Ingénierie des Systèmes d'Information
Vol. 30, No. 1, January, 2025, pp. 111-135
Journal homepage: http://iieta.org/journals/isi

A Comparative Analysis of Service Composition Approaches for the Internet of Things

Farida Retima[1*], Saber Benharzallah[2]

[1] Department of Computer Science, El-Oued University, Biskra 07000, Algeria
[2] Department of Computer Science, Batna 2 University, Batna 05000, Algeria

Corresponding Author Email: retima-farida@univ-eloued.dz

https://doi.org/10.18280/isi.300110

Received: 31 July 2024
Revised: 7 November 2024
Accepted: 28 November 2024
Available online: 25 January 2025

Keywords:
Internet of Things (IoT), service composition, approach, IoT service, QoS (quality of service)

ABSTRACT

In the past ten years, the Internet of Things (IoT) attracted many researchers, by combining diverse and distributed objects to display information about the physical world. The service composition process provides an interaction between the user needs and the smart objects of the IoT environment. Therefore, it is regarded as an essential module. According to previous research findings, different approaches assisted the service composition for IoT. However, most previous research reviewed different service composition approaches in IoT environments, depending on an insufficient number of criteria. Additionally, there is no complete and exhaustive review of this field. Therefore, our contribution to this paper is to comprehensively analyze IoT's popular service composition techniques, considering all possible criteria that could influence this process. Also, we describe the different service composition techniques in seven (07) main categories: agent-based, heuristic-based, QoS-based, probabilistic-based, social network-based, Petri net-based, and recommendation-based. Additionally, we discuss the benefits and drawbacks of the important technique in each category. Thereby, this paper aids researchers in selecting the optimal technique based on the particular requirements and constraints of the application domain. It also clarifies future directions and challenges related to service composition in the IoT that need to be addressed in this field. Four objectives are included in this paper: 1) firstly, it defines a set of specific criteria: privacy, security, scalability, service selection, heterogeneity, performance, adaptability, composition mechanism, energy consumption, QoS estimation, anomaly detection, optimization, interoperability, trust management, monitoring, service representation, and implementation tools; 2) secondly, it presents various challenges and possible solutions to these issues for service composition in IoT systems; 3) thirdly, it makes use of the previous issues to compare the well-known current approaches; 4) fourthly, it helps researchers to identify the most important techniques for IoT service composition and the applicability of each one of them. Consequently, this paper aids researchers in developing more efficient service composition methods for future research.

1. INTRODUCTION

Nowadays, the Internet of Things (IoT) is a new paradigm that enlarges our daily lives through a wireless network of uniquely identifiable objects [1]. The number of physical objects connected to the Internet is soaring rapidly to embody the ideas of the IoT. Cisco's evaluation shows that 50 billion devices will be connected to the Internet by 2020 to achieve an intense IoT environment [2]. IoT applications are opted to enable connected objects to create robust and valuable services. IoT services are primarily differentiated from traditional services by their direct connection to the physical world [3]. By merging some resources in many IoT applications, the demand of users can be satisfied with a process called service composition [2]. This latter comprises combining the functionality of different services into a single process to fit complex demands that cannot be satisfied by a single service. Nevertheless, it is a challenging task because of the heterogeneity and mobility of objects in an IoT environment. Accordingly, different approaches are suggested by researchers to address the service composition in the IoT where each of them focuses on various aspects of the IoT.

In this paper, we conducted a comparison analysis of the popular approaches using different criteria. These issues (described in section 3) directly influence the service composition in IoT. However, few works have reviewed service composition approaches in IoT environments.

Aoudia et al. [4] compared 30 service composition approaches using insufficient criteria regarding the various issues faced in IoT. This survey article did not address important criteria, including scalability, heterogeneity, energy consumption, QoS parameters, and interoperability.

Asghari et al. [5] further reviewed some IoT service composition approaches without comparing them. It covered a discussion of the primary issues and offered open perspectives and future research directions.

Kashyap et al. [6] provided an overview of the approaches currently used to develop an efficient IoT service composition.

The authors adopted various techniques to provide an optimized solution to the service composition (SC) problem in IoT environments.

Rabah et al. [7] further addressed the composition mechanism of IoT services, concentrating only on QoS-aware approaches and compared some proposed methods in the related literature.

On the other hand, the research paper by Safaei et al. [8] compared different enterprise service composition techniques in the IoT field to uncover the power of each one and address challenges in real-world applications. This comparison was addressed on three criteria (the problem, the QoS parameter, and the model).

Han et al. [9] clarified the practicability of the future full-IP IoT with real-time web protocols to clearly explain the service composition issue for IP intelligent objects. They also discussed the research challenges and included a literature review. In other words, they provided an overview of service composition models using some composition needs.

Razian et al. [10] reviewed previous research findings in service composition in various environments, such as cloud computing, mobile computing, the Internet of Things (IoT), and Fog (Edge) computing, which are based only on QoS uncertainty.

Barakat et al. [11] evaluated existing IoT service composition mechanisms to identify their scalability requirements within the IoT environment.

Hamzei and Navimipour [2] further conducted a comparative study of some service composition methods in the IoT. They distinguished these selected approaches into four types: framework, service-oriented architecture (SOA), heuristic, and model-based. They displayed the similarities and changes in the current service composition in the IoT according to a few parameters.

Most of the previous surveys compared different service composition approaches in IoT according to an insufficient number of parameters or untargeted criteria. However, in our current work, we evaluate twenty-eight (28) popular service composition approaches in the IoT using all possible criteria. Thus, this paper provides an extensive survey of comparative studies for service composition approaches based on different issues: privacy, security, scalability, service selection, heterogeneity, performance, adaptability, composition mechanism, energy consumption, QoS estimation, anomaly detection, optimization, interoperability, trust management, monitoring, service representation, and implementation tools. These criteria directly influence the service composition in IoT. Thereby, this paper aids researchers in selecting the optimal technique and developing more efficient service composition methods in future research. It also clarifies future directions and challenges related to service composition in the IoT that need to be addressed in this field.

The following sections of this paper are organized as follows: Section 2 demonstrates the background for service composition. Section 3 describes all possible technical and non-technical issues that need to be studied for service composition in IoT. Section 4 presents different approaches proposed for service composition in IoT. In contrast, Section 5 compares these approaches according to our defined criteria with their sub-criteria, illustrated in following tables. In Section 6, we present a detailed discussion of our comparison study. In Section 7, we discuss open research challenges for future works. Finally, Section 8 concludes this paper.

2. BACKGROUND

The Internet of Things (IoT) is a new paradigm that makes connectivity between physical and virtual things using advanced distributed services [12]. Currently, the IoT environment increases rapidly and appropriately for user interactions, which can compose heterogeneous service technologies. Each atomic service has an associated IoT device that performs the service's functionality. A single IoT service cannot fulfill users' needs. Therefore, composite services are necessary to satisfy complex requirements from multiple application domains, and intelligent object services can be aggregated through the service composition process [5]. It can be efficiently used to develop innovative applications. An efficient service composition process makes supporting applications in a dynamic network environment possible. This process is the key to supporting the development of Internet of Things applications [9]. Service composition becomes more challenging in IoT environments due to their high scalability, complexity, heterogeneity, and dynamicity [13].

The need for practical service composition algorithms that can handle the increasing number of equivalent services in these large-scale service environments is one of many problems. Another difficult challenge is finding the optimal combination of services to satisfy QoS constraints, known as QoS-aware service composition. In this case, quality of service (QoS) is the primary factor determining whether the composed application succeeds or fails [2]. Current service composition techniques aim to identify a QoS-optimal service composition solution [2]. The QoS of the user requests should be evaluated by gathering the QoS criteria for each service. The QoS of composite services is determined by the composition patterns. This latter could be sequential, parallel, loop, and branch models, which provide distinct QoS aggregation functions [5].

The IoT architecture's service composition is briefly described in Figure 1. The five layers of this architecture are represented as follows:

(1) Sensors and IoT devices are part of the data collection layer, which gathers data from the Internet of Things environment.

(2) The connection layer connects smart objects to other servers. This layer allows data acquired from the collection layer to be processed and transmitted.

(3) Multiple private or public clouds provide different services through the cloud layer.

(4) Depending on the user's functional or non-functional needs, the service composition layer combines several sub-services.

(5) The application layer gives specific composited services to end users based on their needs [5].

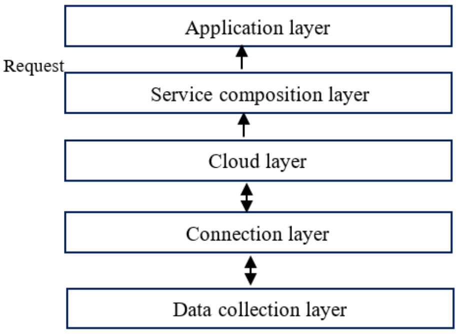

Figure 1. Service composition architecture in IoT

3. FUNDAMENTALS CHALLENGES OF SERVICE COMPOSITION IN IOT

When discussing service composition in the IoT, we currently encounter numerous challenges in the literature. Among these issues, we are particularly interested in privacy, security, scalability, service selection, heterogeneity, performance, adaptability, composition mechanism, energy consumption, QoS estimation, anomaly detection, optimization, interoperability, trust management, monitoring, service representation, and implementation tools. These challenges directly influence the service composition in IoT.

3.1 Privacy

With the development of the Internet of Things (IoT), the increasing quantity of interactions between connected objects forces a reconsideration of the security of personal data. Service composition in IoT has become more complex due to the creation of numerous services and complex client demands. A significant consideration when composing different IoT services is protecting privacy when exchanging data [14-44]. Most research has concentrated on adding a dynamic security check between plan generation and execution to protect user data. As a result, we can ensure that implementing this plan will not provide user data to an unauthorized IoT service [44].

Three challenges are available for this criterion:

(1) Hardware layer: should protect privacy when data is temporarily gathered and stored on the device.

(2) At the protocol layer, ensure adequate security for communication between IoT services.

(3) The application layer: monitors who can access the context.

3.2 Security

For various reasons, the IoT environment is highly susceptible to attacks. Although service composition is essential in IoT, security issues have yet to be fully explored. Additionally, many characteristics that make the composition of IoT services interesting, such as increased data accessibility, dynamic application-to-application connections, and the absence of human intervention, are incompatible with conventional security models.

Two perspectives are available for this criterion.

(1) Service-level security involves basic security features. Web services (WS) can generally embed security headers to protect SOAP messages transmitted through unsecured channels. The WS-Security standard specifies how security tokens, cipher texts, and signatures may be included in such headers.

(2) Composition-level security concentrates on security problems faced during web service composition [45].

3.3 Scalability

As the number of connected devices and services rapidly increases, IoT systems must interact with billions of services. Therefore, scalability becomes crucial. Scalability poses the problem of dynamically composing services based on different IoT resources [22]. The scalability factor indicates the capacity to deal with increasing workloads by integrating new operations and devices as service nodes for user functions while maintaining the quality of the existing services [5].

For this criterion, there are three possible perspectives:

(1) Horizontal scalability: includes adding or removing entities from an IoT system. It aims to divide the workload among several things to reduce overall workloads and response time.

(2) Vertical scalability: indicates the insertion or deletion of computational resources within a single entity, such as increasing processor capacity to accelerate processing or increasing memory capacity to expand the buffer size.

(3) Functional scalability: supporting many IoT services [12].

