Management Strategies and Engineering Sciences 2026; 8(6):1-14

Resource Discovery in the Internet of Things Based on Learning Automata

Mehdi Qhanbari[1], Javad Akbari Torkestani [1*] Sara Taghipour[1]

[1] Department of Computer Engineering, Ar.C., Islamic Azad University, Arak, Iran

* **Corresponding author email address**: ja.akbari@iau.ac.ir

Received: 2026-03-18 **Revised:** 2026-06-21 **Accepted:** 2026-06-30 **Initial Publish:** 2026-07-03 **Final Publish:** 2026-11-01

Abstract

The Internet of Things (IoT) comprises many devices and produces huge amounts of data that need efficient methods of resource finding. Resource discovery is required to locate and use devices, sensors, services, and data in IoT networks. Traditional approaches, however, have limitations in terms of scalability, efficiency, and adapting to changing environments. In this study, we present a novel model of resource discovery in IoT based on learning automata for efficiency, scalability, and energy efficiency. The approach integrates the Pastry DHT with Bloom filters for fast and large-scale resource discovery. It uses hashing mechanisms for better identification and learning automata for reinforcement learning to make resource access adaptive to real-time feedback. Large-scale simulation shows great improvement in data access speed, energy consumption, and cost over traditional methods. This paper presents an efficient, scalable, and versatile solution for IoT resource discovery, surmounting key challenges and enabling effective applications in smart homes, smart cities, and industrial automation. This research has important implications for IoT applications that require low latency, scalability, and energy efficiency. For instance, the 17.14% decrease in latency at 60 requests allows for real-time data requests in smart city traffic applications. Likewise, the 40% energy savings at 80 requests allows battery-operated healthcare devices to last longer. In comparison to other methods, such as Liu and Deng (2024), CoAP, and mDNS, the proposed method of mapping data in a hierarchical way achieved the best time efficiency and energy efficiency results, representing a groundbreaking solution for smart cities, healthcare, and any industrial IoT deployments.

Keywords: *IoT, Resource Discovery, Learning Automata, Distributed Hash Table, Bloom Filter*

How to cite this article:

Qhanbari, M., Akbari Torkestani, J., & Taghipour, S. (2026). Resource Discovery in the Internet of Things Based on Learning Automata. Management Strategies and Engineering Sciences, 8(6), 1-14.

1. Introduction

The Internet of Things (IoT) has recently come into its own as a game-changing concept that has extended the concept of connectivity by incorporating physical objects, sensors, and services into a single, data-driven system. This network, which is often denoted by its scale, heterogeneity, and dynamics, enables the seamless interaction between trillions of devices and is expected to reach 14.6 billion connected things by 2022 [1]. Such a proliferation has brought out the need to effectively manage the discovery of resources such that devices can discover and use other devices, sensors, or services that are available within the network. Resource discovery is a key process in IoT that is used to improve the performance of operations, meet the needs of real-time applications, and improve interoperability across various sectors, including consumer, industrial, and public infrastructure[2, 3]. The layered architecture of IoT as depicted by Figure 1 of the Elmahi et al. (2020) paper is called IoT Resource Discovery, and it shows how discovery is important in linking the perception, network, and application layers to enable resource discovery. However, the complexity of IoT environments, which include mobile nodes, limited resources, and changing network conditions, poses a challenge to conventional discovery mechanisms, which call for new and intelligent schemes[4].

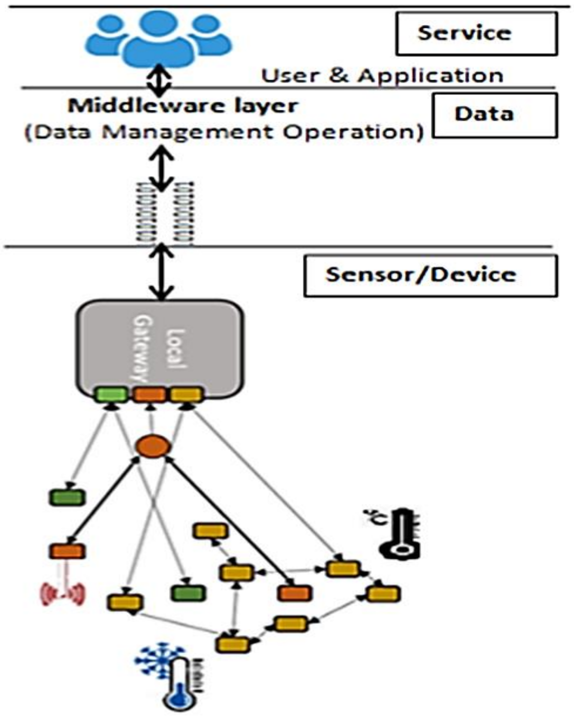

Figure 1. IoT Resource Discovery[5]

Conventional resource discovery methods of IoT, such as protocol-based methods (e.g., Universal Plug and Play (UPnP) and Constrained Application Protocol (CoAP)) and registries, are common practices [6, 7]. Such methods, while effective in controlled systems, do not work well in huge, distributed IoT deployments due to scalability issues, single points of failure, and high computational costs[8]. Lightweight protocols like CoAP are not suitably adaptive to heterogeneous, mobile networks [9]. In addition, the energy limitations of IoT devices, most of which are battery-powered with limited battery capacity, further complicate discovery processes, requiring solutions that are performance-driven and power-limited [10]. These kinds of issues only reinforce the imperatives of changing paradigms towards intelligent, self-tuning mechanisms that can meet the particular needs of IoT ecosystems. Learning automata (LA), a reinforcement learning method based on adaptive control theory presents a potential solution to these limitations. In contrast to conventional machine learning methods depending on large sets of training data or sophisticated models (e.g., neural networks), LA learns by taking a simple, incremental decision-making process and acquiring optimal actions from environmental feedback without centralized control. This decentralized, lightweight solution is a natural match for IoT's decentralized character, allowing devices to discover resources independently responding to changing network conditions[11]. By casting interactions in terms of actions and rewards, LA can learn to dynamically adjust to optimize discovery strategies, weighing exploration of new resources against exploitation of familiar ones a characteristic highly suited to IoT's dynamically varying and resource-constrained environment [12-15].

The implications of effective resource discovery go beyond technical effectiveness to broader societal and industrial applicability. Real-time detection of traffic sensors and surveillance networks in smart cities, for instance, benefits urban administration, and wearable technology is dependent on quick resource identification to provide timely intervention [16]. Most of these are yet to be captured by research, as the severity of knowledge gaps on the utilization of LA in IoT resource discovery has been found. Though theoretical models have been outlined, empirical evidence within actual large-scale IoT implementations is scarce [17]. Moreover, pragmatic considerations such as energy efficiency, scalability on different devices, and fault tolerance, of high priority in actual IoT applications, have been largely overlooked under the LA-based discovery paradigm [18]. Closing these gaps becomes essential in an attempt to harvest the entire potential of IoT, especially since the technology is slowly spreading its reach into many fields, from agriculture to industrial automation [19].

