. M. Elahi, K. Romer, B. Ostermaier, M. Fahrmair, and W. Kellerer, "Sensor ranking: A primitive for efficient content-based sensor search," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, San Francisco, CA, USA, 2009, pp. 217–228.
[30] 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)*, Tokyo, Japan, 2010, pp. 1–8.
[31] R. Mietz and K. Römer, "Exploiting correlations for efficient content-based sensor search," in *Proc. IEEE Sensors*, 2011, pp. 187–190.
[32] C. Truong, K. Römer, and K. Chen, "Fuzzy-based sensor search in the Web of Things," in *Proc. Int. Conf. Internet Things (IOT)*, Wuxi, China, 2012, pp. 127–134.
[33] C. Truong and K. Römer, "Content-based sensor search for the Web of Things," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, 2013, pp. 2654–2660.
[34] Y. Zhou, S. De, W. Wang, and K. Moessner, "Enabling query of frequently updated data from mobile sensing sources," in *Proc. IEEE 13th Int. Conf. Ubiquitous Comput. Commun. (IUCC)*, Chengdu, China, 2014, pp. 946–952.
[35] P. Zhang, Y.-A. Liu, F. Wu, and B. Tang, "Matching state estimation scheme for content-based sensor search in the Web of Things," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 11, pp. 1–12, 2015.
[36] 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.
[37] 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.
[38] 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, pp. 1–10, May 2014.
[39] K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, "Selective dissemination of XML documents based on genetically learned user model and support vector machines," *Intell. Data Anal.*, vol. 11, no. 5, pp. 481–496, 2007.
[40] M. Jiang *et al.*, "A cuckoo search-support vector machine model for predicting dynamic measurement errors of sensors," *IEEE Access*, vol. 4, pp. 5030–5037, 2016.
[41] P. Zhang, Y.-A. Liu, F. Wu, W. Fan, and B. Tang, "Content-based sensor search with a matching estimation mechanism," *Special Section Integr. Technol. Ambient Intell. Sensor Netw.*, vol. 99, no. 9, pp. 1949–1957, 2016.
[42] M. Anas, S. Khatab, and A. Badr, "HeuristicIoT: A framework for augmenting heuristic search algorithms by Internet-of-Things data," *Int. J. Comput. Inf. Sci.*, vol. 12, no. 1, pp. 133–141, 2016.
[43] A. Shemshadi, Q. Z. Sheng, and Y. Qin, "Thingseek: A crawler and search engine for the Internet of Things," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Pisa, Italy, 2016, pp. 1149–1152.
[44] A. Shemshadi *et al.*, "Searching for the Internet of Things: Where it is and what it looks like," *Pers. Ubiquitous Comput.*, vol. 21, no. 6, pp. 1097–1112, 2017.
[45] V. Vasilev, G. Iliev, V. Poulkov, and A. Mihovska, "Optimization of wireless node discovery in an IoT network," in *Proc. IEEE Globecom Workshops (GC Workshops)*, San Diego, CA, USA, 2015, pp. 1–5.
[46] W.-S. Li and K. S. Candan, "Integrating content search with structure analysis for hypermedia retrieval and management," *ACM Comput. Surveys*, vol. 31, no. 4, pp. 13–38, 1999.
[47] G. D. Abowd *et al.*, "Towards a better understanding of context and context-awareness," in *Proc. Int. Symp. Handheld Ubiquitous Comput.*, vol. 40. Karlsruhe, Germany, 1999, pp. 304–307.
[48] D. Le-Phuoc, H. Quoc, J. X. Parreira, and M. Hauswirth, "The linked sensor middleware—Connecting the real world and the semantic Web," in *Proc. Semantic Web Challenge*, 2011, pp. 1–8.
[49] *SPARQL 1.1 Federated Query*. Accessed: Jan. 19, 2018. [Online]. Available: https://www.w3.org/TR/rdf-sparql-query/
[50] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *Proc. IEEE Int. Conf. Mobile Data Manag.*, Mannheim, Germany, 2007, pp. 198–205.
[51] S. Nath, J. Liu, and F. Zhao, "SensorMap for wide-area sensor Webs," *Computer*, vol. 40, no. 7, pp. 90–93, 2007.
[52] A. Kansal, S. Nath, J. Liu, and F. Zhao, "SenseWeb: An infrastructure for shared sensing," *IEEE Multimedia*, vol. 14, no. 4, pp. 8–13, Oct./Dec. 2007.
[53] G. Fersi, "Middleware for Internet of Things: A study," in *Proc. IEEE Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Fortaleza, Brazil, 2015, pp. 230–235.
[54] C. Zhang, "FuzWare: A fuzzy-based middleware for context-aware service," in *Proc. IEEE 2nd Adv. Inf. Technol. Electron. Autom. Control Conf. (IAEAC)*, Chongqing, China, 2017, pp. 1181–1185.
[55] D. Pfisterer *et al.*, "SPITFIRE: Toward a semantic Web of Things," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 40–48, Nov. 2011.
[56] S. Mayer and D. Guinard, "An extensible discovery service for smart things," in *Proc. 2nd Int. Workshop Web Things (WoT)*, San Francisco, CA, USA, 2011, pp. 1–6.
[57] 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 Manag. (MDM)*, vol. 14. Milan, Italy, 2013, pp. 314–322.
[58] C. Perera *et al.*, "Sensor search techniques for sensing as a service architecture for the Internet of Things," *IEEE Sensors J.*, vol. 14, no. 2, pp. 406–420, Feb. 2014.
[59] L. Lefort *et al.*, *Semantic Sensor Network XG Final Report: W3C Incubator Group Report*, World Wide Web Consortium, Cambridge, MA, USA, 2011. [Online]. Available: http://www.w3.org/2005/Incubator/ssn/XGRssn-20110628/
[60] N. Buchina, "Extending service discovery protocols with support for context information," M.S. thesis, Dept. Math. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, pp. 1–125, 2014.
[61] 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. (SCC)*, New York, NY, USA, 2015, pp. 264–270.
[62] M. Ebrahimi, E. ShafieiBavani, R. K. Wong, S. Fong, and J. Fiaidhi, "An adaptive meta-heuristic search for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 76, pp. 486–494, Nov. 2017.
[63] P. D. Shenoy, K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, "Dynamic association rule mining using genetic algorithms," *Intell. Data Anal.*, vol. 9, no. 5, pp. 439–453, 2005.
[64] K. R. Venugopal, K. G. Srinivasa, and L. M. Patnaik, *Soft Computing for Data Mining Applications*, vol. 190. Heidelberg, Germany: Springer, 2009, pp. 63–80.
[65] J. Michel, C. Julien, and J. Payton, "Gander: Mobile, pervasive search of the here and now in the here and now," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 483–496, Oct. 2014.