3.4 Service selection

The capacity to compose services is used to identify the best services to satisfy the composition needs. The selection algorithm results are a list of services that satisfy the user's functional and quality needs [46].

3.5 Heterogeneity

The service composition approach fulfills user requirements based on heterogeneous services from different providers. Hardware and software resources employ Heterogeneous technologies and languages [47].

For this criterion, there are two possible sub-criteria:

(1) Semantic heterogeneity of objects and IoT system services.

(2) Technological heterogeneity.

3.6 Performance

The service composition process requires calculations and resources to create a service that satisfies the user's preferences. This process consumes a lot of time, resources, and processing. This challenge is due to the increasing number of services available in the IoT.

3.7 Adaptability

IoT environments are very dynamic; new services could appear anytime, current services could be discontinued or unavailable for a period, and service features (QoS attributes) could change. The service composition should enable dynamic service binding or automatically substitute certain component services with new ones that provide a higher quality of service at runtime.

Concerning this criterion, the following points can be studied:

(1) Type: Adaptations can take two forms: proactive adaptation manifests before a specific action, and reactive adaptation occurs after it.

(2) Mechanism: indicates the technique used to permit the adaptation. This technique is based on specific methods.

(3) Time: demonstrates the time appropriate to adapt (development time, compilation time, deployment time, and runtime).

(4) Implementation of adaptive code: This code may be installed in the operating system, the executing engine, or the composition process's source code [48].

3.8 Composition mechanism

Combining services is done through a composition process.

As a result, an IoT system needs a things infrastructure, a definition of a service, and the choice of a composition mechanism. A service composition mechanism defines a significant interaction between services. The definition of the service composition mechanisms involves two workflows: orchestration (which can be distributed or centralized) and choreography.

(1) Centralized orchestration: A single coordinator node executes a composite service specification.

(2) Decentralized orchestration: Multiple coordinators collaboratively define workflow control.

(3) Choreography applies a decentralized method to compose services. It is an overall description of the existing services, determined by exchanging messages between them [12].

3.9 Energy consumption

The rapid development of service composition technology in IoT has significant consequences, including increased service energy consumption. Effectively reducing service energy consumption is a challenge.

We can make the following two specifications of the energy consumption:

(1) Environmental energy consumption

(2) Energy consumption of network communication [49].

3.10 QoS estimation

Finding an appropriate service to build a composite service based on non-functional parameters like Quality of Service (QoS) has become a significant concern due to the vast number of web services with similar functionality [50]. The objective of QoS-aware service composition is to identify the best candidate services to satisfy users' QoS needs and optimize the overall quality of the composite service [51]. When estimating QoS, we should consider factors like availability, cost, response time, and more.

3.11 Anomaly detection

Various factors can cause anomalies in the quality of internet services, including network failure, heavy system workload, and temporary machine failures. Anomalies are abnormal data points. Anomaly detection is an extensively studied aspect of the services composition process. Consequently, examining the historical QoS records and eliminating anomalies is crucial to building an accurate QoS model [52].

3.12 Optimization

Optimization becomes a crucial concern with the increasing complexity of computing tasks and the variety of services in the IoT environment. In this context, it is necessary to optimize the service composition reasonably to provide users with the most appropriate services. The optimization problem of IoT services composition needs an optimization method that depends on a constrained multi-objective model based on target criteria [53].

3.13 Interoperability

Several service models are usually involved in the service composition process. These models may represent the composite service from different perspectives. Maintaining interoperability between these models is necessary to ensure the composition process's accuracy. The objective of interoperability is the capacity of multiple systems to reciprocate and benefit information [54]. Services can communicate and exchange data. Also, various middleware can interact and utilize shared data.

3.14 Trust management

Services interact in dynamic environments, including the Internet of Things (IoT), in an undefined way, leading to a significant aspect in trust management. In addition, the prevalent usage of services and composite services in distributed environments discloses the increasing necessity of trust management regarding the service composition process. Therefore, users in these environments need more confidence to identify trustworthy service providers. As a result, trust can be regarded as one of the most essential elements for choosing the relevant providers and improving the quality-of-service composition process [55].

3.15 Monitoring

One major issue recognized is the runtime monitoring of the service composition process. To obtain extensive knowledge about the composite service and its component services work within the execution environment, it is necessary to oversee and monitor its progress after its execution. In addition, requirements and features validated during design time may be invalidated during runtime [56].

3.16 Service representation

Define the architecture type to represent service in the IoT system.

3.17 Implementation tools

Represent different tools for implementing service composition in IoT systems (dataset, programming language, platform).

For a detailed study of these various issues, Table 1 presents the most important challenges and proposed solutions in the literature for each criterion.

Table 1. Challenges and proposed solutions for different technical issues

Technical Issues	Challenges	Solutions
Privacy	➢ Data security in transmission and storage ➢ User authentication and access control ➢ Data minimization and purpose limitations (IoT devices often collect excessive data	➢ Encryption ➢ Edge computing (by processing data closer to where it's generated) ➢ Multi-Factor Authentication 2(MFA) ➢ Role-Based Access Control (RBAC)

	beyond what’s necessary for their functionality) ➢ End-to-End privacy policies ➢ standardization and interoperability	➢ Device Identity Management ➢ Data Minimization Techniques (Configure IoT devices to collect only essential data, applying anonymization) ➢ Local data processing (process data locally to retain only relevant information) ➢ Integrate privacy concerns into the design of the Internet of Things system from the beginning. ➢ Blockchain (use blockchain to create an immutable record of transactions) ➢ Industry standards: adopt frameworks like ISO/IEC 30141 for IoT security architecture or NIST guidelines for IoT device cybersecurity ➢ Interoperability frameworks (such as oneM2M for machine-to-machine interactions)
Security	➢ Device resource constraints ➢ Network security and communication risks ➢ Physical security of devices ➢ Firmware and software vulnerabilities	➢ Lightweight cryptography ➢ Efficient authentication: DTLS (Datagram Transport Layer Security) or CoAP (Constrained Application Protocol). ➢ Encrypted communication protocols like TLS/DTLS ➢ Intrusion Detection and Prevention Systems (IDPS) ➢ Tamper-resistant hardware ➢ Regular physical audits ➢ Regular Firmware Updates ➢ Secure boot and firmware integrity checks
Scalability	➢ Network bandwidth and latency ➢ Processing and Analytics ➢ Data storage and management ➢ Energy management and battery life	➢ Edge Computing ➢ Bandwidth-efficient protocols (MQTT and CoAP protocols) ➢ Edge analytics ➢ Distributed databases (use distributed databases like Apache Cassandra, and MongoDB) ➢ Data compression and aggregation ➢ Power-Efficient Protocols ➢ Adaptive power modes
Service selection	➢ Heterogeneity of IoT devices and services ➢ Quality of Service (QoS) requirements ➢ Dynamicity ➢ Cost efficiency and resource optimization ➢ Data integrity and reliability	➢ Standardized communication protocols (like MQTT, CoAP, and RESTful APIs) ➢ Interoperability frameworks (Implement IoT platforms like oneM2M, OCF) ➢ QoS-Based selection algorithms ➢ Multi-Criteria Decision Making (MCDM) ➢ Service Level Agreements (SLAs) ➢ Context-aware middleware (that gathers and processes contextual information) ➢ Edge Computing for real-time adaptation ➢ Cost-aware selection algorithms ➢ Reputation and trust-based selection
Heterogeneity	➢ Device and platform diversity ➢ Inconsistent data formats and structures ➢ Software and firmware compatibility ➢ Data processing and aggregation	➢ Interoperability standards (open IoT standards, such as oneM2M, Open Connectivity Foundation (OCF)) ➢ Use common APIs or software development kits (SDKs) ➢ Data normalization and transformation ➢ Ontology-based data models (Semantic Sensor Network ontology) ➢ Standardized firmware platforms (such as Zephyr RTOS or RIOT OS) ➢ Over-the-air (OTA) updates ➢ Unified data models ➢ Edge Analytics
Adaptability	➢ Dynamic network topology ➢ User preference and behavior adaptation ➢ Handling device resource constraints ➢ Real-Time decision-making ➢ Scalability of adaptive algorithms	➢ Implement self-organizing network protocols ➢ Use auto-discovery protocols (like mDNS, UPnP) ➢ Implement user profiling to build personalized service profiles ➢ Collaborative filtering and recommendation engines ➢ Implement energy-efficient protocols (e.g., MQTT-SN, CoAP) ➢ Edge data processing ➢ Edge and Fog Computing ➢ Implement low-latency communication protocols (e.g., 5G, TSN) ➢ Implement hierarchical and distributed adaptive algorithms. ➢ Scalable middleware solutions (Apache Kafka, RabbitMQ)

Energy consumption	➢ Limited battery life ➢ Frequent data transmission ➢ Processing and computation overhead ➢ Energy constraints in high-density IoT deployments	➢ Incorporate energy-harvesting techniques (like: solar, kinetic, and thermal) ➢ Low-power hardware components ➢ Data compression and aggregation ➢ Batching and scheduling transmissions ➢ Edge and Cloud Offloading ➢ Energy-Efficient algorithms ➢ Optimize network traffic patterns ➢ Implement energy-aware routing protocols
Anomaly detection	➢ High volume and velocity of data ➢ Data heterogeneity ➢ Dynamic and evolving patterns ➢ Scalability in large-scale IoT networks	➢ Perform preliminary anomaly detection on edge devices. ➢ Data normalization and standardization ➢ Feature extraction and transformation ➢ Context-aware detection ➢ Distributed anomaly detection ➢ Hierarchical detection frameworks
Optimization	➢ Energy consumption optimization ➢ Network bandwidth optimization ➢ Optimization for adaptability in dynamic environments	➢ Duty Cycling and low-power modes ➢ Energy-efficient communication Protocols (like LoRaWAN, BLE, and Zigbee) ➢ Bandwidth-efficient protocols (like MQTT and CoAP) ➢ Edge data filtering (Filter data at the edge) ➢ Self-learning algorithms (reinforcement learning) ➢ Context-aware optimization
Interoperability	➢ Diverse communication protocols ➢ Heterogeneous data formats and structures ➢ Security and privacy interoperability ➢ Inconsistent device discovery mechanisms	➢ Protocol gateways ➢ Adoption of universal protocols (MQTT or CoAP) ➢ Common data models (such as Sensor Markup Language) ➢ Middleware for data translation ➢ Standardized security protocols (like TLS, DTLS, and OAuth 2.0 for encryption) ➢ Unified privacy frameworks ➢ Universal discovery protocols ➢ Middleware for device discovery
Trust management	➢ Trust establishment between heterogeneous devices ➢ Scalability of trust management solutions ➢ Trust management in decentralized IoT networks ➢ Privacy protection for user trust	➢ Trust evaluation models ➢ Edge computing for trust processing ➢ Reputation-based trust models ➢ Blockchain-based trust ➢ Data anonymization ➢ Decentralized data storage
Monitoring	➢ Scalability of monitoring solutions ➢ Resource constraints of IoT devices ➢ Real-time data collection and analysis ➢ Maintaining data privacy during monitoring	➢ Hierarchical and Edge-based monitoring ➢ Cloud-based monitoring platforms ➢ Edge Computing for intensive processing ➢ Edge and Fog Computing ➢ Data encryption ➢ Access control and authentication

4. SERVICE COMPOSITION APPROACHES IN IOT

The related review shows many approaches that allow service composition in the Internet of Things. In this section, we present the most representative and well-known approaches. In addition, to highlight the most important differences between these approaches, Table 2 shows the most important of them.