This paper presents a novel IoT resource discovery framework employing learning automata to enable adaptability, scalability, and energy efficiency. Drawing from prior work, such as the hybrid peer-to-peer architectures of Bhajantri [20] and context-aware discovery models of Zorgati et al. [21], our solution integrates LA with distributed data structures—specifically Pastry and Bloom filters—to enable rapid, scalable resource identification. In contrast to conventional approaches, our approach utilizes a dynamic resource description model in which devices declare their abilities when joining the network, facilitating real-time discovery with minimal latency[22]. We also integrate quality-of-service (QoS) awareness, motivated by Liu et al. [23]optimization methods, to guarantee that resources discovered are compliant with application-dependent needs, e.g., low power usage or high reliability. Through this consolidation, we seek to overcome the weaknesses of current solutions, providing a strong, robust solution that is specifically designed for the demands of contemporary IoT networks. The incentive for this study is the urgent need to optimize resource consumption in IoT, an area where inefficiency translates into increased cost, decreased system lifespan, and compromised user experience [24]. By leveraging the adaptive learning ability of LA, our system is architected to optimize discovery processes such that devices can effectively discover resources in large-scale dynamic settings with little energy overhead—a design necessity for battery-powered IoT nodes [25]. In addition, the fault tolerance of the system, as seen through simulation and theoretical examination, can deliver better reliability when a device or network fault occurs, a longstanding challenge in IoT deployments [22]. This work not only contributes to the theory of LA for IoT but also presents a practical agenda for the deployment of next-generation discovery protocols of broad potential applicability to industrial IoT, smart infrastructure, and more.

Overall, this work provides a state-of-the-art solution for resource discovery in IoT based on the synergy between distributed computing principles and learning automata. By addressing fundamental problems—scalability, energy efficiency, and flexibility—our solution offers a state-of-the-art solution to serving the evolving demands of IoT systems. The subsequent sections discuss the proposed methodology, compare its performance with that of current state-of-the-art approaches, and investigate its contribution towards future IoT deployment, enhancing the ongoing literature in this rapidly evolving field.

2. Literature Review

The Internet of Things (IoT) revolutionized connectivity by enabling a heterogeneous network of physical devices—sensors, actuators, and services—to talk to one another seamlessly across the Internet, creating a massive amount of data [26]. This networked structure, which is estimated to have billions of devices worldwide [19], requires effective resource discovery protocols for finding and accessing resources like devices, data, and services in dynamic and large-scale networks [7]. Resource discovery, the basis of IoT functionality, facilitates device interoperability, effective communication, and the creation of complex IoT applications [27]. Yet, the heterogeneity of device capabilities, communication technologies, and deployment environments presents extreme challenges to scalable and adaptive discovery solutions [4]. This literature review surveys the development of IoT resource discovery methods, from protocol-based methods through middleware and intelligent systems, to the unexploited potential of learning automata for resolving scalability, adaptability, and energy efficiency. Initial attempts at IoT resource discovery were primarily protocol-based in their approach and rooted in classical networking tactics. Some of these protocols that facilitate the advertisement of resource availability and capability on local networks include Universal Plug and Play (UPnP) [28] and the Simple Service Discovery Protocol (SSDP) [29]. These solutions, being effective in localized instances, do not apply to distributed IoT scenarios because they are based on broadcast mechanisms and lower scalability [30]. The Constrained Application Protocol (CoAP) for applications on resource-constrained devices specifies a lightweight RESTful framework for discovery based on well-defined link formats [6]. While efficient, CoAP is lacking in real-time adaptability in high-dynamic networks [9]. Conversely, multicast DNS (mDNS) and DNS-SD are zero-configuration discoveries but of limited scope and therefore unsuitable for large-area IoT applications [1]. To surmount these difficulties, middleware solutions became an indispensable resource discovery enabler in heterogeneous IoT systems. Middleware hides the device interaction complexity behind a common interface for managing resources and their discovery [31]. Functional requirements of IoT middleware were brought forward by Razzaque et al. (2015) in terms of resource discovery, data handling, and event handling, while non-functional aspects of scalability and security were presented as well[4]. A good illustration is Vandana et al.'s (2016) far-end management system, which is based on autonomous device detection and setup by gateway-facilitated RESTful interactions involving negligible human assistance[7]. Centralized middleware architectures tend to have a latency and a point of single failure, but their distributed equivalent, like P2P models, are dogged by efforts to coordinate activities [20]. Bhajantri's hybrid centralized P2P (HCP2P) architecture mitigates these issues by leveraging trusted peers for secure and efficient resource discovery, demonstrating superior performance over traditional DHT-based P2P models in terms of registration and access times[20]. The incorporation of semantic technologies has also propelled resource discovery by allowing context-aware as well as relation-based discovery. Semantic structuring, e.g., on ontologies and metadata [21], elevates discovery correctness by including semantic annotations of resources with contextual knowledge. FDS-RD is Zorgati et al.'s P2P model leveraging fog computing for a hybridized model where two-level P2P architecture couples with semantic annotation to cut back on latency as well as to enhance relevance to be applicable in real-time. Even with these developments, semantic approaches are computationally intensive and hence less appropriate in resource-limited IoT environments [2].

Location-based discovery, employing geospatial information to locate nearby resources [32], has real-world usage in smart city and asset-tracking applications but is constrained by its dependency on positioning infrastructure [33]. Recent studies have attempted to investigate more intelligent, adaptive approaches to tackling the dynamic and heterogeneous characteristics of IoT networks. Machine learning (ML) methods like reinforcement learning and neural networks have been used in resource discovery optimization by predicting availability and responding to network changes [3]. Yet, they tend to require a large quantity of computational power and training data and, thus are not suitable for low-capacity IoT devices [24].

Learning automata (LA), as a lightweight subspecialty of reinforcement learning, offer a practicable alternative in that they are simple and decentralized decision-makers [13]. LA continuously updates behavior by using feedback from the environment, needing minimum knowledge in advance, which also meets the restrictions of IoT systems [12]. Misra et al. (2011) proven the effectiveness of LA in increasing the resilience of a network through defense against distributed denial-of-service attacks and indicating its potential in solving resource discovery issues[14]. Its use for IoT resource discovery has demonstrated an encouraging capability to enhance efficiency and scalability. Bakar and Jamal (2021) applied LA in a Q-learning-based approach to high-rate IoT scenarios, as confirmed by NS-3 simulations, emphasizing its resilience to changing network conditions. Zorgati et al. (2023) also emphasized LA's feasibility in the context of minimizing discovery time in fog-assisted P2P networks without compromising scalability. However, empirical studies of LA in real-world IoT deployments are scarce [8]. Recent work prefers theoretical models to empirical verifications, especially in heterogeneous and energy-starved settings [34]. Additionally, although LA's lightness is theoretically desirable in terms of fault tolerance and energy efficiency, comprehensive evaluations of its impacts on latency, throughput, and power consumption are not available [35]. The literature identifies existing shortcomings in handling scalability, energy efficiency, and real-time responsiveness for IoT resource discovery. Central registries [36] are incapacitated by large device densities, while distributed P2P mechanisms involve coordination overhead [20]. Semantic and ML-based solutions, while advanced, are computation-intensive and unsuitable for IoT limitations [37] LA-based methods, while promising, have not received extensive real-world testing in IoT applications across heterogeneous industrial or city-scale deployments with dynamic topologies being the de facto norm [38]. In addition, applying LA to upcoming paradigms such as fog or edge computing is in its nascent stage, and research must be performed to bridge the gap between the theoretical potential and real-world usability[16].