[66] Y.-C. Hsu, C.-H. Lin, and W.-T. Chen, "Design of a sensing service architecture for Internet of Things with semantic sensor selection," in *Proc. IEEE Int. Conf. Ubiquitous Intell. Comput.*, 2015, pp. 290–298.

[67] W. T. Lunardi *et al.*, "Context-based search engine for industrial IoT: Discovery, search, selection, and usage of devices," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Luxembourg City, Luxembourg, 2015, pp. 1–8.

[68] H. Gong, Y. Shi, and G. Li, "Context-aware sensor search framework in semantic Web of Things," in *Proc. 8th Int. Conf. Intell. Human Mach. Syst. Cybern. Context Aware*, Hangzhou, China, 2016, pp. 94–98.

[69] Y. Wang and K. Cao, "Context-aware complex event processing for event cloud in Internet of Things," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2012, pp. 1–6.

[70] V. Arnaboldi, M. Conti, and F. Delmastro, "CAMEO: A novel context-aware middleware for opportunistic mobile social networks," *Pervasive Mobile Comput.*, vol. 11, pp. 148–167, Apr. 2014.

[71] Y. Chen, J. Zhou, and M. Guo, "A context-aware search system for Internet of Things based on hierarchical context model," *Telecommun. Syst.*, vol. 62, no. 1, pp. 77–91, 2016.

[72] B. T. H. Kumar, L. Vibha, and K. R. Venugopal, "Web page access prediction using hierarchical clustering based on modified Levenshtein distance and higher order Markov model," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, 2016, pp. 1–6.

[73] I. Paparrizos, H. Jeung, and K. Aberer, "Advanced search, visualization and tagging of sensor metadata," in *Proc. Int. Conf. Data Eng.*, 2011, pp. 1356–1359.

[74] R. Kolcun and J. A. McCann, "Dragon: Data discovery and collection architecture for distributed IoT," in *Proc. Int. Conf. Internet Things (IOT)*, Cambridge, MA, USA, 2014, pp. 91–96.

[75] X. Hu *et al.*, "Towards a service centric contextualized vehicular cloud," in *Proc. 4th ACM Int. Symp. Develop. Anal. Intell. Veh. Netw. Appl.*, Montreal, QC, Canada, 2014, pp. 73–80.

[76] S. Mayer, D. Guinard, and V. Trifa, "Searching in a Web-based infrastructure for smart things," in *Proc. Int. Conf. Internet Things (IOT)*, 2012, pp. 119–126.

[77] A. Kanavalli, D. Sserubiri, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, "A flat routing protocol for sensor networks," in *Proc. Int. Conf. Methods Models Comput. Sci. (ICM2CS)*, Wuxi, China, 2009, pp. 1–5.

[78] S. H. L. Liang and C.-Y. Huang, "GeoCENS: A geospatial cyberinfrastructure for the world-wide sensor Web," *Sensors*, vol. 13, no. 10, pp. 13402–13424, 2013.