Asghari et al. [14] proposed a privacy-aware cloud service composition approach to optimize QoS in the IoT environment. It presented an IoT-based cloud service composition framework, incorporating the privacy level computing model and a composite hybrid evolutionary algorithm (SFLA-GA). This latter used the Shufed Frog Leaping Algorithm (SFLA) and the Genetic Algorithm (GA). This algorithm optimizes the proposed service composition by aggregating the different QoS parameters used as fitness values. The simulation results indicated that the suggested approach ameliorates the fitness compared to other recent algorithms.

Razian et al. [15] focused on the issue of QoS value uncertainty in the dynamic environments of Cloud and IoT by proposing a new anomaly-aware robust service composition (ARC). The suggested approach aims to compose services using Bertsimas and Sim's mathematical robust optimization method. The results show that the proposed approach finds optimal solutions with an average improvement of 14.55% compared to information theory-based and clustering-based methods.

Pavan Kumar et al. [16] suggested an approach for QoS-aware IoT optimal service composition. It uses a decision tree and genetic algorithm (GA). The operation uses two levels of QoS parameters to arrive at the optimal service composition. This system's scalability is more reliable and accurate than the regular optimal service composition.

Baker et al. [17] proposed a new multi-cloud IoT service composition algorithm called (E2C2). This algorithm pursues selecting the least possible number of IoT services to create an energy-aware composition plan to satisfy user needs. The suggested algorithm was evaluated considering four well-known service composition algorithms in several cloud environments. Considering the results of this evaluation, the suggested approach has a better performance.

In the research by Zhang et al. [13], a relation between services has been suggested by applying a model for service dependency and minimizing the traversal space using efficient filtering techniques. The generated service composition can be directly displayed to avoid a backtracking search based on a composition path traversal sequence. Experiments of this paper have indicated that this approach can reach the top-k QoS-optimal service composition and improve performance time.

Badidi et al. [18] have proposed an integrated framework for improving personalized mobile cloud services. The proposed framework used a standard service metadata specification model to adapt context information. To demonstrate the suggested algorithm's effectiveness, this paper's authors evaluated their framework using an application scenario.

Bao and Chen [19] and Chen et al. [20] have suggested adaptive and scalable trust management to enable service composition applications in SOA-based Internet of Things systems. The researchers promoted a technique based on distributed collaborative filtering to sort out feedback using similarity ratings of friendship, social contact, and community of interest relationships. The efficacy of the suggested trust management is demonstrated through service composition use cases and a comparative performance evaluation against EigenTrust and PeerTrust.

An enhanced eagle strategy algorithm has been presented by Rajendran et al. [21] to improve the computation time in large-scale cloud-based Dynamic Web Service Composition (DWSC) and on both functional and non-functional service attributes. Proposing the improved bio-inspired method can ameliorate computation time, particularly in a large-scale repository of IoT.

In the study by Dar et al. [22], service orchestration and choreography ideas were used to create a flexible and adaptive service composition for Large-scale (VLS) IoT systems. To evaluate the composition process for this vast number of devices, the authors intend to experiment with real large-scale sensor network applications and deploy a very large-scale open wireless sensor network platform via the SensLab platform.

Two popular meta-heuristic algorithms, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), were suggested by Kashyap et al. [23] to address QoS for service composition issues. This paper focuses on selecting a suitable SC (Service Composition) for Internet of Things applications, aiming to optimize the QoS parameters to satisfy the user's needs. Experimental findings show that GA and PSO can improve the effectiveness of solutions for service composition problems in the Internet of Things.

White et al. [24] applied goal-oriented service composition and collaborative filtering to propose an approach to self-managing dependable systems. QoS prediction permits the goal-driven model to activate adaptation decisions by endorsing execution paths to adapt dynamically. The experiment results show that the prediction composition has primarily improved compared to the without prediction.

Razian et al. [25] have proposed a scalable anomaly-aware approach (SAIoT). It comprises two main components: the first uses machine learning anomaly detection to model QoS values, and the second employs an efficient meta-heuristic algorithm to arrive at the ideal composition. Comparing the proposed approach to previous research, like information theory-based and advertised QoS-based methods, the experimental results on real-world datasets demonstrate a 30.64% average improvement in the QoS value of a composite plan, achieved with equal or lower cost.

The challenge of achieving the best possible balance between the QoS level and the consumed energy of the IoT service composition has been formulated by Alsaryrah et al. [26] as a bi-objective shortest Path Optimization (BSPO) problem. For this purpose, it employed Pulse's algorithm to solve this issue. The suggested strategy, known as BSPO, aims to reduce two elements: the energy consumption of the composite service and the quality of service (QoS), which includes execution time, network latency, and service cost. The research evaluations proved that the suggested approach was executed rapidly in various complex service profiles. In the interim, it can gain good performance in terms of energy consumption and, consequently, network lifetime by keeping an acceptable QoS level.

Yang et al. [27] suggested a Petri net-based model for IoT service composition. It evaluates cost-effectiveness using a comprehensive performance function (RTC) considering cost, response time, and reliability. Moreover, the FBasedMonitor algorithm does well in solving composition in dynamically changing environments. The research proves the soundness and correctness of the proposed model and algorithms.

Berrani et al. [28, 29] have suggested a multi-agent system-based approach for IoT service composition where multiple agents are employed to meet user demands. The use-case scenarios and comprehensive testing clarify the multi-agent system's interest and suitability for service composition.

From the sensory data aggregation and selection perspective, Yang and Li have presented an efficient method [30]. In other words, IoT quality of service (QoS) is addressed when selecting candidate services. A method for global optimization called upgraded binary coded genetic algorithm (GA) is applied. The service composition scheme with the highest overall function value is considered the optimal solution. The research results detect the leverage of the proposed method in IoT.

Kashyap and Kumari [31] have proposed a hyper-heuristic approach to solving the service composition problem in the Internet of Things. Twenty-five test data instances are used to measure the effectiveness of the hyper-heuristic approach, and the results are compared to the genetic algorithm.

A cloud-based IoT platform medical monitoring program is found by Asghari et al. [32]. The patient's medical conditions are determined using his physiological data from the Internet of Things devices to predict diseases. Considering the patient's non-functional preferences, meta-heuristic approaches can be used to find suitable composite services. Results show that the suggested approach successfully provides composite health prescriptions by achieving accurate disease diagnosis.

Li et al. [33] have formally described and analyzed reliability and cost-related features of service composition in the Internet of Things using a probabilistic approach. The authors examine and verify the properties of the suggested service composition models using a probabilistic model checker called PRISM.

Urbieta et al. [34] have offered an adaptive service composition framework. This latter is based on the abstract service model WEASEL, which describes services and user tasks in terms of their signature, specification, and conversation. According to the evaluation, the wEASEL-based system outperforms other methods regarding accuracy and provides end-users with more composition opportunities.

Kashyap et al. [35] have proposed an evolutionary multi-objective optimization algorithm to solve service composition issues for IoT-based applications. The authors of this work demonstrated how to apply the Non-dominated Sorting

Genetic Algorithm (NSGAII), one of the widely used algorithms. In other words, the evaluation showed that NSGAII could be initiated to solve the service composition problem efficiently.

Chen et al. [36] have presented a distributed method for IoT device management and service composition. IoT services are modeled on three dimensions using social network theory: location, type, and correlation based on the relationships between IoT devices. Simulation experiments using real-world datasets are carried out to verify the efficacy of the suggested method.

A hybrid method has been suggested by Bouazza et al. [37] to provide users with personalized IoT service recommendations by combining ontology and implicit collaborative filtering. The proposed approach involves extending the social relationships between users and their objects to satisfy the service requirements by finding alternate sources of the missing information. From the results, the suggested hybrid algorithm outperforms the current algorithms considering accuracy and performance.

Wanigasekara et al. [38] have presented a novel method that uses contextual bandit algorithms to sense information from the community of mobile devices interacting with the environment and use it to learn and suggest contextually relevant IoT resources. The researchers described the general methodology, a functional prototype system, and a validation of the system obtained from two experimental scenarios with encouraging preliminary results.

Faieq et al. [39] adopted a context-aware system based on recommendations for service composition. A collaborative filtering framework based on auto-encoders to predict the QoS values has been used to select the best services to fulfill each task in the service model. The service with the best-predicted value is then selected. The experiment results demonstrate the efficacy and efficiency of the suggested recommendation policies.

Meissa et al. [40] presented a new recommendation approach to finding and recommending personalized and reliable web services appropriate for composition. Seven different variant algorithms (item-based, user-based, and popularity-based) are compared using the MovieLens 20M dataset to verify the effectiveness of the suggested approach. The results of the experiments demonstrate that the proposed model increases recommendation accuracy by 12% compared to other methods.

Cao et al. [41] have suggested a QoS-driven service recommendation framework leveraging RTM (Relational Topic Model) and FMs (Factorization Machines) to develop value-added composite web services for IoT Mashup applications. According to experimental results, the proposed approach significantly improves precision, recall, and F-measure parameters compared to other recommendation systems.

Chen et al. [42] have proposed an access service recommendation strategy in Social IoT (SIoT) environments to improve service composition efficiency and protect against malicious attacks. Furthermore, an energy-aware mechanism was considered for workload balancing and network stability. Experiments show that the suggested scheme is beneficial and effective in rating accuracy, dynamic behavior, network stability, and quality of service composition.

Table 2. Differences between service composition approaches in the IoT environment

Ref	Strengths	Weaknesses
[14]	Unique focus on privacy in QoS optimization Combing QoS and privacy criteria into the service composition framework A quantitative assessment is conducted to validate the potential benefits of this approach	Based on simulated evaluations rather than real-world IoT deployments Scalability for large IoT networks is not considered. Privacy-aware QoS optimization can be computationally intensive, which might challenge IoT devices with limited resources Limited comparison with existing models
[15]	Focus on anomaly detection and robustness Consideration of QoS uncertainty Enhanced reliability for IoT applications Quantitative validation through simulations	It has not been evaluated in practical IoT environments The computational requirements are not considered The ARC model’s performance in large-scale IoT environments is not discussed The model considers only uncertainties in QoS analysis
[16]	Effective use of genetic algorithms for QoS optimization Heuristic structure to enhance the search efficiency Focus on multiple QoS Parameters Flexibility for diverse IoT applications	The results are based on simulations rather than actual IoT environments Computational complexity of genetic algorithms especially in large IoT networks Scalability for large-scale IoT systems is not explored Lack of a detailed exploration of heuristic parameters and their impact on performance
[17]	Focus on energy efficiency Applicability to multiple IoT applications The algorithm is more practical for real-world IoT devices that operate on limited power	Limited focus on other QoS metrics Scalability concerns are not addressed The study does not consider context-specific factors that can impact energy usage A detailed analysis of the computational costs is not discussed
[13]	Innovative use of service dependency graphs Top-K service compositions based on QoS Adaptability to dynamic environments QoS-optimized compositions	Lack of real-world implementation Scalability for large-scale service graphs is not discussed Computational costs of generating a service dependency graph are not explored Dependency graph complexity The trade-offs between different QoS metrics are not explored
[18]	Focus on personalization which can improve user satisfaction Integrating context-aware service composition Adaptive service provisioning for user context in real-time	Limited focus on privacy Computational overhead is not considered Scalability challenges are not discussed