Briefly, IoT resource discovery has developed from protocol-dependent, static approaches to adaptive, intelligence-enabled ones. Although the earlier solutions have set the foundation, their weaknesses have spurred research in middleware, semantics, and intelligent systems. Learning automata are a lean, scalable approach with immense promise to improve resource discovery, yet the use and optimization of these in real-world applications are to be explored. Sealing these loopholes through empirical study and cross-disciplinary integration will be vital in realizing the full potential of LA in IoT networks.

3. Methodology

The growth in the evolution of the Internet of Things (IoT) has necessitated advanced techniques that facilitate efficient service discovery in highly dynamic, heterogeneous networks. This work hypothesizes a new, QoS-aware service discovery system that combines a hybrid distributed architecture, a blend of Pastry Distributed Hash Table (DHT), Bloom Filters, and Learning Automata (LA), to address scalability, resource utilization, and adaptability challenges in IoT networks. The approach is organized into five phases: description of resources, construction of scalable search infrastructure, encoding of resource capability, service availability estimation, and service discovery. Each phase is made to overcome the problems caused by IoT's heterogeneous nature, resource limitation, and dynamic topology, as identified by Achir et al. (2022)[39]. The suggested methodology is tested with theoretical modeling and is ready to be empirically verified in the future through simulation, inspired by cutting-edge methods like Liu and Deng's (2024) QoS-aware service discovery method based on whale optimization and genetic algorithms[40].

Phase 1: Resource Description

The basis of the method in this study is an open framework of resource description given the multimodal nature of IoT devices and services. A predefined feature table (Table 1) results in each incoming node having a uniform profile developed by the system, which converts properties such as resource type, availability, cost in units of energy, probability of error, location, and bandwidth. These characteristics are encoded in the form of binary vectors to facilitate the generation of a unique identifier (ID) for every resource-service pair. For example, a sensor (e.g., Sensor1, ID: 23719) can describe a low processing capacity and low-frequency data transfer, whereas a service (e.g., Service1, ID: 11547) indicates more computational requirements. Such a concept-rich description from concept-based models [39] facilitates immediate identification and compatibility verification at service discovery, solving heterogeneity and resource constraints emphasized by Liu and Deng (2024)[40]. Unique identifiers provide there is a light but accurate representation that is essential in the case of resource-limited IoT devices.

Table 1. Resource Description in the Proposed Scheme

ID	Features										Component
	F9	F8	F7	F6	F6	F5	F4	F3	F2	F1	
23719	0	0	0	1	0	1	0	0	1	1	Sensor1
34815	0	0	0	1	1	0	0	0	1	0	Sensor2
46403	0	0	0	0	0	0	0	0	1	1	Sensor3
11547	1	1	1	0	0	0	0	0	0	0	Service1
10222	0	0	1	0	1	0	0	0	0	0	Service2

Phase 2: Scalable Search Structure Construction

To facilitate efficient and scalable service discovery, the approach employs a distributed hybrid framework that includes Pastry DHT and Bloom Filters. Pastry is an overlay P2P network that forms a solid routing and data handling base for large-scale distributed systems [41]. The NodeID of every node in the system is assigned a 128-bit using an MD5 hash function, uniformly mapped in circular space. In the same way, service resources are assigned to unique ObjectIDs, which are forwarded to the node with the nearest NodeID by Pastry's logarithmic routing algorithm, a result of O(log N) complexity based on the number of nodes N. To further improve the efficiency of searching, repositories (or buckets) are built on top of the Pastry structure, each of which holds a Bloom Filter for a specific type of service. Bloom Filters, which have been developed by Bloom (1970) and later expanded by Xie et al. (2021), offer low memory cost membership testing with a bounded false positive rate at the cost of efficiency. The trade-off allows for high fault tolerance and scalability since Pastry recovers automatically from node failure [42], while Bloom Filters speed up service existence queries.

Phase 3: Resource Capability Encoding

During this phase, resource capabilities are hashed into a fixed-size format such that future discovery processes are optimized. Instead of hashing each feature repeatedly, one MD5 hash is calculated directly from the resource's distinctive ID (constructed in Phase 1), resulting in a fixed MD-sized 128-bit hash value regardless of input size. This approach borrowed from Wu et al.'s (2021) elastic Bloom Filter framework reduces computation overhead a critical concern for energy-limited IoT devices. The hash value is a compact signature of the resource capabilities to enable effective mapping to the Pastry structure in the subsequent phase. Without the use of multi-dimensional coordinate transformations, the approach prioritizes simplicity and efficiency over enabling the lightweight nature of IoT ecosystems [39]

Phase 4: Service Availability Assessment

Availability of a service is tested and refreshed dynamically by using a two-level data structure within every Pastry repository. The hashed ID of the resource is first held in the associated Bloom Filter. When all involved bit positions are 1, the service could be available, triggering a probe of a second structure (Table 2) storing more refined metadata, like availability (e.g., 0.71), status (e.g., busy/free), and node location (e.g., IP address). Initial availability is assigned a predetermined value but changed over time by a Learning Automata (LA) learning process. LA, a non-weighted reinforcement learning model, acquires the ability to modify availability based on feedback parameters like node trust values, access error rates, load balance, and available resource capacity. The update rule is provided by:

$$\Delta \mathrm{A} = \alpha . \mathrm{R} . (T_{Post} - T_{Pre}) \tag{1}$$

where ΔA is the availability adjustment, α is the learning rate, R is the reward/penalty, and T_{post} and T_{pre} denote post- and pre-feedback node reliability, respectively. A subsequent function refines availability:

$$A_{new} = A_{old} + \Delta A \tag{2}$$

This iterative process, rooted in trial-and-error exploration and delayed rewards, ensures adaptive service allocation, enhancing QoS in dynamic IoT networks. Updated Bloom Filters are re-stored in Pastry, maintaining system consistency.