[79] C. Frank, P. Bolliger, F. Mattern, and W. Kellerer, "The sensor Internet at work: Locating everyday items using mobile phones," *Pervasive Mobile Comput.*, vol. 4, no. 3, pp. 421–447, 2008.

[80] K.-K. Yap, V. Srinivasan, and M. Motani, "Max: Wide area human-centric search of the physical world," *ACM Trans. Sensor Netw.*, vol. 4, no. 4, pp. 1–26, 2008.

[81] W. Wang, S. De, G. Cassar, and K. Moessner, "An experimental study on geospatial indexing for sensor service discovery," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3528–3538, 2015.

[82] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, Los Angeles, CA, USA, 2003, pp. 63–75.

[83] C. Du, Z. Zhou, L. Shu, X. Jia, and Q. Wang, "An effective IoT services indexing and query technique," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber Phys. Soc. Comput. (GreenCom-iThings-CPSCom)*, Beijing, China, 2013, pp. 777–782.

[84] C. Du, Z. Zhou, S. Ying, J. Niu, and Q. Wang, "An efficient indexing and query mechanism for ubiquitous IoT services," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 18, no. 4, pp. 245–255, 2015.

[85] Y. Fathy, P. Barnaghi, S. Enshaeifar, and R. Tafazolli, "A distributed in-network indexing mechanism for the Internet of Things," in *Proc. Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Reston, VA, USA, 2016, pp. 2–7.

[86] S. B. Fredj, M. Boussard, D. Kofman, and L. Noirie, "A scalable IoT service search based on clustering and aggregation," in *Proc. IEEE Int. Conf. IEEE Cyber Phys. Soc. Comput. Green Comput. Commun. (GreenCom) IEEE Internet Things (iThings/CPSCom)*, Beijing, China, 2013, pp. 403–410.

[87] S. Tarannum, B. Aravinda, L. Nalini, K. R. Venugopla, and L. M. Patnaik, "Routing protocol for lifetime maximization of wireless sensor networks," in *Proc. IEEE 14th Int. Conf. Adv. Comput. Commun.*, 2006, pp. 401–406.

[88] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Cloud of things for sensing as a service: Sensing resource discovery and virtualization," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–7.

[89] A. Shemshadi, L. Yao, Y. Qin, Q. Z. Sheng, and Y. Zhang, "ECS: A framework for diversified and relevant search in the Internet of Things," in *Web Information Systems Engineering—WISE 2015*, vol. 9418. Cham, Switzerland: Springer, 2015, pp. 448–462.

[90] A. Shemshadi, Q. Z. Sheng, Y. Qin, and A. Alzubaidi, "CEIoT: A framework for interlinking smart things in the Internet of Things," in *Proc. 12th Anniversary Int. Conf. Adv. Data Min. Appl. (ADMA)*, vol. 27, 2016, pp. 1–15.

[91] J. Michel and C. Julien, "A cloudlet-based proximal discovery service for machine-to-machine applications," *Mobile Comput. Appl. Services*, vol. 18, no. 9, pp. 334–347, 2014.

[92] H. Shen, J. Liu, K. Chen, J. Liu, and S. Moyer, "SCPS: A social-aware distributed cyber-physical human-centric search engine," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 518–532, Feb. 2015.

[93] G. Liang and J. Cao, "Social context-aware middleware: A survey," *Pervasive Mobile Comput.*, vol. 17, pp. 207–219, Feb. 2015.

[94] M. Nitti, L. Atzori, and I. P. Cvijikj, "Network navigability in the social Internet of Things," in *Proc. World Forum Internet Things*, Seoul, South Korea, 2015, pp. 405–410.

[95] J. Jung, S. Chun, X. Jin, and K.-H. Lee, "Enabling smart objects discovery via constructing hypergraphs of heterogeneous IoT interactions," *J. Inf. Sci.*, vol. 27, no. 6, pp. 1–15, 2016.

[96] C. Bhaumik, A. K. Agrawal, and P. Sinha, "Using social network graphs for search space reduction in Internet of Things," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, Pittsburgh, PA, USA, 2012, pp. 602–603.

[97] P. Deshpande *et al.*, "M4M: A model for enabling social network based sharing in the Internet of Things," in *Proc. Int. Conf. Commun. Syst. Netw.*, Bengaluru, India, 2015, pp. 1–8.

[98] Y. Deng, Z. Zhang, J. Liao, and L. T. Yang, "SIM: A search engine by correlating scattered data sets for cyber, physical, and social systems," *IEEE Syst. J.*, vol. 11, no. 1, pp. 345–355, Mar. 2017.

[99] A. Pintus, D. Carboni, and A. Piras, "Paraimpu: A platform for a social Web of Things," in *Proc. 21st Int. Conf. Companion World Wide Web (WWW Companion)*, 2012, pp. 401–404.