	The focus on mobile users makes the research applicable to the real world	Data inaccuracy or delayed data retrieval could affect service quality, but these challenges are not fully addressed
[19, 20]	Applying trust management to IoT service composition Providing a practical use case, showing how trust mechanisms can improve IoT service reliability and user satisfaction Enhancing overall IoT system security and stability. Decentralized trust model Combination of direct and indirect trust Framework adaptable to different types of IoT devices and services	Lack of quantitative evaluation of the proposed trust model Scalability issue in large networks is not addressed Limited discussion on trust dynamics for IoT environments The mechanism to filter unreliable recommendations (indirect trust) is not addressed A discussion on adapting the model for low-resource devices is not mentioned
[21]	Novel use of nature-inspired algorithm in the context of IoT Adaptation to dynamic environments The conceptual approach handles large-scale IoT networks effectively Optimization for QoS parameters	Complexity of nature-inspired algorithms Lack of empirical results or real-world validation Limited focus on privacy and security concerns Absence of comparative analysis
[22]	Focus on scalability and dynamic Adaptation The approach applies to a wide range of IoT applications. Efficient resource utilization	Limited real-world validation The approach focuses on scalability and adaptability but does not address other important factors Potential complexity in implementation
[23]	Innovative approach with hybrid optimization (combining genetic algorithm (GA) and particle swarm optimization (PSO)) Focus on QoS optimization Potential for scalability A quantitative assessment of the GA-PSO model's performance	The computational complexity of the genetic algorithm and particle swarm optimization are not addressed Sensitivity to parameter tuning (population size, mutation rate...) Lacks a comprehensive comparison with other service composition methods, such as heuristic-based
[24]	Focus on reliability in IoT service composition Application of machine learning for QoS prediction Support the scalability of IoT systems Comprehensive evaluation to prove the validity of the approach in different IoT scenarios	Dependence on historical data quality is not addressed Limited exploration of multi-factor influences on QoS (network conditions, device mobility) Machine learning-based QoS prediction needs a high computational cost
[25]	Focus on scalability in CloudIoT. Anomaly-aware composition to improve the reliability of service composition Dynamic adaptation Emphasize modularity, making it adaptable for diverse IoT use cases	Lack of real-world testing The anomaly detection and scaling mechanisms introduce significant computational costs The impact of detected anomalies on QoS has not been explored
[26]	Addressing energy efficiency in IoT Focusing on both energy consumption and QoS Quantitative analysis and simulations Practicality for IoT applications	Limited real-world validation Scalability concern is not considered The trade-offs between energy consumption and QoS are not explored
[27]	The model facilitates real-time monitoring. Petri nets are well-suited to distributed systems (distributed nature of IoT environments) Flexibility and scalability using Petri nets	Complexity of Petri Net models High computational overhead Limited focus on QoS optimization and privacy concerns
[28, 29]	Innovative use of multi-agent systems Enhanced flexibility and scalability Real-time adaptability Focus on decentralization Potential for autonomous decision-making	The complexity of coordination and communication in multi-agent systems Addressing agent reliability and fault tolerance has not been fully explored Ensuring consistency and handling concurrency across agents are not addressed
[30]	The user-centric approach enhances the applicability of the proposed framework Flexibility and adaptability	Limited real-world implementation Limited practical validation Scalability concern is not addressed Lack of consideration for QoS and security
[31]	Innovative hyper-heuristic approach The proposed hyper-heuristic framework is designed to handle the scalability Dynamic adaptability	Limited real-world application Complexity of implementation Hyper-heuristic approaches often depend on the tuning of various parameters While heuristics can improve efficiency, they may not always guarantee optimal solutions
[32]	The study focuses on medical monitoring Cloud-based IoT Integration which provides scalability, flexibility, and enhanced data management capabilities Focus on data security and privacy Potential for real-world application	High scalability concern is not considered. The financial implications of adopting cloud-based IoT solutions for healthcare monitoring are not discussed Complexity of Cloud-based IoT integration
[33]	Probabilistic modeling approach Focus on reliability and cost The cost-oriented aspect of the model makes it practical for resource-constrained IoT environments	Limited applicability to dynamic IoT Environments Lack of privacy considerations Scalability for large-scale IoT Systems is not considered

	Quantitative assessments demonstrate the model's ability to optimize reliability and cost	
[34]	Relevance to smart city applications Emphasis on adaptivity and context-awareness Innovative use of contextual data	Incomplete, or inaccurate context data is not addressed. Context-aware adaptivity can introduce computational overhead Limited real-world implementation
[35]	Effective use of NSGA-II for Multi-Objective Optimization Focus on multiple QoS criteria Scalability is addressed	Lack of consideration for privacy and security NSGA-II approach requires careful parameter tuning (population size and crossover rate), which may be complex and affect performance Lack of comparison with other optimization methods
[36]	The use of social networks in IoT device management and service composition is unique Enhanced flexibility and scalability Efficient device discovery Dynamic service composition	Complexity in large-scale implementations Privacy and security considerations are not addressed. The approach assumes that IoT devices can effectively mimic social network relationships, which may not always be applicable in highly heterogeneous IoT environments
[37]	Novel use of Social IoT (SIoT) concepts Combining collaborative filtering with SIoT relationships provides a more comprehensive recommendation mechanism Emphasis on trust and reliability Enhanced user satisfaction	Data privacy and security risks Scalability concerns in large networks computational overhead Cold-start issues for new devices occur if there is insufficient data to establish trusting relationships or social connections
[38]	The application of a bandit approach to IoT service composition is innovative Adaptability to heterogeneous environments Consideration of QoS in decision-making Intelligent service composition using a reinforcement learning approach	Limited scalability analysis Computational overhead
[39]	Innovative use of context-awareness Integration of recommendation mechanisms Enhanced user experience The system is designed to adapt to a variety of smart environment applications Quantitative evaluation	Limited exploration of privacy concerns High computational requirements Contextual data can often be incomplete, which impacts the system's effectiveness Lack of real-world testing
[40]	The paper focuses on web API discovery within the social Web of Things, making the research highly relevant and forward-looking The system potentially enhances interoperability within IoT networks by enabling easier API discovery By emphasizing personalized recommendations, the approach addresses the diverse preferences of IoT users	Limited evaluation metrics Scalability concern is not discussed Privacy concern is not discussed
[41]	Relevance to IoT mashup applications Personalized QoS recommendations The combination of RTM and factorization machines is a well-considered choice	Combining RTM and factorization machines increases the model's complexity, which could lead to overfitting Limited real-world deployment
[42]	Novel approach to social IoT service recommendation Consideration of device relationships Improved service accessibility Quantitative evaluation to validate the model's feasibility	The system may fail to provide effective recommendations if the devices are poorly interconnected The scalability challenge is not addressed Privacy and security concerns are not addressed Recommendation processes can add computational and communication overhead, especially in real-time applications

5. THE COMPARISON OF THE DIFFERENT APPROACHES PROPOSED FOR SERVICE COMPOSITION IN IOT

The different approaches that have already been presented, along with their characteristics with other issues, are shown in Tables 3-8. Our comparison relies on the clearly defined criteria mentioned earlier.

Table 3. Comparative study between approaches (1)

Criteria	Sub Criteria	Asghari et al. [14]	Razian et al. [15]	Pavan Kumar et al. [16]	Baker et al. [17]	Zhang et al. [13]
Privacy	Hardware Layer	+	-	-	-	-
	Protocol layer	+	-	-	-	-
	Application layer	+	-	-	-	-
Security	Service level	-	-	-	-	-
	Composition level	-	-	-	-	-
Scalability	Horizontal	-	-	-	-	-
	Vertical	-	-	-	-	-
	Functional	-	-	+	+	-

Service selection		An SFLA-GA hybrid evolutionary method to select the best candidate composite service.	The ARC (Anomaly-aware Robust service Composition) algorithm chooses services that align with the user's needs by minimizing response time and cost.	Genetic Algorithm	The multi-cloud IoT service composition method (E2C2) creates an energy-aware composition strategy by choosing a few IoT services.	The top-k QoS optimal service composition method is selected using a service dependency graph.
Heterogeneity	Semantic	?	+	?	?	+
	Technological	?	+	?	?	+
Energy consumption	Environmental	-	-	-	+	-
	Network communication	-	-	-	+	-
QoS estimation		Response time, Availability, Reliability, Throughput, Latency, Success ability, Compliance, Best practice	Response time, Availability, Reputation	Cost (C), Reliability (R), Availability (A), Response time (Rt), Throughput (Th), and Reputation (Rp)	Energy consumption	Response time
Anomaly detection		-	+ (anomaly detection technique using machine learning)	-	-	-
Interoperability	Internal	+	?	+	+	+
	External	+	?	+	+	+
Trust management		-	-	-	-	-
Monitoring		-	-	-	-	+
Adaptability	Type	-	Reactive	-	-	-
	Mechanism		Using machine learning anomaly detection to identify erroneous QoS data, address uncertain services, and adapt with another candidate.			
	Time		Runtime			
	Implementation of adaptive code		The adaptive code is installed in the engine that runs the composition.			
Composition mechanism	Centralized Orchestration	-	+	-	+	-
	Decentralized Orchestration	+	-	-	-	-
	Choreography	-	-	+	-	+
Optimization		QoS optimization uses a hybrid evolutionary algorithm, the Shufed Frog Leaping Algorithm (SFLA), and a Genetic Algorithm (GA).	A mathematical optimization model that can handle uncertain QoS value by minimizing cost considering user restrictions.	Apply a decision tree and a genetic algorithm (GA) to provide the best possible QoS-aware IoT service composition.	Optimization of the final composition using energy efficiency as the main parameter for IoT resources.	The service dependency graph constructs a connection between services and minimizes traversal space by implementing efficient filtering techniques.
Performance		Compared to the GA, the SFLA-GA	When the suggested approach is	The execution cost is significantly	This algorithm was evaluated by performance	Tests demonstrate that this method can

		outperformed it, indicating a 90% efficiency rate for the suggested approach.	compared to previous research, including information theory-based and clustering-based methods, the results show that it produces 14.55% of the average improvement in discovering optimal solutions.	decreased by this method. The assessments show a significant increase in matching with user needs.	comparison with other algorithms, including All Cloud, Base Cloud, Smart Cloud, and COM2. The simulation results showed that the suggested method identified the minimum number of services needed to find the ideal, energy-efficient composition.	provide improved time performance and consistent accuracy.
Service representation		Abstract service	Abstract service	Abstract service	XML configuration file	As a node in the service dependency graph
Implementation tools		QWS dataset, C# language and using Azure cloud environment in Visual Studio 2017 version and NCSS 12.0.2 version, SPSS software.	Real-world QoS dataset, IBM ILOG CPLEX Optimizer, sci-kit-learn machine learning library in Python sci.	Random QoS datasets.	The experiments were executed on an Apple iMac, NetBeans 8.1(prototype development platform), Java EE 8 (programming language).	WSBen tool Java language.