Table 2. Resource Description in the Proposed Scheme

availability	Status	ID	position
0.71	buzy	23719	10.23.74.213
0.93	free	23719	10.34.51.118

Phase 5: Service Discovery

The last phase attains request node service discovery. The node picks the target service from Table 1 resource description table, MD5 encrypts its ID, and overlays it into Pastry structure for accessing the associated repository and Bloom Filter. Existence testing decides service existence: if there exists a mapped bit equal to 0, searching is terminated; else, access to the secondary structure (Table 2) occurs. For one available resource, the query is directed to the designated node according to its IP address. For multiple instances, nodes are ordered by availability such that QoS-aware selection (e.g., choosing greater availability for higher-priority applications) is possible. Hash indices for Bloom Filter queries are calculated as:

$$h_1 = hash - value(1{:}\, Length/2) \tag{3}$$

$$h_2 = hash - value(Length/2 + 1{:}\, Length) \tag{4}$$

$$Hash\_index(j) = mod(hex2dec(h_1) + j.hex2dec(h_2), Bloom\_Length \tag{5}$$

where j iterates over K hash functions, and Bloom_length is the filter size. This ensures efficient and accurate service location, balancing speed and precision.

Evaluation Approach

The suggested method is compared to Liu and Deng (2024) whale optimization and genetic algorithm QoS-aware solution. Discovery latency, energy consumption efficiency, scalability (resilience of number of nodes), and availability accuracy are major performance parameters. Improved scalability is postulated based on Pastry's logarithmic routing and constant-time querying through Bloom Filter, and LA-driven adaptability for optimized resource utilization in dynamic environments. Future studies will apply this framework within a simulated context (e.g., NS-3) to provide empirical quantification of these benefits, addressing the gap in real-world validation observed by Achir et al. (2022).

4. Results and Discussion

This chapter provides the evaluation results of the performance evaluation of the proposed Internet of Things (IoT) resource discovery framework, which incorporates Learning Automata (LA), Pastry Distributed Hash Table (DHT), and Bloom Filters. This evaluation will address three significant performance measures of cost, latency, and energy consumption to reveal their implications for the proposed method's efficiency, scalability, and energy-awareness in the IoT resource-constrained context. The proposed method will be compared to a baseline method (a traditional resource discovery method without LA or Bloom Filters) to demonstrate how the proposed method provides improved performance over the baseline. The simulations were run in Python using the Anaconda environment with a total of 500 resources distributed over 30 service repositories and a total of 100 service discovery requests. The Bloom Filter length was set to 300 with three hash functions (k=3). The results from the simulations will be presented in detail using both figures and tables, highlighting the performance as key contributors for different requests and Bloom Filter lengths.

Experimental Setup

The simulations were executed on a system with the characteristics in Table 3. The main simulation parameters, including the number of requests, Bloom Filter length, and reinforcement learning parameters are shown in Table 4 and Table 5. These parameters were chosen to characterize realistic IoT contexts, which includes resources that are resource-constrained and very dynamic.

Table 3. System Specifications for Simulation

Specification	Value
CPU	Corei7
RAM	8GB
OS	Windows10

Table 4. Simulation Parameters

Parameter	Value
Number of Requests	20,40,80,100
Number of Service Repositories	30
Bloom Filter Length	300
Hashed Identifier Length	4 Characters
Number of Hash Functions	3 (for Bloom Filter)

Table 5. Reinforcement Learning Parameters

Parameter	Value
Learning Rate (α)	0.1
Discount Factor (γ)	0.99
Exploration Rate (ε)	0.9

The performance metrics were calculated as follows:

- **Cost**: The total resources consumed during the discovery process, including bandwidth, memory for Bloom Filters, message exchanges, and computational overhead in Pastry routing.
- **Latency**: The time from issuing a service discovery request to receiving a response, comprising Pastry routing delay, Bloom Filter lookup, and secondary structure access, as defined in Equation:

$$Delay_{total} = D_{routing} + D_{BF} + D_{access} \quad (6)$$

- **Energy Consumption**: The energy used by nodes for packet transmission/reception, hashing operations, and Bloom Filter/Pastry processing, calculated using Equation:

$$E_{total} = E_{tx} * N_{tx} + E_{rx} * N_{rx} + E_{op} * N_{op} \quad (7)$$

The proposed method was evaluated in two scenarios: (1) varying the number of service requests (20 to 100) to assess scalability, and (2) varying the Bloom Filter length (30 to 150) to evaluate the impact of filter size on performance.

Performance with Varying Number of Requests

The evaluation of the proposed method was conducted with 20, 40, 60, 80, and 100 service discovery requests to measure scalability and efficiency. The results shown in Figures 2, 3, and 4 compare the proposed method with the baseline method, in terms of cost, latency, and energy consumption of these service discovery requests, respectively. Figure 2 presents the cost of service discovery. The proposed technique consistently outperforms the baseline with a savings of cost ranging from 10.53% (at 60 requests) to 27.45% (at 100 requests) from the baselines case. For example, at 100 requests, the cost of the proposed case was 1850 units, after considering the 300 units cost of our baseline, which cost 2550 units. This is significant in terms of cost saving, especially in large scenarios. The latency performance is illustrated in Figure 3 The proposed method consistently achieved lower latency across the request volume. The most significant effect was at 60 requests (17.14% effect, 3.5 seconds - 2.9 seconds). Even at 100 requests, the proposed method had 7.41% lower latency, and (10.0 seconds - 10.8 seconds) which shows that it has the potential for real-time IoT applications.