[100] F. Luis-Ferreira and R. Jardim-Goncalves, "Modelling of things on the Internet for the search by the human brain," in *Proc. 4th IFIP WG 5.5/SOCOLNET Doctoral Conf. Comput. Elect. Ind. Syst. (DoCEIS)*, 2013, pp. 646–657.

[101] J. Wu *et al.*, "Chance discovery based security service selection for social P2P based sensor networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6.

[102] Y. Ohsawa and P. McBurney, *Chance Discovery* (Advanced Information Processing). Heidelberg, Germany: Springer-Verlag, 2003.

[103] J. Calbimonte, H. Jeung, O. Corcho, and K. Aberer, "Semantic sensor data search in a large-scale federated sensor network," in *Proc. Semantic Sensor Netw.*, vol. 839, 2011, pp. 23–38.

[104] R. García-Castro, O. Corcho, and C. Hill, "A core ontological model for semantic sensor Web infrastructures," *Int. J. Semantic Web Inf. Syst.*, vol. 8, no. 1, pp. 22–42, 2012.

[105] S. De, T. Elsaleh, P. Barnaghi, and S. Meissner, "An Internet of Things platform for real-world and digital objects," *Scalable Comput.*, vol. 13, no. 1, pp. 45–57, 2012.

[106] R. Mietz, S. Groppe, K. Rmer, and D. Pfisterer, "Semantic models for scalable search in the Internet of Things," *J. Sensor Actuator Netw.*, vol. 2, no. 2, pp. 172–195, 2013.

[107] C. Nayak and M. Parhi, "A semantic search engine to discover and select sensor Web services for wireless sensor network," in *Proc. ICAdC AISC*, 2013, pp. 585–591.

[108] B. Christophe, V. Verdot, and V. Toubiana, "Searching the 'Web of Things,'" in *Proc. 5th IEEE Int. Conf. Semantic Comput. (ICSC)*, Palo Alto, CA, USA, 2011, pp. 308–315.

[109] C. H. Yun, Y. W. Lee, and H. S. Jung, "An evaluation of semantic service discovery of a U-city middleware," in *Proc. 12th Int. Conf. Adv. Commun. Technol. (ICACT)*, vol. 1, 2010, pp. 600–603.

[110] S. Alam and J. Noll, "A semantic enhanced service proxy framework for Internet of Things," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun. (GreenCom) Int. Conf. Cyber Phys. Soc. Comput. (CPSCom)*, Hangzhou, China, 2010, pp. 488–495.

[111] C. Hu *et al.*, "An observation capability metadata model for EO sensor discovery in sensor Web enablement environments," *Remote Sens.*, vol. 6, no. 11, pp. 10546–10570, 2014.

[112] S. Yang, Y. Xu, and Q. He, "Ontology based service discovery method for Internet of Things," in *Proc. 4th Int. Conf. Internet Things Cyber Phys. Soc. Comput.*, Dalian, China, 2011, pp. 43–47.

[113] C. Perera and A. V. Vasilakos, "A knowledge-based resource discovery for Internet of Things," *Knowl. Based Syst.*, vol. 109, pp. 122–136, Oct. 2016.

[114] W. Chaochaisit, M. Bessho, N. Koshizuka, and K. Sakamura, "Human localization sensor ontology: Enabling OWL 2 DL-based search for user's location-aware sensors in the IoT," in *Proc. IEEE 10th Int. Conf. Semantic Comput. (ICSC)*, 2016, pp. 107–111.

[115] M. Zhou and Y. Ma, "A Web service discovery computational method for IoT system," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, vol. 2. Hangzhou, China, 2012, pp. 1009–1012.

[116] S. Dey, D. Jaiswal, R. Dasgupta, and A. Mukherjee, "Organization and management of semantic sensor information using SSN ontology: An energy meter use case," in *Proc. 9th Int. Conf. Sens. Technol. (ICST)*, 2016, pp. 468–473.

[117] P. Gomes *et al.*, "A federated discovery service for the Internet of Things," in *Proc. 2nd Workshop Middleware Context Aware Appl. IoT*, 2015, pp. 25–30.

[118] L. Spalazzi, G. Taccari, and A. Bernardini, "An Internet of Things ontology for earthquake emergency evaluation and response," in *Proc. Int. Conf. Collaboration Technol. Syst.*, Minneapolis, MN, USA, 2014, pp. 528–534.

[119] *W3C Geospatial Vocabulary*. Accessed: Jan. 19, 2018. [Online]. Available: http://www.w3.org/2005/Incubator/geo/XGR-geo/

[120] D. L. Martin *et al.*, "Bringing semantics to Web services: The OWL-S approach," in *Proc. 1st Int. Workshop Semantic Web Services Web Process Composition*, vol. 3387. San Diego, CA, USA, 2005, pp. 26–42.