Table 4. Comparative study between approaches (2)

Criteria	Sub criteria	Badidi et al. [18]	Rajendran et al. [21]	Dar et al. [22]	Kashyap et al. [23]	Razian et al. [25]
Privacy	Hardware Layer	-	-	-	-	-
	Protocol layer	-	-	-	-	-
	Application layer	-	-	-	-	-
Security	Service level	-	-	-	-	-
	Composition level	-	-	-	-	-
Scalability	Horizontal	-	-	-	-	-
	Vertical	-	-	-	-	-
	Functional	-	+	+	-	+
Service selection		An algorithm for efficiently choosing an appropriate context service that satisfies cloud service provider requirements concerning context quality and information.	Proposing an Improved Eagle Strategy algorithm by using K-means clustering (depending on a particular cluster, a service is chosen according to the non-functional characteristics).	Aiming to address the problem of integrating sensor services with conventional IT systems by applying the ideas of service orchestration and choreography.	Two meta-heuristic methods address the QoS-based service composition challenge: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).	A powerful and efficient meta-heuristic approach.
Heterogeneity	Semantic	+	+	+	?	?
	Technological	+	+	+	?	?
Energy consumption	Environmental	-	-	-	-	-
	Network communication	-	-	-	-	-

QoS estimation		QoC attributes for context service providers (Freshness, Precision, Probability of correctness).	Availability, response time, and performance	-	Execution time(t), service cost(c) and reliability(r)	Cost, response Time, availability, reputation
Anomaly detection		-	-	-	-	+
Interoperability	Internal	+	+	+	+	+
	External	+	+	+	+	+
Trust management		-	-	-	-	-
Monitoring		+	+	-	+	-
Adaptability	Type	Proactive	-	Reactive	-	Proactive
	Mechanism	Adaptation of the service composition plan using the adaptation engine about the user's profile, preferences, and context information.		Reconfiguration of a part or a complete service composition during the design or execution phase to create a more adaptable composition process.		QoS values are used in the QoS modeling to determine the utility of each proposed service. This module uses Isolation Forest, a machine-learning anomaly detection technique, to remove the inaccurate QoS data.
	Time	Runtime		Runtime		Design phase
	Implementation of adaptive code	The service composition and the adaptation process apply a standard description model.		The adaptation model is instantly deployed after being converted into executable code.		The adaptive code is installed in the anomaly detection component for abnormal historical QoS data.
Composition mechanism	Centralized Orchestration	-	-	+ (local orchestration process)	?	+
	Decentralized Orchestration	-	-	-	?	-
	Choreography	+	+	+ (global choreography process)	?	-
Optimization		-	Dynamic service selection improves the response time.	-	The service composition multi-objective issue is optimized using meta-heuristic search algorithms, which give equal weights to each of the three QoS metrics.	An efficient optimization technique called ACFS selects CloudIoT services from many candidate services to reduce costs.
Performance		Considering an application scenario where the temperature reading at the mobile user's location is the necessary context information. Freshness, accuracy, and probability of	The simulation results showed that the suggested strategy (IES) performed better than other approaches, such as WOA and PSO.	The suggested solution's scalability is examined considering the large number of resources to be composed. The system's accuracy, efficiency, and response time are also checked. After the	Two QoS criteria, response time and reliability, were used to empirically evaluate the data set for ten tasks and 40 candidates. The results show that using meta-heuristic search algorithms to solve the SC	After comparing the suggested strategy with previous research, including information theory-based and advertised QoS-based techniques, the results demonstrate an average improvement of

		correctness are the three QoC properties considered during this procedure to determine a good context provider for temperature provisioning.		composition application is effectively installed, the associated adaptability, reconfigurability, and awareness aspects will also be examined.	problem in IoT is feasible.	30.64% in QoS value for a composite plan with equivalent or even lower cost.
Service representation		?	RESTful services	RESTful services	?	Abstract service
Implementation tools		Application scenario	WS-DREAM repository+Microsoft Azure Machine Learning Studio+Microsoft Visual Studio+Netlogo tool	SensLab platform	Real-world data sets	Real-world datasets+ MATLAB R2018b+ sci-kit-learn machine learning library in Python

Table 5. Comparative study between approaches (3)

Criteria	Sub criteria	Alsaryrah et al. [26]	Yang et al. [27]	Berrani et al. [28, 29]	Yang et al. [30]	Kashyap et al. [31]
Privacy	Hardware Layer	-	-	-	-	-
	Protocol layer	-	-	-	-	-
	Application layer	-	-	-	-	-
Security	Service level	-	-	-	-	-
	Composition level	-	-	-	-	-
Scalability	Horizontal	-	-	-	-	-
	Vertical	-	-	-	-	-
	Functional	+	-	-	-	-
Service selection		The suggested selection strategy converted the issue into a bi-objective optimization problem, maximizing the QoS and minimizing energy consumption.	Composition algorithm based on Petri nets, called FindTOptimal, to determine the most optimum composition method, which includes a high user satisfaction rate (high RTC value).	Considering devices, services, targets, and requests as agents, the composer agent selects suitable service agents based on the target.	The improved Genetic Algorithm.	Hyper-heuristic evolutionary algorithm (HypEA).
Heterogeneity	Semantic	+	+	+	+	+
	Technological	+	+	+	+	+
Energy consumption	Environmental	+	-	-	-	-
	Network communication	+	-	-	-	-
QoS estimation		Execution time, cost, and energy consumption.	Reliability, response time, and cost.	-	Cost, response time, reliability, reputation, and geographic location.	Execution time, Execution cost, Reliability.
Anomaly detection		-	-	-	-	-
Interoperability	Internal	+	+	+	+	+
	External	+	+	+	+	+
Trust management		-	-	-	-	-
Monitoring		-	-	+	-	+
Adaptability	Type					
	Mechanism	-	-	-	-	-
	Time					

	Implementation of adaptive code					
Composition mechanism	Centralized Orchestration	+	-	+	-	-
	Decentralized Orchestration	-	-	-	-	-
	Choreography	-	+	-	+	+
Optimization		The best possible balance between the QoS and the energy consumed in the IoT service composition.	A composition path with an effective cost is found by addressing the FindTOptimal algorithm.	-	The service composition scheme with the highest overall service function value is considered the optimal solution in this approach.	The optimal service composition in IoT by considering three QoS attributes.
Performance		BOSC, EPC, and QoSC measured the composition lifetimes. The tests demonstrate that the suggested method effectively selects smart and provides an ideal balance between the QoS data and energy consumed.	The performance of FindTOptimal was compared to two other methods. Experimental results demonstrated the algorithm's accuracy and robustness.	The use-case scenarios demonstrated the effectiveness of a multi-agent system for Service Composition in IoT.	Improved genetic algorithms are a feasible method of optimizing IoT service composition.	The experiment's findings show that with 25 test inputs, the hyper-heuristic technique can identify service compositions that are approximately four times more accurate than GA.
Service representation		Abstract service	?	Abstract service	?	?
Implementation tools		Java language, synthetically generated data.	MATLAB	Netlogo language	The software development environment is VS2010, based on Windows XP.	Generated data sets, MATLAB 7.6.0

Table 6. Comparative study between approaches (4)

Criteria	Sub criteria	Asghari et al. [32]	Li et al. [33]	Urbieta et al. [34]	Kashyap et al. [35]	Chen et al. [36]
Privacy	Hardware layer	+	-	-	-	-
	Protocol layer	+	-	-	-	-
	application layer	+	-	-	-	-
Security	Service level	+	-	-	-	-
	Composition level	+	-	-	-	-
Scalability	Horizontal	-	-	-	-	-
	Vertical	-	-	-	-	-
	Functional	+	-	-	-	+
Service selection		Data mining techniques to identify the best composite service.	The quality attributes of the service composition can be analyzed using probabilistic model checking.	wEASEL (Context-Aware Web Service Description Language) enables the description of services.	A multi-objective meta-heuristic search technique is used to find the best solution for the service composition problem.	Service selection strategies are based on the social network model. Three distinct service search algorithms operate independently in location, type, and correlation contexts. In parallel, each

						algorithm provides the service broker with the results of local selection.
Heterogeneity	Semantic	+	+	+	?	+
	Technological	+	+	+	?	+
Energy consumption	Environmental	-	+	-	-	-
	Network communication	-	+	-	-	-
QoS estimation		Cost and time	reliability and cost	-	Execution time, Reliability	-
Anomaly detection		+	-	-	-	-
Interoperability	Internal	+	+	+	+	?
	External	+	+	+	+	?
Trust management		-	-	-	-	-
Monitoring		-	-	-	+	-
Adaptability	Type		Reactive	Proactive		
	Mechanism	-	If the selected service fails, the successful execution probability can notify the system to determine another candidate service.	The requested service conversation can be modified through reshuffling capabilities by using adaptive task conversation reshuffling.	-	-
	Time		Runtime	Design phase		
	Implementation of adaptive code		?	?		
Composition mechanism	Centralized Orchestration	-	-	-	?	+
	Decentralized Orchestration	-	+	-	?	-
	Choreography	+	-	+	?	-
Optimization		Determine the patient's preferences to select the best composite health/medical service (reduce cost and time).	The best composite service is based on reliability and cost.	-	IoT's Service Composition Problem (SCP) can be optimally solved using a multi-objective meta-heuristic search algorithm.	-
Performance		The results show that the suggested approach provides a functional scenario for an efficient diagnosis and disease prediction and gives patients an appropriate medical recommendation (health-medical service composition).	PRISM version 4.0.1 executes the verifications with three checks: reaching capability, execution time probability, and estimated cost value.	The evaluation demonstrates that the suggested wEASEL-based system executes composition more accurately than other techniques. The proposed system enables end-users to find and explore more composition opportunities.	A range of 10 to 50 candidates is used to display and evaluate this approach's performance. The optimal result was obtained by considering the execution time and reliability parameters.	According to the results, the increasing speed of the execution time decreases significantly with the growth of the search space compared to the exponential solution. The outcomes demonstrate that the search algorithm can effectively select services.
Service representation		Abstract service	?	wEASEL model	?	RESTful web service

Implementation tools		IoTD online dataset+ Weka 3.6	PRISM version 4.0.1,	Raspberry Pi 3 Model B	MATLAB R2017b +randomly generated datasets	Open-source IoT dataset

Table 7. Comparative study between approaches (5)

Criteria	Sub criteria	Bao et al. [19, 20]	White et al. [24]	Bouazza et al. [37]	Wanigasekara et al. [38]	Faieq et al. [39]
Privacy	Hardware Layer	-	-	-	-	-
	Protocol layer	-	-	-	-	-
	Application layer	-	-	-	-	-
Security	Service level	+	-	-	-	-
	Composition level	+	-	-	-	-
Scalability	Horizontal	+	-	-	-	-
	Vertical	-	-	-	-	-
	Functional	+	-	-	-	-
Service selection		The aim is to identify the most reliable service providers to maximize the utility score, which indicates the quality of the service composition.	The predictive composition approach selects the best flow using predicted values produced by collaborative filtering.	A hybrid recommender approach using ontology and collaborative filtering to recommend the most suitable services to users in an IoT environment.	Using the extended LinUCB-Hybrid algorithm	Collaborative filtering framework that uses auto-encoders to predict the QoS values and select the service with the best-predicted value.
Heterogeneity	Semantic	+	+	+	+	+
	Technological	+	+	+	+	+
Energy consumption	Environmental	-	-	-	-	-
	Network communication	-	-	-	-	-
QoS estimation		QoS trust metrics (honesty trust, Cooperativeness trust, and community-interest trust).	Response Time, Throughput.	-	-	The accuracy of the predicted values.
Anomaly detection		+	-	-	-	-
Interoperability	Internal	+	+	+	+	+
	External	+	+	+	+	+
Trust management		+	-	-	-	-
Monitoring		-	-	-	-	-
Adaptability	Type	Reactive	Reactive		Reactive	
	Mechanism	A novel adaptive filtering method that minimizes trust estimation error by dynamically adjusting the parameters of the trust protocol.	Online prediction techniques have been proposed to detect QoS anomalies and service failures in the currently running services. The goal-driven model may make adaptation decisions using QoS prediction, which enables the execution process to adapt dynamically.	-	Adaptation to changing context information and finding contextually suitable services.	-
	Time	Runtime	Runtime		Runtime	