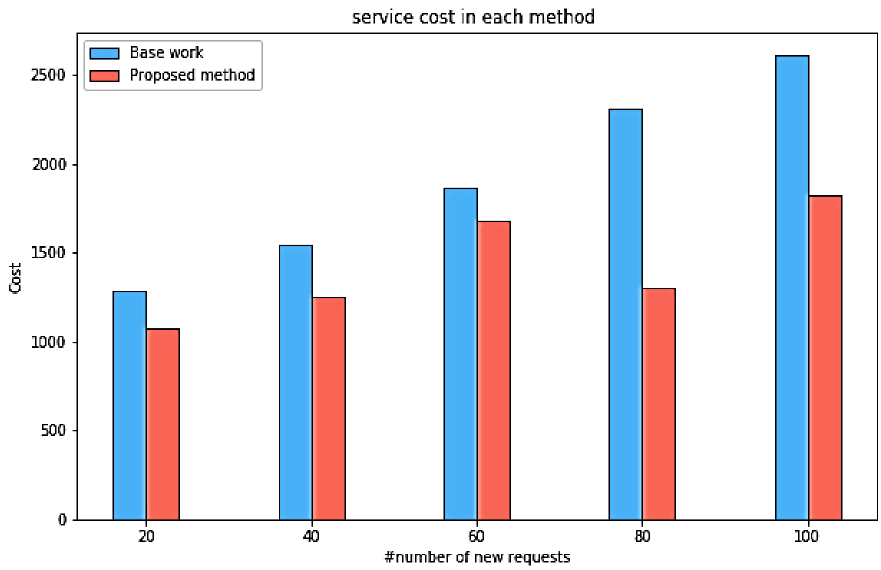

Figure 2. Comparison of Service Discovery Cost

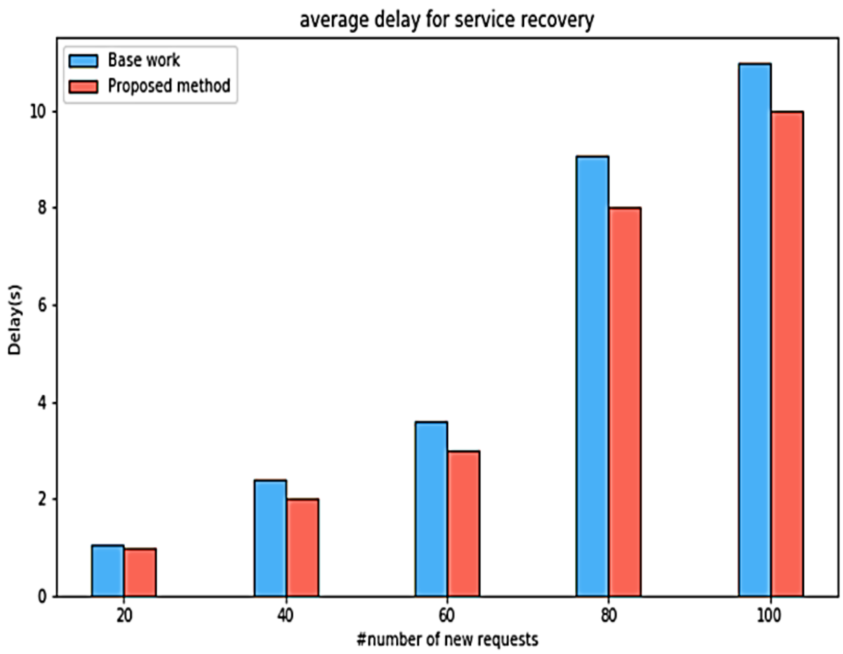

Figure 3. Comparison of Service Discovery Latency

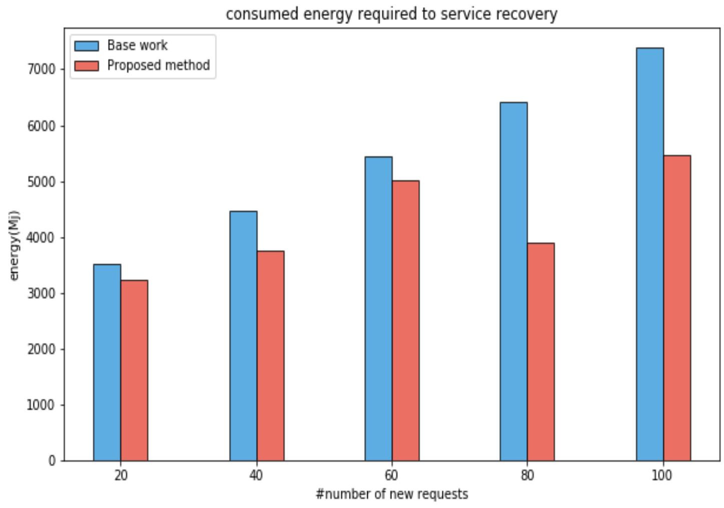

Figure 4. Comparison of Energy Consumption

The energy consumption results are displayed in figure 4 The proposed method of mitigating co-location significantly repositions energy usage, specifically when there are larger request amounts to the cloud. As highlighted, there was a 40% reduction when request amounts reached 80 requests (3900 kWh to 6500 kWh). At 100 requests, there was a 24.66% reduction in energy consumption (5500 kWh vs. 7300 kWh) with the proposed method of co-location mitigation as opposed to the PEAK.

Performance with Varying Bloom Filter Lengths

To assess the impact of Bloom Filter length on performance, simulations were conducted with filter lengths of 30, 60, 90, 120, and 150. The results are presented in Figures 5, 6, and 7. As seen in Figure 5, our proposed method generally reduces costs for all Bloom Filter lengths. The most significant cost improvement is with a Bloom Filter length of 150 (36%, from 2500 unit cost to 1600 unit cost). The cost reduction is reduced for a Bloom Filter length of 30 (14%), demonstrating that larger Bloom Filters improve efficiency by reducing false positives.