[121] O. B. Sezer, S. Z. Can, and E. Dogdu, "Development of a smart home ontology and the implementation of a semantic sensor network simulator: An Internet of Things approach," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, Atlanta, GA, USA, 2015, pp. 12–18.

[122] K. R. Venugopal and R. Buyya, *Mastering C++*, 2nd ed. New Delhi, India: McGraw Hill Educ., 2013.

[123] L. Cabral, M. Compton, and H. Müller, "A use case in semantic modelling and ranking for the sensor Web," in *Proc. Semantic Web (ISWC)*, vol. 8797, 2014, pp. 276–291.

[124] M. Ruta, F. Scioscia, E. Bove, A. Cinquepalmi, and E. Di, "A knowledge-based approach for resource discovery and allotment in swarm middleware," *Inf. Sci.*, vol. 290, no. 5, pp. 1–11, 2016.

[125] S. K. Datta, R. P. F. Da Costa, and C. Bonnet, "Resource discovery in Internet of Things: Current trends and future standardization aspects," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Milan, Italy, 2016, pp. 542–547.

[126] T. Maekawa *et al.*, "Context-aware Web search in ubiquitous sensor environments," *ACM Trans. Internet Technol.*, vol. 11, no. 3, pp. 1–23, 2012.

[127] N. K. Giang, M. Ha, and D. Kim, "Buddy thing: Browsing as a service for the Internet of Things," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, New York, NY, USA, 2015, pp. 122–129.

[128] L. H. Nunes *et al.*, "A distributed sensor data search platform for Internet of Things environment," *Int. J. Services Comput.*, vol. 4, no. 1, pp. 1–12, 2016.

[129] X. Jin, D. Zhang, Q. Zou, G. Ji, and X. Qian, "Where searching will go in Internet of Things?" in *Proc. IFIP Wireless Days*, vol. 1. Niagara Falls, ON, Canada, 2011, pp. 1–3.

[130] G. Fortino, M. Lackovic, W. Russo, and P. Trunfio, "A discovery service for smart objects over an agent-based middleware," in *Internet Distributed Computing Systems*, vol. 8223. Heidelberg, Germany: Springer, 2013, pp. 281–293.

[131] M. Leu *et al.*, "Distributed resource discovery in the machine-to-machine applications," in *Proc. 10th Int. Conf. Mobile Ad-Hoc and Sensor Syst.*, Hangzhou, China, 2013, pp. 411–412.

[132] Z. Ding, J. Dai, X. Gao, and Q. Yang, "A hybrid search engine framework for the Internet of Things," in *Proc. 9th Web Inf. Syst. Appl. Conf. (WISA)*, Haikou, China, 2012, pp. 57–60.

[133] H. Wang, C. C. Tan, and Q. Li, "Snoogle: A search engine for pervasive environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 8, pp. 1188–1202, Aug. 2010.

[134] C. C. Tan, B. Sheng, H. Wang, and Q. Li, "Microsearch: A search engine for embedded devices used in pervasive computing," *ACM Trans. Embedded Comput. Syst.*, vol. 9, no. 4, p. 43, 2010.

[135] P. Federica and P. David, "A DHT-based discovery service for the Internet of Things," *J. Comput. Netw. Commun.*, vol. 2012, pp. 281–293, Sep. 2012.

[136] Z. Li, R. Chen, L. Liu, and G. Min, "Dynamic resource discovery based on preference and movement pattern similarity for large-scale social Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 581–589, Aug. 2016.

[137] V. H. Bhat, V. R. Malkani, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, "Classification of email using BeaKS: Behavior and keyword stemming," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2011, pp. 1139–1143.

[138] H. Noguchi, T. Mori, and T. Sato, "Framework for search application based on time segment of sensor data in home environment," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, 2010, pp. 261–264.

[139] A. Yachir, Y. Amirat, A. Chibani, and N. Badache, "Event-aware framework for dynamic services discovery and selection in the context of ambient intelligence and Internet of Things," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 85–102, Jan. 2016.

[140] M. Li, H. Chen, X. Huang, and L. Cui, "EasiCrawl: A sleep-aware schedule method for crawling IoT sensors," in *Proc. Int. Conf. Parallel Distrib. Syst. (ICPADS)*, vol. 3. Melbourne, VIC, Australia, 2016, pp. 148–155.

[141] A. J. Jara *et al.*, "Mobile digcovery: A global service discovery for the Internet of Things," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Barcelona, Spain, 2013, pp. 1325–1330.

[142] A. Kamilaris, K. Papakonstantinou, and A. Pitsillides, "Exploring the use of DNS as a search engine for the Web of Things," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Seoul, South Korea, 2014, pp. 100–105.