	Implementation of adaptive code	The adaptive code that executes the composition operates in the engine.	?		?	
Composition mechanism	Centralized Orchestration	?	-	?	-	-
	Decentralized Orchestration	?	-	?	-	-
	Choreography	?	+	?	+	+
Optimization		The efficiency of the suggested trust management protocol was demonstrated through an IoT service composition application. According to the results, trust-based service composition with a recommendation system outperforms random.	The predictive composition method selects the best service composition using predicted values produced by collaborative filtering.	-	The LinUCB-Hybrid method aims to enhance performance by optimizing partial feedback.	Optimize the accuracy of the predicted values.
Performance		According to the findings, trust-based service composition performs better than random service composition and gets close to the best possible performance based on ground truth.	The response time of the service composition is between 1 to 100 users, with and without prediction: the prediction composition outperforms the composition without prediction.	User preferences are predicted using collaborative filtering. The results demonstrated that the suggested hybrid algorithm outperforms the other methods considering accuracy and performance.	The findings show that new users can benefit from the good recommendations of the suggested bandit strategy. The LinUCB-Hybrid technique can facilitate generating an appropriate IoT service.	The system works with developers and domain experts to suggest pertinent tasks that should be integrated into service models. The results demonstrate the efficacy of the proposed recommendation method.
Service representation		?	?	?	RESTful style	?
Implementation tools		Real dataset, ns3 simulator	QoS dataset	Python 3.7 with Anaconda distribution.	Crowd-sensed data, Ambient Dynamix	AutoRec framework, WSDREAM.

Table 8. Comparative study between approaches (6)

Criteria	Sub Criteria	Meissa et al. [40]	Cao et al. [41]	Chen et al. [42]
Privacy	Hardware Layer	-	-	-
	Protocol layer	-	-	-
	Application layer	-	-	-
Security	Service level	-	-	+
	Composition level	-	-	+
Scalability	Horizontal	-	-	-
	Vertical	-	-	-
	Functional	-	-	-
Service selection		An enhanced recommendation approach (Personalized Web API Recommendation PWR) aims to select APIs and offer users specific recommendations without decreasing accuracy.	A QoS-aware service recommendation based on relational structure and factorization machines was proposed for IoT Mashup applications. The aim is to	Selections based on access service recommendation scheme.

			select the top N web API recommendations.	
Heterogeneity	Semantic	+	+	+
	Technological	+	+	+
Energy consumption	Environmental	-	-	+
	Network communication	-	-	+
QoS estimation		-	Co-occurrence and popularity of services in historical Mashup applications	Trustworthiness
Anomaly detection		-	-	-
Interoperability	Internal	+	+	+
	External	+	+	+
Trust management		-	-	+
Monitoring		-	-	-
Adaptability	Type Mechanism Time Implementation of adaptive code	-	-	-
Composition mechanism	Centralized Orchestration	-	-	?
	decentralized Orchestration	-	-	?
	Choreography	+	+	?
Optimization		The Receiver Operating Characteristic (ROC) metric is used to evaluate the predicted accuracy of the suggested model.	Reducing the overall error between the actual and predicted value.	The suggested approach produces the lowest possible recommendation error.
Performance		The suggested model PWR performs better in prediction accuracy and performance than the other models (UPCC, IPCC, UC-KNN, IC-KNN, PHR, PMI, and PHS). The tests also demonstrate that the proposed model has the highest score and increases recommendation accuracy by 12%.	The results demonstrate that the proposed approach considerably improves precision, recall, and F-measure compared with other existing recommendation systems.	The results demonstrate the efficiency of the proposed model, particularly the rating accuracy, dynamic behaviour, network stability, and service composition quality.
Service representation		?	?	?
Implementation tools		MovieLens 20M dataset	Crawled Dataset+ service network platform	Simulative IoT environment

1: **+** Solved problem; 2: **-** Unsolved problem; 3: **?** Not mentioned

6. DISCUSSION

This paper reviews the latest prominent solutions for service composition in IoT. It reviews 28 selected approaches according to the mentioned criteria. It also discusses and describes their service selection methods, composition mechanism, performance, and other issues. Currently, the most proposed solutions focus on the service composition mechanism, which defines behavior by workflows using the choreography technique. Tables 3-8 show that only a few approaches consider security or privacy protection by applying different strategies, like in Asghari et al. [14], Asghari et al. [32], Bao and Chen [19], and Chen et al. [42].

The criterion of scalability becomes a critical requirement for implementing Internet of Things systems. Most research approaches focus on functional scalability like in Pavan Kumar et al. [16], Baker et al. [17], Rajendran et al. [21], Dar et al. [22], Razian et al. [25], Alsaryrah et al. [26], Asghari et al. [32], and Chen et al. [36].

Specifically, it allows for the composition of unlimited services without negatively affecting global system features like monitoring, maintenance, and performance.

Considering the trustworthiness of services is crucial to allowing users to choose reliable resources in Internet of Things environments. A few studies on trust management for IoT environments have been conducted, including those by Bao and Chen [19, 20], and Chen et al. [42].

Each service composition solution concentrates on different aspects of the IoT system like the capacity to exchange and explore information between several heterogeneous objects or systems. All approaches satisfy the heterogeneous and interoperability issues.

The goal of the studies in Baker et al. [17], Alsaryrah et al. [26], and Chen et al. [42] aim to reduce the energy consumption of service composition.

Concerning the adaptability issue, the approaches of Razian et al. [15], Badidi et al. [18], Dar et al. [22], White et al. [24], Razian et al. [25], Li et al. [33], Urbieta et al. [34], and Wanigasekara et al. [38] considered to respond to three questions related to adaptability: How, where, and when the adaptation takes place.

This comparison of the presented approaches in IoT has exposed the different techniques for service composition. We can divide these approaches into seven (07) distinct categories: agent-based, heuristic-based, QoS-based, probabilistic-based, social network-based, Petri net-based, and recommendation-based. In addition, we have deducted the differences between these techniques according to two important criteria (execution time, and resource efficiency) as follows:

(1) Execution Time:

Heuristic-based mechanisms generally offer low execution times as they apply problem-specific optimizations, especially for tasks that are not computationally intensive.

Probabilistic-based and Recommendation-Based mechanisms may have medium execution times for the following reasons: probabilistic mechanisms are often lightweight and adapt to dynamic network conditions, while recommendation-based approaches leverage historical data to handle increasing service demands.

Agent-Based and QoS-Based mechanisms may have medium to high execution times, particularly as the network grows due to increased interactions and QoS checks.

Petri Net-Based mechanisms can have higher execution times, especially in larger systems, due to the overhead in managing detailed models and ensuring synchronization.

Social network-based: social relationships can introduce delays due to the need to gather and analyze data about service interactions, service history, and user preferences.

(2) Resource Efficiency:

The agent-based mechanism is moderate to highly resource-efficient. The efficiency depends on the agent's ability to adapt to changing resource conditions. Agents can optimize resource usage by dynamically selecting the most efficient services. However, the overhead associated with agent communication and decision-making can reduce efficiency in resource-constrained environments.

Petri Net-based mechanisms can be resource-intensive due to the computational requirements for managing complex workflows, especially in resource-limited IoT devices.

The heuristic-based mechanisms are moderate for resource efficiency. These methods often balance between computational complexity and efficiency, but they may not always find the globally optimal solution, especially in resource-constrained environments.

A QoS-based mechanism is highly resource-efficient. This method can lead to efficient service compositions by focusing on service parameters like resource consumption.

The probabilistic-based mechanism is moderate to highly resource-efficient. This technique can offer resource-efficient solutions by considering probabilistic factors. But, the overhead of maintaining probabilistic models can reduce overall efficiency.

The social network-based mechanisms are highly resource-efficient. These methods can reduce unnecessary resource consumption, as services can be selected based on previous experiences and reputations.

The recommendation-based mechanism is moderately resource-efficient. However, the computational overhead of maintaining recommendation algorithms can impact resource efficiency, especially in real-time applications.

The methodology of each technique, its applicability, and its limitations are shown in Table 9.

Table 9. Methodology, applicability, and limitations for each mechanism

Mechanism	Example of a Simulation Study	Methodology	Applicability	Limitations
Agent-Based	[28]	Based on computational models that simulate the behaviors and interactions of autonomous agents and evaluate their impact on the service composition process.	It is relevant to dynamic, decentralized IoT environments due to its high quality for task control [57]. It is effective in systems that require autonomous interactions and adapt to changing environments. In IoT networks of moderate size, agent-based systems can manage scalability well.	It challenges high messages transmitted between agents and composition time in large networks. Developing and managing the agents can be complex and computationally intensive. Agents may make inconsistent decisions, especially in highly heterogeneous IoT environments. Resource constraints in IoT devices can restrict the complexity of agent-based solutions.
Heuristic-Based	[23, 25, 26, 30, 31, 35]	It solves the IoT service composition problem as a multi-objective optimization to satisfy the user's QoS requirement.	This method efficiently finds high-quality service composition plans and optimization [57]. It is particularly effective in dynamic and changing environments in IoT. It can manage complex decision-making processes that involve multiple objectives. It provides reasonable solutions when trade-offs between different QoS attributes are required.	It does not guarantee the best possible solution; it often provides good but not necessarily optimal results. Scalability challenges Many heuristic approaches are based on historical data and may not adapt well to new situations. Complexity in heuristic rules.
QoS-Based	[14-17, 26]	It is based on QoS evaluation for composite services, which generally requires a mathematical model.	It is essential in applications where performance metrics (latency, throughput, and reliability) are critical, such as in healthcare. It can adaptively select services	IoT systems are composed of heterogeneous devices, which can complicate QoS management. Complexity of dynamic QoS changes.