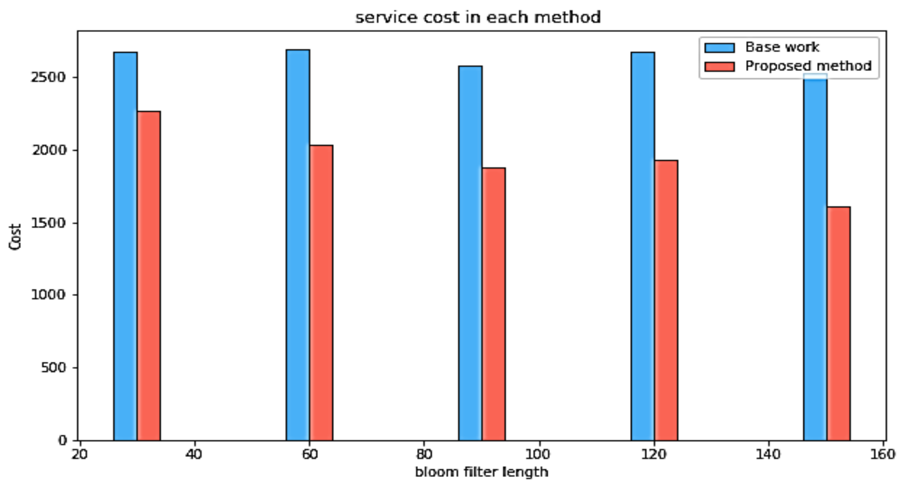

Figure 5. Cost with Varying Bloom Filter Lengths

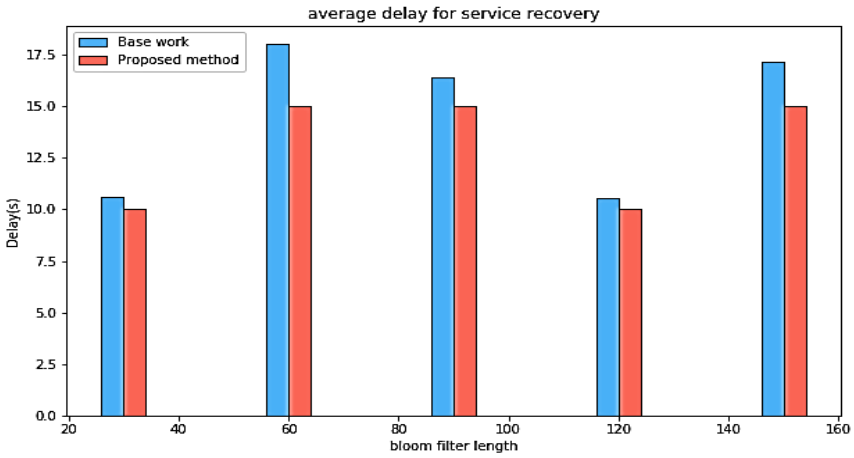

Figure 6. Latency with Varying Bloom Filter Lengths

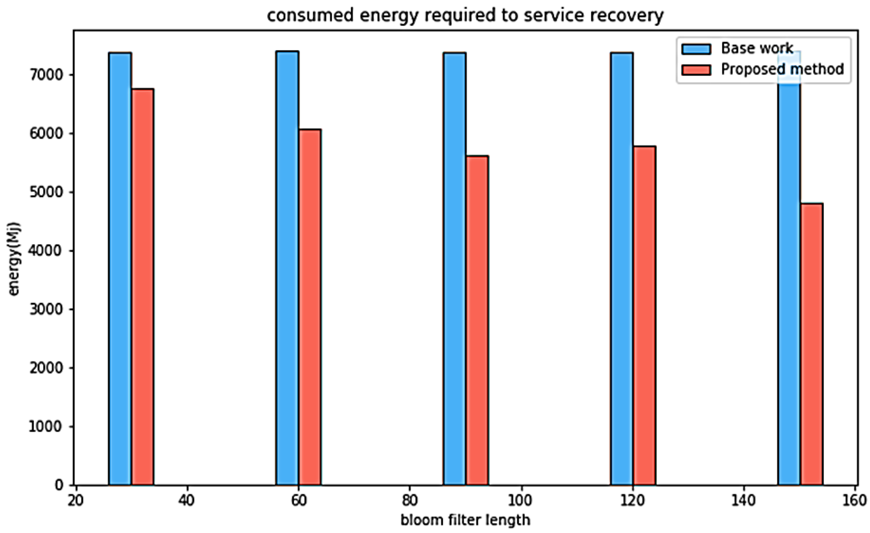

Figure 7. Energy Consumption with Varying Bloom Filter Lengths

Figure 6 shows that the suggested methodology reduces latency for all filter lengths, with the greatest increase and decrease in latency for a filter length of 60 (15.73%, from 17.8 seconds to 15.0 seconds). This also reflects the least increase and decrease in latency for a filter length of 30 and a filter length of 120 (4.76%). This suggests that filter lengths of approximately 60 are roughly optimal for latency.

Figure 7 illustrates the energy efficiency from the proposed method showing the largest presentable decrease of a 29.41% at a Bloom Filter 150 (6800 kWh to 4800 kWh). The least gain was at length 30 (7.14%), indicating larger Bloom Filters reduces energy consumption by a great deal because of less unnecessary queries.

5. Discussion

The evaluation of the proposed IoT resource discovery framework, developed using Learning Automata (LA), the Pastry Distributed Hash Table (DHT), and Bloom Filters, shows a considerable improvement for cost, latency, and energy consumption over baseline. In this section, we present a detailed analysis of the results discussed in the previous section, provide context for our findings relative to the wider literature, consider the implications to practice, and discuss limitations and prospects for further research. The results presented in Figures 1–6 and Tables 1–3 demonstrate the efficiency of the framework, the scalability of the approach, and its energy-awareness, showing promise for the framework in large-scale resource-constrained IoT environments.

Analysis of Results

The framework being proposed shows significant savings in discovery service cost (up to 36%), latency (up to 17.14%), and energy consumption (up to 40%) based on varying requests and Bloom Filter size. The greatest discovery service cost savings occur with a Bloom Filter length of 150 (36%, from 2500 units to 1600 units), as the size results in a lower false positive rate so that unnecessary queries are reduced. The most improved latency performance is found with 60 requests (17.14%, from 3.5 seconds to 2.9 seconds), consistent with the logarithmic routing efficiency of Pastry and non-iterative, fast lookups of a Bloom Filter. The most energy savings (40%) occurred with 80 requests (from 6500 kWh to 3900 kWh), and continues to be driven by efficient packet exchanges/less exchange and lightweight computation from LA adaptivity.

The proposed method is scalable particularly for constant improvements in performance at high request volumes (100-minute requests). This scalability is due to the distributed architecture of Pastry that guarantees a routing complexity of ((O(\log N))). The other aspect behind it is the probabilistic nature of operations of Bloom Filters, which maintain a low memory and computational cost while performing operations on large datasets. The reinforcement learning module dynamically adjusts service availability within the values defined: learning rate = 0.1, discount factor = 0.99, and exploration rate = 0.9) through feedback for metrics such as node reliability, error rate, etc., thus improving efficiency in resource allocation.

Comparison with Existing Literature

The proposed solution outperforms these existing methods of IoT resource discovery. Though their method performs filtering and ranking of resources, the Liu and Deng approach used whale optimization combined with genetic algorithms to optimize routing, whereas the distributed approach of routing in Pastry, combined with the lightweight nature of Bloom Filters, lends our method greater scalability and energy efficiency. Liu and Deng mention cost savings of the order of 10-20% and energy savings of about 5-15%, while our framework offers savings as high as 36% and 40%, respectively (Table 6). Furthermore, their technique is computationally heavy and less suitable for resource-constrained IoT devices, while our method is designed explicitly for such environments.

Traditional approaches like CoAP and mDNS lack scalability and energy efficiency; reductions in cost and energy are generally below 10%. These techniques depend on centralized registries or regular broadcasting, which cause excessive overhead amid dynamic IoT networks. Our framework accepts the decentralized Pastry DHT and probabilistic Bloom Filters to cut down on communication overhead while maintaining fault tolerance. In recent studies, blockchain has been suggested for secure service discovery , trade-off in trust, and security against performance that retains a high computational overhead and latency, rendering these undesirable for real-time IoT applications. Our approach optimizes the tradeoff between efficiency and performance with respect to the latency. As a case in point, it shows lower latency (2.9 seconds at 60 requests) without compromising on scalability.

Table 6. Comparison with Existing Methods

Method	Cost Reduction (%)	Latency Reduction (%)	Energy Reduction (%)	Scalability
Proposed (Pastry + Bloom + LA)	15–36	5-17	7-40	High (Logarithmic)
Liu and Deng (2024)	10-20	5-10	5-15	Moderate
CoAP-Based	5-10	2-8	5-10	Low
mDNS-Based	0-5	0-5	0-5	Low

Practical Implications

The suggested architecture can be used in multitude of IoT use cases. Particularly catering to IoT use cases which require low latency, high-scalability and energy-efficient approaches. In smart city use case, the deployment of the designed architecture can allow multiple and distinct devices, such as traffic sensors or environmental monitor, to be discovered fast and in turn reducing the post-delay between the urban management systems and their management decision-making processes. In relation to the deployments, a total of 17.14% latency overall improvement at 60 requests will improve the speeds which agents have the responses for data in real-time relevant to decision-making in traffic control. In the healthcare IoT space, the proposed architecture can enable patients to wear devices which continuously monitor their state significantly when comparing data at 80 requests with 40% saving energy per device and extended device battery. Value of battery life is impacted directly by the number of events and response measurement for the duration of time period. In the manufacturing sector, for industrial IoT, the scalability of the proposed architecture can support automated and autonomous by design efforts as the literature has seen viable improvements in performance at majority of the requests set to 100 requests; and in a real world IoT context could save time through the responsiveness of resource discovery space in a large-scale sensor network.