[143] D. Georgakopoulos, P. P. Jayaraman, M. Zhang, and R. Ranjan, "Discovery-driven service oriented IoT architecture," in *Proc. IEEE Conf. Collaboration Internet Comput. (CIC)*, 2015, pp. 142–149.

[144] J. Rykowski, "Managing, searching, and accessing IoT devices via rest resources," *Int. J. Comput. Netw. Commun.*, vol. 7, no. 3, pp. 171–191, 2015.

[145] N. Namatame *et al.*, "A distributed resource management architecture for interconnecting Web-of-Things using uBox," in *Proc. 2nd Int. Workshop Web Things (WoT)*, 2011, pp. 1–6.

[146] K. Bastani, W. Kim, Z. Kong, M. Nussbaum, and W. Huang, "Online classification and sensor selection optimization with applications to human material handling tasks using wearable sensing technologies," *IEEE Trans. Human–Mach. Syst.*, vol. 46, no. 4, pp. 485–497, Aug. 2016.

[147] X. Qian and X. Che, "Security-enhanced search engine design in Internet of Things," *J. Universal Comput. Sci.*, vol. 18, no. 9, pp. 1218–1235, 2012.

[148] M. Faqeeh, M. Al-Ayyoub, M. Wardat, I. Hmeidi, and Y. Jararweh, "Topical search engine for Internet of Things," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. (AICCSA)*, vol. 2014. Doha, Qatar, 2014, pp. 829–835.

[149] X. Tang, C. Jiang, and M. Zhou, "Automatic Web service composition based on horn clauses and Petri nets," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13024–13031, 2011.

[150] J. El Hadad, M. Manouvrier, and M. Rukoz, "TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition," *IEEE Trans. Services Comput.*, vol. 3, no. 1, pp. 73–85, Jan./Mar. 2010.

[151] C.-S. Wu and I. Khoury, "Tree-based search algorithm for Web service composition in SaaS," in *Proc. 9th Int. Conf. Inf. Technol. New Gener. (ITNG)*, Las Vegas, NV, USA, 2012, pp. 132–138.

[152] S. Raghavendra *et al.*, "Survey on data storage and retrieval techniques over encrypted cloud data," *Int. J. Comput. Sci. Inf. Security*, vol. 14, no. 9, pp. 1–26, 2016.

[153] R. H. Vishwanath, T. V. Samartha, K. C. Srikantaiah, K. R. Venugopal, and L. M. Patnaik, "DRSP: Dimension reduction for similarity matching and pruning of time series data streams," *Int. J. Data Min. Knowl. Manag. Process*, vol. 6, no. 3, pp. 107–126, 2013.

[154] S. Cirani *et al.*, "A scalable and self-configuring architecture for service discovery in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 508–521, Oct. 2014.

[155] P. Leitner, W. Hummer, and S. Dustdar, "Cost-based optimization of service compositions," *IEEE Trans. Services Comput.*, vol. 6, no. 2, pp. 239–251, Apr./Jun. 2013.

[156] T. A. Butt, I. Phillips, L. Guan, and G. Oikonomou, "TRENDY: An adaptive and context-aware service discovery protocol for 6LoWPANs," in *Proc. 3rd Int. Workshop Web Things*, Newcastle upon Tyne, U.K., 2012, pp. 1–6.