			based on real-time QoS metrics. It helps in filtering services based on user-defined QoS parameters. It facilitates the optimization of multiple attributes simultaneously.	Balancing the trade-offs between QoS parameters may complicate decision-making. Definitions of QoS metrics are often lacking across different IoT platforms and devices, making it difficult to implement this mechanism.
Probabilistic-Based	[33]	It specifies the desired quality attributes of the service composition and enables us to define different characteristics formally.	It is well-suited for IoT environments that have frequent changes and uncertainties. In IoT systems where service reliability is not guaranteed, this model can help estimate the likelihood of service success and optimize service selection accordingly. It is very expressive for modeling and analyzing service composition methods with probabilistic behavior. It assists in distributing workloads across services based on probabilistic assessments of service performance.	Complexity of modeling Evaluating and processing probabilistic models can introduce computational overhead. It does not guarantee that a selected service composition will meet reliability expectations, which can be critical in applications such as safety-critical systems.
Social Network-Based	[36]	Considering the social connections among IoT services, heterogeneous IoT devices can collaborate to satisfy user needs automatically.	It is well-suited for IoT environments where devices act autonomously but collaboratively. It is inherently adaptable, making it suitable for IoT networks with high device mobility. It can establish trust based on past interactions, reliability, and reputation, which is useful in applications like healthcare.	The complexity of managing social relationships can become computationally intensive. Not all IoT applications have well-defined social relationships. Creating artificial social relationships where none naturally exist can reduce effectiveness.
Petri Net-Based	[27]	It uses Petri nets, which represent processes in each phase of the service composition task, the internal functions of services, and interactions between them.	Petri Nets are ideal for IoT systems with complex workflows that require concurrent processes. Petri Nets support formal verification of system properties which is advantageous in critical IoT applications (healthcare). Petri Nets work well for moderate-sized IoT systems	Scalability challenges in Large Networks. Adaptability challenge to dynamic environments Verifying properties in a large-scale petri net model can be computationally intensive. Petri Nets traditionally focus on process flow rather than QoS attributes like latency, and reliability. Other approaches may be more appropriate in IoT systems where quality of service is a primary concern.
Recommendation-Based	[19, 24, 37-42]	Use algorithms to suggest users with more accurate services based on user preferences, historical interactions, or contextual information in service composition.	Recommender systems can be helpful in many aspects of service composition, especially when information extraction from sparse and incomplete data is required [39]. IoT systems in smart cities benefit from the contextual awareness provided by recommendations. Recommender systems can solve information overload which results from the growing number of IoT services.	Managing large-scale recommendations becomes computationally expensive. In IoT systems where users may not actively provide feedback, the algorithms may lack the necessary input to refine recommendations. Adaptability issues in dynamic environments

7. OPEN ISSUES

This review leads us to believe that the following directions can be pursued for future research on service composition in IoT environments:

(1) Privacy protection is critical due to the vast amounts of

personal and sensitive data IoT devices collect and transmit. The most important technical methods for enhancing privacy protection of service composition in IoT systems are as follows:

- Data encryption and secure communication protocols: encrypting data transmitted between devices ensures that only authorized entities can decrypt and understand it (like symmetric encryption).
- Access control mechanisms: are essential to regulate who can access devices, data, and network resources within an IoT environment. The primary access control approaches used in IoT are: Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), and Capability-Based Access Control (CapBAC)
- Blockchain and distributed ledger technology: can secure IoT data by recording it in an immutable ledger, preventing unauthorized alterations or breaches.
- Machine learning models can detect unusual patterns in IoT data flows, identifying potential security or privacy threats.
- Edge and Fog Computing for localized privacy control: processing data at the edge or fog layer (closer to IoT devices) minimizes data exposure and allows immediate privacy control.
- Privacy-aware service composition frameworks handle privacy policies, data flow control, and service interactions while ensuring privacy compliance.
- Data anonymization techniques can help anonymize IoT data before sharing with composed services.

(2) Context-aware processing: could enhance the Internet of Things by providing new information for new applications and improving knowledge construction from intelligent objects. Additionally, context-aware processing restores the original use of ontologies as sources of knowledge; however, this is still an unresolved issue legally attached to IoT intelligent objects. The most important technical methods for this direction in IoT systems are as follows:

- Machine learning models help predict future states or changes in context.
- Semantic modeling and ontologies specify relationships and concepts in a form that systems can understand and interpret.
- Edge and Fog Computing: in context-aware systems, processing data at the edge or in the fog layer improves latency and increases responsiveness, which is vital for real-time applications.
- Automated planning techniques allow IoT systems to decide based on dynamic contexts.
- Distributed context-aware systems with blockchain: provides a decentralized approach to managing context information securely across IoT devices.

(3) Energy consumption relates to the enormous amounts of energy used to stream information created by numerous interconnected devices with limited resources, which need to be efficiently stored, processed, and presented. The most important technical methods for energy consumption of service composition in IoT systems are as follows:

- Using energy-efficient communication protocols can reduce power usage
- Data reduction techniques reduce the amount of data sent to central servers, thus saving energy in transmission
- Machine learning models can predict sensor activity and optimize data collection
- Energy harvesting technologies: some IoT devices can recharge battery life by harvesting energy from their environment.
- Network optimization: energy-efficient routing protocols can optimize energy usage by balancing communication loads across nodes.

(4) Resource management: IoT nodes typically have low storage capacity and processing power despite an intense workload. Thus, practical use and management of smart devices are crucial in an Internet of Things environment. Resource management generally involves several issues that can be the subject of further research. The most important technical methods for this issue of service composition in IoT systems are as follows:

- Edge and Fog Computing: processing data at the edge distributes the computational load and minimizes network congestion.
- Dynamic resource allocation: Allocates workloads dynamically across different nodes to optimize the use of available resources, avoiding overloading.
- Machine learning algorithms can predict resource demand based on historical data, enabling proactive resource allocation.
- Using blockchain can create a decentralized framework for resource sharing, reducing the reliance on centralized management

(5) Cost: Service providers engage in interactive commercial activity with clients to establish a cost-effective supply chain and encourage ordering composite services. One of the major challenges is that IoT smart devices may compete when offering innovative services like health care monitoring. Thus, studies should consider offering cost-effective service composition techniques that satisfy other necessary QoS requirements. The most important technical methods for this direction in IoT systems are as follows:

- Edge processing reduces data transfer costs.
- Efficient data management: reduce the amount of data transmitted by data filtering techniques to minimize bandwidth costs.
- Cost-effective cloud solutions: distribute workloads across various cloud providers to prevent vendor lock-in and to take advantage of cost-effective services.
- Resource management: design devices to operate in low-power or sleep modes when not in use, extending the life of batteries and reducing operational costs.
- Data storage management: implement policies to delete old or low-value data automatically, reducing storage costs over time.

(6) Formal verification: The idea of formalization and verification shows an effective mathematical method to determine the accuracy of the service composition strategy as an NP-hard problem in the Internet of Things environment. Evaluating the correctness of the service composition process in IoT platforms, including healthcare systems, is one of the most critical concerns. The main issue of this field is providing

an effective tool for modeling and verifying the service composition techniques in IoT, taking into account a massive number of IoT services [5].

The most important technical methods for this direction in IoT systems are as follows:

- Model checking: applying model checking to the entire service composition rather than individual services.
- Formal contract verification: define expected inputs, outputs, and behaviors for each service in a composition.
- Petri Nets for service composition modeling: help to visualize and verify service workflows.
- Runtime verification monitors the actual behavior of services during execution, validating them against specified properties in real time.
- The use of constraint satisfaction techniques and logic programming guarantees that service compositions follow certain rules.

8. CONCLUSIONS

Service composition aims to coordinate the best available services to offer a new composed service for user needs that a single service cannot satisfy. Service composition in IoT is a very active field of research. This paper has presented:

(1) A synopsis of twenty-eight (28) representative research for service composition in IoT in its literature.

(2) An introduction to most particular criteria depends on the fundamental issues of service composition in an IoT environment.

(3) Various challenges and possible solutions to these criteria for service composition in IoT systems.

(4) A detailed comparative analysis between the studied approaches through well-target criteria.

(5) The most important techniques for IoT service composition and the applicability of each one of them.

(6) The most important challenges that are required for future work.

Each middleware solution for service composition in IoT focuses on different aspects, such as adaptability, energy consumption, privacy, scalability, and many more. Most approaches focus on developing a solution that gives users the best possible composite service based on their requirements.

The upcoming research on service composition for IoT environments should incorporate middleware architecture for QoS-oriented service composition by combining more than one mechanism and having an adaptability component to handle service changes without generating undesired execution interruptions or significantly decreasing the composite application's quality.

REFERENCES

[1] Yang, R., Li, B., Cheng, C. (2014). Adaptable service composition for intelligent logistics: A middleware approach. In 2014 International Conference on Cloud Computing and Big Data, Wuhan, China, pp. 75-82. https://doi.org/10.1109/CCBD.2014.10

[2] Hamzei, M., Navimipour, N.J. (2018). Towards efficient service composition techniques in the internet of things. IEEE Internet of Things Journal, 5(5): 3774-3787. https://doi.org/10.1109/JIOT.2018.2861742

[3] Li, K., Jiang, L. (2012). The research of web services composition based on context in Internet of Things. In 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, pp. 160-163. https://doi.org/10.1109/CSAE.2012.6272570

[4] Aoudia, I., Benharzallah, S., Kahloul, L., Kazar, O. (2019). Service composition approaches for internet of things: A review. International Journal of Communication Networks and Distributed Systems, 23(2): 194-230. https://doi.org/10.1504/IJCNDS.2019.101223

[5] Asghari, P., Rahmani, A.M., Javadi, H.H.S. (2018). Service composition approaches in IoT: A systematic review. Journal of Network and Computer Applications, 120: 61-77. https://doi.org/10.1016/j.jnca.2018.07.013

[6] Kashyap, N., Charan Kumari, A., Chhikara, R. (2020). Service composition in IoT-A review. Intelligent Data Communication Technologies and Internet of Things: ICICI 2019, 2019: 287-291. https://doi.org/10.1007/978-3-030-34080-3_33

[7] Rabah, B., Mounine, H.S., Ouassila, H. (2022). QoS-aware IoT services composition: A survey. In Distributed Sensing and Intelligent Systems: Proceedings of ICDSIS 2020, Switzerland, pp. 477-488. https://doi.org/10.1007/978-3-030-64258-7_41

[8] Safaei, A., Nassiri, R., Rahmani, A.M. (2022). Enterprise service composition models in IoT context: Solutions comparison. The Journal of Supercomputing, 78(2): 2015-2042. https://doi.org/10.1007/s11227-021-03873-7

[9] Han, S.N., Khan, I., Lee, G.M., Crespi, N., Glitho, R.H. (2016). Service composition for IP smart object using realtime Web protocols: Concept and research challenges. Computer Standards & Interfaces, 43: 79-90. https://doi.org/10.1016/j.csi.2015.08.006

[10] Razian, M., Fathian, M., Bahsoon, R., Toosi, A.N., Buyya, R. (2022). Service composition in dynamic environments: A systematic review and future directions. Journal of Systems and Software, 188: 111290. https://doi.org/10.1016/j.jss.2022.111290

[11] Barakat, L., Miles, S., Luck, M. (2018). Adaptive composition in dynamic service environments. Future Generation Computer Systems, 80: 215-228. https://doi.org/10.1016/j.future.2016.12.003

[12] Arellanes, D., Lau, K.K. (2020). Evaluating IoT service composition mechanisms for the scalability of IoT systems. Future Generation Computer Systems, 108: 827-848. https://doi.org/10.1016/j.future.2020.02.073

[13] Zhang, B., Wen, K., Lu, J., Zhong, M. (2021). A top-K QoS-optimal service composition approach based on service dependency graph. Journal of Organizational and End User Computing (JOEUC), 33(3): 50-68. https://doi.org/10.4018/JOEUC.20210501.oa4

[14] Asghari, P., Rahmani, A.M., Javadi, H.S. (2022). Privacy-aware cloud service composition based on QoS optimization in the Internet of Things. Ambient Intelligence and Humanized Computing Journal, 13(3): 5295-5320. https://doi.org/10.1007/s12652-020-01723-7

[15] Razian, M., Fathian, M., Buyya, R. (2020). ARC: Anomaly-aware Robust Cloud-integrated IoT service composition based on uncertainty in advertised quality of service values. Journal of Systems and Software, 164: 110557. https://doi.org/10.1016/j.jss.2020.110557