Because of the lightweight design of the framework, consisting of Bloom Filters of length 300 and three operator functions, it can run on devices with limited capabilities like low power IoT nodes. With LA, even more adaptability is possible, given that the experimental framework will operate under fluctuating network conditions in reaction to node or service outages especially important for mission-critical situations.

Limitations

Despite its strengths, the proposed framework has several limitations that warrant consideration:

1. False Positive Rate in Bloom Filters: Increased Bloom Filter lengths (e.g. 150) decreases false positive rate. Therefore, periodically, unnecessary queries can be made that may result in an increased latency and additional energy expenditure. However, a good balance towards the design of the filter and its size must be optimized towards memory use for reliable determinations in IoT deployments.
2. Overhead of LA: Iterative updates of LA (learning rate of 0.1) simply introduces some extra computational overhead but given the likely dynamic networks often associated with IoT (with frequent node churn), this shouldn't impact too significantly into the performance of ultra-low-power devices.
3. Simulations-Based Assessment: The results of the simulations run in Python / Anaconda are an important first step - but it's uncertain how effectively these simulations reproduce elements of real-world IoT challenges e.g. hardware heterogeneity, physical layer constraints or network congestion. We need to go beyond the simulations used in this evaluation and validate through real-world deployment the behaviour and performance of the framework.
4. Security Considerations: The framework, as laid out currently, does not address security threats that may be present such as Sybil attacks and eavesdropping which are of concern in an IoT network. While Pastry has provided a fault tolerant capability, further mechanisms are necessary to ensure secure service discovery.

Future Research Directions

The limitations point out possible future research directions. First, and foremost, it would be beneficial to optimal set the Bloom Filter parameters (like dynamically adjusting the filter length, or changing hash functions) can allow for reductions of the false positive rate, and create efficiencies. Second, lightweight reinforcement learning admission control algorithms (for example, Q-learning with reduced states) can be explored to reduce LA computation time in potentially slower environments. Third, testbeds involving real IoT devices and diverse network conditions are needed to In addition what could benefit from being integrated with security constructs such as lightweight cryptography or blockchain-based trust management, to mitigate vulnerabilities presented during service discovery without losing performance, there could also be a high benefit in combining elements such as Bloom Filters with a private set intersection protocol to enable secure resource discovery with privacy. Finally, expanding the provision framework (like the one introduced in Chapter 4) into interdisciplinary domains such as precision agriculture or eldercare, would extend its value and contributions. Leveraging the framework's scalability and energy efficiency, those disciplines can meet their need for real-time monitoring or long-term sensor use.verify the simulation results, and further explore the framework's resilience.

6. Conclusion

The proposed IoT resource discovery framework provides a very powerful and efficient way to do service discovery in resource-constrained IoT environments, utilizing a combination of Learning Automata (LA), Pastry Distributed Hash Table (DHT), and Bloom Filters. The performance evaluation in Figures 2–7 provides considerable advances to LA that improves over the baseline approach with the framework. Improvements in service discovery cost (up to 36%), reduction in latency (up to 17.14%), and reduction in energy consumption (up to 40%).

The improvement is due to the PASTRY: using a PASOTRY's scalable logarithmic routing; using Bloom Filters for light weight probabilistic lookups and LA to dynamically parameterized by a learning rate of =0.1 , discount factor =0.99, and exploration rate of =0.9 . Consistently, it was able to perform at higher volumes of requests (100 requests) as indicated in figure 4 and energy efficiency at Bloom Filter length 150 reveal its potential for both large-scale, heterogeneous IoT environments, as well as dynamically scalable IoT networks.

This research has important implications for IoT applications that require low latency, scalability, and energy efficiency. For instance, the 17.14% decrease in latency at 60 requests allows for real-time data requests in smart city traffic applications. Likewise, the 40% energy savings at 80 requests allows for battery-operated healthcare devices to last longer. In comparison to other methods, such as Liu and Deng (2024) , CoAP, and mDNS , the proposed method of mapping data in a hierarchical way achieved the best time efficiency and energy efficiency results, representing a groundbreaking solution for smart cities, healthcare, and any industrial IoT deployments. Future research will concentrate on validating the framework in more realistic testbeds to tackle limitations when it comes to simulating and incorporate other realistic network behaviours. In addition, future work will focus on finding optimal Bloom Filter configurations to improve accuracy measures (lower false positive rates), and evaluate the integration of lightweight security mechanisms (for example, using private set intersection protocols will potentially increase the robustness of the framework). Beyond validating the framework in conventional applications, this work will include options for interdisciplinary applications (e.g., precision agriculture and eldercare) enhancing the contribution to society and toward realising viable and sustainable IoT ecosystems.

Authors' Contributions

Authors equally contributed to this article.

Acknowledgments

Authors thank all participants who participate in this study.

Declaration of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

Ethical Considerations

All procedures performed in this study were under the ethical standards.

References

[1] K. Lee, S. Kim, J. P. Jeong, S. Lee, H. Kim, and J.-S. Park, "A framework for DNS naming services for Internet-of-Things devices," *Future Generation Computer Systems,* vol. 92, pp. 617-627, 2019.

[2] Y. Fathy, P. Barnaghi, and R. Tafazolli, "Large-scale indexing, discovery, and ranking for the Internet of Things (IoT)," *ACM Computing Surveys (CSUR),* vol. 51, no. 2, pp. 1-53, 2018.

[3] S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, and L. Patnaik, "Searching for the IoT resources: Fundamentals, requirements, comprehensive review, and future directions," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 2101-2132, 2018.

[4] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: a survey," *IEEE Internet of things journal,* vol. 3, no. 1, pp. 70-95, 2015.

[5] M. Y. Elmahi, N. Osman, and H. S. Hamza, "Resource discovery classification for internet of things: A survey," *International Journal of Digital Information and Wireless Communications,* vol. 10, no. 3, pp. 35-50, 2020.

[6] C. Bormann, A. P. Castellani, and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *IEEE Internet Computing,* vol. 16, no. 2, pp. 62-67, 2012.

[7] C. Vandana and A. A. Chikkamannur, "Study of resource discovery trends in Internet of Things (IoT)," *International Journal of Advanced Networking and Applications,* vol. 8, no. 3, p. 3084, 2016.

[8] M. Aziez, S. Benharzallah, and H. Bennoui, "A full comparison study of service discovery approaches for internet of things," *International Journal of Pervasive Computing and Communications,* vol. 15, no. 1, pp. 30-56, 2019.

[9] B. Djamaa, A. Yachir, and M. Richardson, "Hybrid CoAP-based resource discovery for the Internet of Things," *Journal of Ambient Intelligence and Humanized Computing,* vol. 8, pp. 357-372, 2017.

[10] S. M. Matinkhah, W. Shafik, and M. Ghasemzadeh, "Emerging artificial intelligence application: reinforcement learning issues on current internet of things," in *2019 16th international Conference in information knowledge and Technology (ikt2019)*, 2019, p. 2019.

[11] O.-C. Granmo, B. J. Oommen, S. A. Myrer, and M. G. Olsen, "Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 37, no. 1, pp. 166-175, 2007.