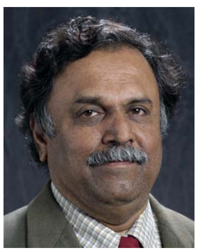
[157] *Resource Description Framework*. Accessed: Jan. 19, 2018. [Online]. Available: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[158] *Web Ontology Language*. Accessed: Jan. 19, 2018. [Online]. Available: https://www.w3.org/standards/techs/owl#w3c_all
[159] *ETH CS Dpt. Online Room Reservation System*. Accessed: Jan. 19, 2018. [Online]. Available: http://www.bookings.inf.ethz.ch
[160] C. R. Wren, Y. A. Ivanov, D. Leigh, and J. Westhues, "The MERL motion detector dataset," in *Proc. Workshop Massive Datasets (MD)*, vol. 54. Nagoya, Japan, 2007, pp. 10–14.
[161] *Bicing*. Accessed: Jan. 19, 2018. [Online]. Available: https://www.bicing.com/
[162] *NOAA Dataset*. Accessed: Jan. 19, 2018. [Online]. Available: http://tidesandcurrents.noaa.gov/gmap3
[163] *IntelLab Dataset*. Accessed: Jan. 19, 2018. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html
[164] *MavHome SmartHome Sensor Data Set*. Accessed: Jan. 19, 2018. [Online]. Available: http://ailab.wsu.edu/mavhome/index.html
[165] *Smart Santander*. Accessed: Jan. 19, 2018. [Online]. Available: http://www.smartsantander.eu/
[166] *Microdata*. Accessed: Jan. 19, 2018. [Online]. Available: https://www.w3.org/TR/microdata/
[167] *Microformat*. Accessed: Jan. 19, 2018. [Online]. Available: https://www.microformats.org/
[168] *Apache Jena TDB*. Accessed: Jan. 19, 2018. [Online]. Available: http://jena.apache.org/documentation/tdb/
[169] Commonwealth Scientific and Industrial Research Organisation, *Phenonet: Distributed Sensor Network for Phenomics Supported by High Resolution Plant Phenomics Centre*, CSIRO ICT Centre, and CSIRO Sensor and Sensor Networks TCP, Pullenvale, QLD, Australia, 2011. [Online]. Available: http://www.phenonet.com
[170] *Linked Sensor Middleware (LSM)*, Digital Enterprise Research Institute, Galway, Ireland, 2011. [Online]. Available: http://lsm.deri.ie/
[171] *Experimental Environmental Linked-Data Published by the Bureau of Meteorology*, Australian Government, Bureau of Meteorology, Melbourne, VIC, Australia, 2012. [Online]. Available: http://lab.environment.data.gov.au/
[172] *Air Quality Sensor Dataset*. Accessed: Jan. 19, 2018. [Online]. Available: https://github.com/BathHacked/documentation/wiki/Bath-Air-Quality-Sensor-Data
[173] D. Balzarotti, P. Costa, and G. P. Picco, "The LighTS tuple space framework and its customization for context-aware applications," *Int. J. Web Intell. Agent Syst.*, vol. 5, no. 2, pp. 215–231, 2007.
[174] N. Dawes, K. A. Kumar, S. Michel, K. Aberer, and M. Lehning, "Sensor metadata management and its application in collaborative environmental research," in *Proc. IEEE 4th Int. Conf. eSci. (eScience)*, 2008, pp. 143–150.
[175] L. Ferreira, L. M. Correia, D. Xavier, I. Vasconcelos, and E. R. Fledderus, "Final report on traffic estimation and services characterisation," *IST SEACORN CEC Deliverable*, vol. 34900, 2003.
[176] *Semantic Service Description Model*. Accessed: Jan. 19, 2018. [Online]. Available: http://artemis.ccsrfi.net:8686/KnowledgeManagement/ontology/IoT.est-Servicev1.9.owl
[177] *Automated Surface Observing System*. Accessed: Jan. 19, 2018. [Online]. Available: http://mesonet.agron.iastate.edu/ASOS/
[178] *OWL-S Service Retrieval Test Collection*. Accessed: Jan. 19, 2018. [Online]. Available: http://projects.semwebcentral.org/projects/owls-tc/
[179] N. Eagle, A. Pentland, and D. Lazer, "Inferring social network structure using mobile phone data," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 36, pp. 15274–15278, 2009.
[180] *Stanford Large Network Dataset Collection*. Accessed: Jan. 19, 2018. [Online]. Available: http://snap.stanford.edu/data/
[181] *CASAS Dataset*. Accessed: Jan. 19, 2018. [Online]. Available: http://ailab.wsu.edu/casas/datasets/
[182] Y. Ohsawa, N. E. Benson, and M. Yachida, "KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor," in *Proc. IEEE Int. Forum Res. Technol. Adv. Digit. Libraries (ADL)*, 1998, pp. 12–18.
[183] *Open Sourced Vulnerability Database Dataset*. Accessed: Jan. 19, 2018. [Online]. Available: http://osvdb.org/
[184] M. Botts and A. Robin, "OpenGIS sensor model language (SensorML) implementation specification," document OGC 07-000, OpenGIS Implementation Specification OGC, Wayland, MA, USA, 2007.
[185] Y. Zhou, S. De, W. Wang, K. Moessner, and M. S. Palaniswami, "Spatial indexing for data searching in mobile sensing environments," *Sensors*, vol. 17, no. 6, pp. 1427–1446, 2017.
[186] Y. Fathy, P. Barnaghi, and R. Tafazolli, "Distributed spatial indexing for the Internet of Things data management," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, Lisbon, Portugal, 2017, pp. 1246–1251.
[187] Y. Fathy, P. Barnaghi, and R. Tafazolli, "Large-scale indexing, discovery, and ranking for the Internet of Things (IoT)," *ACM Comput. Surveys*, vol. 51, no. 2, pp. 1–67, 2018.
[188] A. Akbar, A. Khan, F. Carrez, and K. Moessner, "Predictive analytics for complex IoT data streams," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1571–1582, Oct. 2017.
[189] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *arXiv preprint arXiv:1712.04301*, 2017.
[190] H. Ma and W. Liu, "Progressive search paradigm for Internet of Things," *IEEE MultiMedia*, vol. 25, no. 1, pp. 76–86, Mar. 2018.
[191] M. Alirezaie *et al.*, "An ontology-based context-aware system for smart homes: E-care@home," *Sensors*, vol. 17, no. 7, 2017, Art. no. E1586.
[192] S. Hu, H. Wang, C. She, and J. Wang, "AgOnt: Ontology for agriculture Internet of Things," in *Proc. Int. Conf. Comput. Comput. Technol. Agricult.*, Nanchang, China, 2010, pp. 131–137.
[193] D. Goel, S. Chaudhury, and H. Ghosh, "Smart water management: An ontology-driven context-aware IoT application," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, Kolkata, India, 2017, pp. 639–646.
[194] G. Bajaj, R. Agarwal, P. Singh, N. Georgantas, and V. Issarny, "A study of existing ontologies in the IoT-domain," *arXiv preprint arXiv:1707.00112*, 2017.
[195] Y. Sahni, J. Cao, S. Zhang, and L. Yang, "Edge mesh: A new paradigm to enable distributed Intelligence in Internet of Things," *IEEE Access*, vol. 5, pp. 16441–16458, 2017.
[196] X. Su *et al.*, "Transferring remote ontologies to the edge of Internet of Things systems," in *Proc. Int. Conf. Green Pervasive Cloud Comput.*, 2017, pp. 538–552.
[197] H. Rahman, R. Rahmani, and T. Kanter, "Multi-modal context-aware reasoNer (CAN) at the edge of IoT," *Procedia Comput. Sci.*, vol. 109, pp. 335–342, May 2017.