[16] Pavan Kumar, V., Shetty, S., Janardhana, D.R., Manu, A.P. (2022). Qos aware service composition in IoT using heuristic structure and genetic algorithm. Mathematical Statistician and Engineering Applications, 71(3): 750-766. https://doi.org/10.17762/msea.v71i3.215

[17] Baker, T., Asim, M., Tawfik, H., Aldawsari, B., Buyya, R. (2017). An energy-aware service composition algorithm for multiple cloud-based IoT applications. Journal of Network and Computer Applications, 89: 96-108. https://doi.org/10.1016/j.jnca.2017.03.008

[18] Badidi, E., Atif, Y., Sheng, Q.Z., Maheswaran, M. (2019). On personalized cloud service provisioning for mobile users using adaptive and context-aware service composition. Computing, 101: 291-318. https://doi.org/10.1007/s00607-018-0631-8

[19] Bao, F., Chen, R. (2012). Trust management for the internet of things and its application to service composition. In 2012 IEEE international symposium on a world of wireless, mobile and multimedia networks (WoWMoM), San Francisco, CA, pp. 1-6. https://doi.org/10.1109/WoWMoM.2012.6263792

[20] Chen, R., Guo, J., Bao, F. (2014). Trust management for SOA-based IoT and its application to service composition. IEEE Transactions on Services Computing, 9(3): 482-495. https://doi.org/10.1109/TSC.2014.2365797

[21] Rajendran, V., Ramasamy, R.K., Mohd-Isa, W.N. (2022). Improved eagle strategy algorithm for dynamic web service composition in the IoT: A conceptual approach. Future Internet, 14(2): 56. https://doi.org/10.3390/fi14020056

[22] Dar, K., Taherkordi, A., Rouvoy, R., Eliassen, F. (2011). Adaptable service composition for very-large-scale internet of things systems. In Proceedings of the 8th Middleware Doctoral Symposium, Portugal, pp. 1-6. https://doi.org/10.1145/2088960.2088971

[23] Kashyap, N., Kumari, A.C., Chhikara, R. (2020). Service composition in IoT using genetic algorithm and particle swarm optimization. Open Computer Science Journal, 10(1): 56-64. https://doi.org/10.1515/comp-2020-0011

[24] White, G., Palade, A., Clarke, S. (2018). QoS prediction for reliable service composition in IoT. In Service-Oriented Computing–ICSOC 2017 Workshops: ASOCA, ISyCC, WESOACS, and Satellite Events, Málaga, Spain, pp. 149-160. https://doi.org/10.1007/978-3-319-91764-1_12

[25] Razian, M., Fathian, M., Wu, H., Akbari, A., Buyya, R. (2020). SAIoT: Scalable anomaly-aware services composition in CloudIoT environments. IEEE Internet of Things Journal, 8(5): 3665-3677. https://doi.org/10.1109/JIOT.2020.3023938

[26] Alsaryrah, O., Mashal, I., Chung, T.Y. (2018). Bi-objective optimization for energy-aware internet of things service composition. IEEE Access Journal, 6(C): 26809-26819. https://doi.org/10.1109/ACCESS.2018.2836334

[27] Yang, R., Li, B., Cheng, C. (2014). A Petri net-based approach to service composition and monitoring in the IOT. In 2014 Asia-Pacific Services Computing Conference, Fuzhou, China, pp. 16-22. https://doi.org/10.1109/APSCC.2014.11

[28] Berrani, S., Yachir, A., Djamaa, B., Aissani, M. (2018). Multi-agent system based service composition in the internet of things. In Computational Intelligence and Its Applications: 6th IFIP TC 5 International Conference, CIIA 2018, Oran, Algeria, pp. 521-532. https://doi.org/10.1007/978-3-319-89743-1_45

[29] Berrani, S., Yachir, A., Djemaa, B., Aissani, M. (2018). Extended multi-agent system based service composition in the Internet of things. In 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa, Algeria, pp. 1-8. https://doi.org/10.1109/PAIS.2018.8598503

[30] Yang, Z., Li, D. (2014). IoT information service composition driven by user requirement. In 2014 IEEE 17th International Conference on Computational Science and Engineering, Chengdu, China, pp. 1509-1513. https://doi.org/10.1109/CSE. 2014.280

[31] Kashyap, N., Kumari, A.C. (2018). Hyper-heuristic approach for service composition in internet of things. Electronic Government, an International Journal, 14(4): 321-339. https://doi.org/10.1504/EG.2018.10013553

[32] Asghari, P., Rahmani, A.M., Haj Seyyed Javadi, H. (2019). A medical monitoring scheme and health-medical service composition model in cloud-based IoT platform. Transactions on Emerging Telecommunications Technologies, 30(6): e3637. https://doi.org/10.1002/ett.3637

[33] Li, L., Jin, Z., Li, G., Zheng, L., Wei, Q. (2012). Modeling and analyzing the reliability and cost of service composition in the IoT: A probabilistic approach. In 2012 IEEE 19th International Conference on Web Services, Honolulu, HI, USA, pp. 584-591. https://doi.org/10.1109/ICWS.2012.25

[34] Urbieta, A., Gonzalez-Beltran, S., Ben Mokhtar, S., Hossain, A. Capra, L. (2017). Adaptive and context-aware service composition for IoT-based smart cities. Future Generation Computer Systems Journal, 76: 262-274. https://doi.org/10.1016/j.future.2016.12.038

[35] Kashyap, N., Kumari, A.C., Chhikara, R. (2019). Multi-objective optimization using NSGA II for service composition in IoT. International Conference on Computational Intelligence and Data Science (ICCIDS 2019), 167(3): 1928-1933. https://doi.org/10.1016/j.procs.2020.03.214

[36] Chen, G., Huang, J., Cheng, B., Chen, J. (2015). A social network based approach for IoT device management and service composition. In 2015 IEEE World Congress on Services, New York, NY, USA, pp. 1-8. https://doi.org/10.1109/SERVICES. 2015.9

[37] Bouazza, H., Said, B., Laallam, F.Z. (2022). A hybrid IoT services recommender system using social IoT. Journal of King Saud University-Computer and Information Sciences, 34(8): 5633-5645. https://doi.org/10.1016/j.jksuci.2022.02.003

[38] Wanigasekara, N., Schmalfuss, J., Carlson, D., Rosenblum, D.S. (2016). A bandit approach for intelligent IoT service composition across heterogeneous smart spaces. In Proceedings of the 6th International Conference on the Internet of Things, Germany, pp. 121-129. https://doi.org/10.1145/2991561.2991562

[39] Faieq, S., Front, A., Saidi, R., El Ghazi, H., Rahmani, M.D. (2019). A context-aware recommendation-based system for service composition in smart environments. Service Oriented Computing and Applications Journal, 13(4): 341-355. https://doi.org/10.1007/s11761-019-00277-7

[40] Meissa, M., Benharzallah, S., Kahloul, L., Kazar, O.

(2021). A personalized recommendation for web API discovery in social web of things. The International Arab Journal of Information Technology, 18(3A): 438-445. https://doi.org/10.34028/iajit/18/3A/7

[41] Cao, B., Liu, J., Wen, Y., Li, H., Xiao, Q., Chen, J. (2018). QoS-aware service recommendation based on the relational topic model and factorization machines for IoT Mashup applications. Parallel and Distributed Computing Journal, 132: 177-189. https://doi.org/10.1016/j.jpdc.2018.04.002

[42] Chen, Z., Ling, R., Huang, C., Zhu, X. (2016). A scheme of access service recommendation for the Social Internet of Things. International Journal of Communication Systems, 29(4): 694-706. https://doi.org/10.1002/dac.2930

[43] Retima, F., Benharzallah, S., Kahloul, L., Kazar, O. (2017). A comparative analysis of context-management approaches for the internet of things. The International Arab Journal of Information Technology, Special Issue 14(4A): 578-585.

[44] Tbahriti, S.E., Ghedira, C., Medjahed, B., Mrissa, M. (2013). Privacy-enhanced web service composition. IEEE Transactions on Services Computing, 7(2): 210-222. https://doi.org/10.1109/TSC.2013.18

[45] Rouached, M. (2012). Security analysis for web services composition. International Journal of Scientific & Engineering Research, 3(5): 2229-5518.

[46] Pessoa, R.M., Silva, E., Van Sinderen, M., Quartel, D.A., Pires, L.F. (2008). Enterprise interoperability with SOA: A survey of service composition approaches. In 2008 12th Enterprise Distributed Object Computing Conference Workshops, Munich, Germany, pp. 238-251. https://doi.org/10.1109/EDOCW.2008.32

[47] Souri, A., Rahmani, A.M., Jafari Navimipour, N. (2018). Formal verification approaches in the web service composition: A comprehensive analysis of the current challenges for future research. International Journal of Communication Systems, 31(17): e3808. https://doi.org/10.1002/dac.3808

[48] Lins, F.A.A., dos Santos Júnior, J.C., Rosa, N.S. (2007). Improving transparent adaptability in web service composition. In IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07), Newport Beach, CA, USA, pp. 80-87. https://doi.org/10.1109/SOCA.2007.26

[49] Wang, W., Zhai, S., Shi, X. (2014). Towards energy-aware web service composition. International Journal of u- and e-Service, Science and Technology, 7(6): 369-380.

[50] Li, Q., Dou, R., Chen, F., Nan, G. (2014). A QoS-oriented Web service composition approach based on multi-population genetic algorithm for Internet of Things. International Journal of Computational Intelligence Systems, 7(2): 26-34. https://doi.org/10.1080/18756891.2014.947090

[51] Wang, Y., Wu, Z. (2013). Selecting skyline services for QoS-aware composition by upgrading the MapReduce paradigm. Cluster Computing Journal, 16(4): 693-706. https://doi.org/10.1007/s10586-012-0240-9

[52] Kil, H., Cha, R., Nam, W. (2016). Transaction history-based web service composition for uncertain QoS. International Journal of Web and Grid Services, 12(1): 42-62. https://doi.org/10.1504/IJWGS.2016.074180

[53] Yang, Y., Yao, H., Wang, S. (2018). An optimization method for reliable cloud service composition with low resource occupancy. In 2017 4th International Conference on Machinery, Materials and Computer (MACMC 2017), Atlantis Press, pp. 735-740. https://doi.org/10.2991/macmc-17.2018.139

[54] Quartel, D., Sinderen, M.V. (2008). Modeling and analyzing interoperability in service compositions using COSMO. Enterprise Information Systems Journal, 2(4): 347-366. https://doi.org/10.1080/17517570802422461

[55] Hayyolalam, V., Pourghebleh, B., Pourhaji Kazem, A.A. (2020). Trust management of services (TMoS): investigating the current mechanisms. Transactions on Emerging Telecommunications Technologies, 31(10): e4063. https://doi.org/10.1002/ett. 4063

[56] Sheng, Q., Qiao, X., Vasilakos, A.V., Szabo, C., Bourne, S., Xu, X. (2014). Web services composition: A decade's overview. Journal of Information Sciences, 280(2): 218-238. https://doi.org/10.1016/j.ins.2014.04.054

[57] Navimipour, N.J., Vakili, A. (2017). A comprehensive and systematic review of the service composition mechanisms in the cloud environments. Journal of Network and Computer Applications, 81: 24-36. https://doi.org/10.1016/j.jnca.2017.01.005