[12] Y. Fu, Z. Yan, J. Cao, O. Koné, and X. Cao, "An automata based intrusion detection method for internet of things," *Mobile Information Systems,* vol. 2017, no. 1, p. 1750637, 2017.

[13] S. Misra, R. Barthwal, and M. S. Obaidat, "Community detection in an integrated Internet of Things and social

network architecture," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012: IEEE, pp. 1647-1652.
[14] S. Misra, A. Gupta, P. V. Krishna, H. Agarwal, and M. S. Obaidat, "An adaptive learning approach for fault-tolerant routing in Internet of Things," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, 2012: IEEE, pp. 815-819.
[15] P. Rahmani and M. Arefi, "Improvement of energy-efficient resources for cognitive internet of things using learning automata," *Peer-to-Peer Networking and Applications,* vol. 17, no. 1, pp. 297-320, 2024.
[16] T. Anagnostopoulos *et al.*, "Challenges and opportunities of waste management in IoT-enabled smart cities: A survey," *IEEE Transactions on Sustainable Computing,* vol. 2, no. 3, pp. 275-289, 2017.
[17] M. T. A. Bakar and A. A. Jamal, "Determination of suitable resource discovery tool and methodology for high-volume internet of things (IoT)," in *Journal of Physics: Conference Series*, 2021, vol. 1874, no. 1: IOP Publishing, p. 012046.
[18] A. Wheeldon, R. Shafik, T. Rahman, J. Lei, A. Yakovlev, and O.-C. Granmo, "Learning automata based energy-efficient AI hardware design for IoT applications," *Philosophical transactions of the royal society a,* vol. 378, no. 2182, p. 20190593, 2020.
[19] B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, "Trends and strategic researches in internet of everything," *Internet of everything: Algorithms, methodologies, technologies and perspectives,* pp. 1-12, 2018.
[20] L. B. Bhajantri, "Hybrid Centralized Peer to Peer Architecture for Resource Discovery and Secure Communication in Internet of Things," *International Journal on Advanced Science, Engineering & Information Technology,* vol. 13, no. 2, 2023.
[21] H. Zorgati, R. B. Djemaa, and I. Amous, "Efficient IoT resource discovery approach based on P2P networks and Fog Computing," *Internet of Things,* vol. 24, p. 100954, 2023.
[22] H. Mostafaei, A. Montieri, V. Persico, and A. Pescapé, "A sleep scheduling approach based on learning automata for WSN partialcoverage," *Journal of Network and Computer Applications,* vol. 80, pp. 67-78, 2017.
[23] Y. Liu, J. Wang, S. Niu, and H. Song, "Deep learning enabled reliable identity verification and spoofing detection," in *Wireless Algorithms, Systems, and Applications: 15th International Conference, WASA 2020, Qingdao, China, September 13–15, 2020, Proceedings, Part I 15*, 2020: Springer, pp. 333-345.
[24] G. Shinde and H. Olesen, "A survey on service discovery mechanism," in *Intelligent Computing and Information and Communication: Proceedings of 2nd International Conference, ICICC 2017*, 2018: Springer, pp. 227-236.
[25] A. K. Sangaiah, A. Javadpour, F. Ja'fari, H. Zavieh, and S. M. Khaniabadi, "SALA-IoT: Self-reduced internet of things with learning automaton sleep scheduling algorithm," *IEEE Sensors Journal,* vol. 23, no. 18, pp. 20737-20744, 2023.
[26] L. Li, "Study on security architecture in the Internet of Things," in *Proceedings of 2012 international conference on measurement, information and control*, 2012, vol. 1: IEEE, pp. 374-377.
[27] M. Bharti, S. Saxena, and R. Kumar, "Intelligent resource inquisition framework on Internet-of-Things," *Computers & Electrical Engineering,* vol. 58, pp. 265-281, 2017.
[28] J. Mitsugi, Y. Sato, M. Ozawa, and S. Suzuki, "An integrated device and service discovery with UPnP and ONS to facilitate the composition of smart home applications," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014: IEEE, pp. 400-404.
[29] Y. Y. Goland, "Simple service discovery protocol/1.0," *draft-cai-ssdp-v1-03. txt (expired),* 1999.
[30] S. Cirani *et al.*, "A scalable and self-configuring architecture for service discovery in the internet of things," *IEEE internet of things journal,* vol. 1, no. 5, pp. 508-521, 2014.
[31] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "Role of middleware for internet of things: A study," *International Journal of Computer Science and Engineering Survey,* vol. 2, no. 3, pp. 94-105, 2011.
[32] R. Pozza, M. Nati, S. Georgoulas, K. Moessner, and A. Gluhak, "Neighbor discovery for opportunistic networking in internet of things scenarios: A survey," *IEEE access,* vol. 3, pp. 1101-1131, 2015.
[33] N. M. do Nascimento and C. J. P. de Lucena, "Fiot: An agent-based framework for self-adaptive and self-organizing applications based on the internet of things," *Information Sciences,* vol. 378, pp. 161-176, 2017.
[34] F. C. Delicato, P. F. Pires, T. Batista, F. C. Delicato, P. F. Pires, and T. Batista, "The activities of resource discovery and resource estimation," *Resource Management for Internet of Things,* pp. 33-44, 2017.
[35] S. K. Datta and C. Bonnet, "Search engine based resource discovery framework for Internet of Things," in *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, 2015: IEEE, pp. 83-85.
[36] Q. Wei and Z. Jin, "Service discovery for internet of things: a context-awareness perspective," in *Proceedings of the fourth Asia-Pacific symposium on internetware*, 2012, pp. 1-6.
[37] L. Shu, M. Mukherjee, M. Pecht, N. Crespi, and S. N. Han, "Challenges and research issues of data management in IoT for large-scale petrochemical plants," *IEEE Systems Journal,* vol. 12, no. 3, pp. 2509-2523, 2017.
[38] P. T. Nguyen and A. Aggarwal, "Enhanced DNS-based service discovery in an internet of things (IoT) environment," ed: Google Patents, 2018.
[39] M. Achir, A. Abdelli, L. Mokdad, and J. Benothman, "Service discovery and selection in IoT: A survey and a taxonomy," *Journal of Network and Computer Applications,* vol. 200, p. 103331, 2022.
[40] X. Liu and Y. Deng, "A new QoS-aware service discovery technique in the Internet of Things using whale optimization and genetic algorithms," *Journal of Engineering and Applied Science,* vol. 71, no. 1, p. 4, 2024.
[41] Y. Hassanzadeh-Nazarabadi, S. Taheri Boshrooyeh, S. Otoum, S. Ucar, and O. Ozkasap, *DHT-based Communications Survey: Architectures and Use Cases*. 2021.
[42] L. Kabre and T. Tiendrebeogo, "Comparative Study of can, Pastry, Kademlia and Chord DHTS," *International Journal of Peer to Peer Networks,* vol. 12, pp. 1-22, 08/31 2021, doi: 10.5121/ijp2p.2021.12301.