Santosh Pattar (S'17) received the Bachelor of Engineering degree in computer science from the RNS Institute of Technology, Visvesvaraya Technological University, Bengaluru, India, in 2012 and the Master of Engineering degree in bioinformatics from the University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, in 2015, where he is currently pursuing the Ph.D. degree in computer science. His current research interests include the Internet of Things, with a focus on device search and discovery techniques, cloud computing, data mining, and bioinformatics.

Rajkumar Buyya (F'15) is a Redmond Barry Distinguished Professor and the Director of the Cloud Computing and Distributed Systems Laboratory, University of Melbourne, Australia. He has authored over 625 publications and seven text books, including the book entitled *Mastering Cloud Computing* published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese, and international markets, respectively. He is one of the highly cited authors in computer science and software engineering worldwide with an H-index of 115 and over 69 100 citations. Software technologies for Cloud computing developed under his leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. He was a recipient of the Web of Science Highly Cited Researcher Award by Thomson Reuters in 2016 and 2017, and Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud computing.

K. R. Venugopal (F'16) received the Bachelor of Engineering degree from the University Visvesvaraya College of Engineering (UVCE), the master's degree in computer science and automation from the Indian Institute of Science Bangalore, and the Ph.D. degree in economics from Bangalore University and in computer science from the Indian Institute of Technology Madras. He is currently the Principal with the University Visvesvaraya College of Engineering, Bangalore University, Bengaluru. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science, and Journalism. He has authored and edited 64 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++, and Digital Circuits and Systems. He has filed 101 patents. During his three decades of service at UVCE he has over 700 research papers to his credit. His research interests include computer networks, wireless sensor networks, parallel and distributed systems, digital signal processing, and data mining. He is a ACM Distinguished Educator.



S. S. Iyengar (F'95) received the Ph.D. degree from MSU, USA in 1974. He is currently a Ryder Professor with Florida International University, USA. He was a Roy Paul Daniels Professor and the Chairman of the Computer Science Department, Louisiana State University. He heads the Wireless Sensor Networks Laboratory and the Robotics Research Laboratory with USA. He has been involved with research in High Performance Algorithms, Data Structures, Sensor Fusion, and Intelligent Systems. He has directed over 40 Ph.D. students and 100 post graduate students, many of whom are faculty of Major Universities worldwide or Scientists or Engineers at National Labs/Industries around the world. He has published over 800 research papers and has authored/co-authored six books and edited seven books. His books are published by Wiley, CRC Press, Prentice Hall, Springer-Verlag, IEEE Computer Society Press. One of his books entitled *Introduction to Parallel Algorithms* has been translated to Chinese. He is a fellow of ACM.

L. M. Patnaik (F'92) is currently a Senior Scientist with Consciousness Studies Program, National Institute of Advanced Studies, Indian Institute of Science, India. He was the Vice Chancellor with Defense Institute of Advanced Technology, Pune, India, and has been a Professor since 1986 with the Department of CSA, Indian Institute of Science, Bengaluru. During the past 35 years of his service at the Institute he has over 1150 research publications in refereed international journals and conference proceedings. His areas of research interest have been parallel and distributed computing, mobile computing, CAD, soft computing, and computational neuroscience. He was a recipient of 20 national and international awards; notable among them is the IEEE Technical Achievement Award for his significant contributions to High Performance Computing and Soft Computing. He is a fellow of all the four leading Science and Engineering Academies in India and a fellow the Academy of Science for the Developing World.