



Data replica placement approaches in fog computing: a review

Esmail Torabi¹ · Mostafa Ghobaei-Arani¹ · Ali Shahidinejad¹

Received: 19 August 2021 / Revised: 13 January 2022 / Accepted: 28 February 2022 / Published online: 4 April 2022
© Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Recently, we are witnessing an enormous burst of data due to the ever-increasing number of Internet of Things (IoT) devices. The traditional cloud computing paradigm has failed to scale; to be specific, its latency and bandwidth utilization are remarkably increased and consequently, Quality of Service (QoS) is decreased. On the other hand, the data management scope in fog computing require much more considerations in terms of performance and scalability. This is because of deploying IoT applications over fog nodes considering their resource-limited and heterogeneity. However, to the best of our knowledge, there is not any literature review that systematically categorizes these issues. In this paper, we have presented a classification of data replica placement approaches considering four main categories: framework-based, graph-based, heuristic-based, and meta-heuristic-based algorithms. To sum up, the primary contribution of this study is as follows: studying articles on data replica placement in fog computing, as well as presenting their strengths and weaknesses, providing a comprehensive systematic review of current approaches and categorizing them comprehensively, discussing research challenges, and future works to improve computing and evaluation mechanisms in the fog computing environment. This paper generally provides a classification, briefly explains the reviewed techniques, and then compares these methods in the end.

Keywords Data management · Data replica placement · Edge computing · Fog computing · Systematic review

Abbreviations

ACO	Ant colony algorithm	DRC-DS	Dynamic replica creation based on domain structure
ACOV	Optimized ant colony algorithm	FDA	Fog data analytics
CCWN	Centrality complex weighted networks	FLP	Facility location problem
DDS	Distributed data stores	GA	Genetic algorithm
DDoS	Distributed denial of services	GAP	Generalized assignment problem
DHT	Distributed hash table	HDFS	Hadoop distributed file system
DNS	Domain name system	IoT	Internet of things
D-Rep	Distributed—replica placement	IP	Integer programming
DSEGA	Data-intensive service edge genetic algorithm	IoV	Internet of vehicles
DRCA	Dynamic replica creation algorithm	MDP	Markov decision-making process
DRC-AH	Data replica creation based on access heat	MILP	Mixed-integer linear programming
DRC-GM	Dynamic replica creation algorithm granular and comprehensive	NFV	Network function virtualization
DRSA	Data replica scheduling algorithm	PoP	Point of presence
DRS-NSC	Data replica selection based on node service capability	QoS	Quality of service
		FNSG	Fast non-dominated sorting genetic
		RP-FNSG	Fast non-dominated sorting genetic replica placement
		SA	Simulated annealing
		SDN	Software-defined networking
		SLR	Systematic literature review
		TQ	Technical questions
		UAV	Unmanned aerial vehicles
		VM	Virtual machine

✉ Mostafa Ghobaei-Arani
m.ghobaei@qom-iau.ac.ir

¹ Department of Computer Engineering, Qom Branch, Islamic Azad University, Qom, Iran

W2H	Web to home
WIEBRS	Wireless IoT edge-enabled block replica strategy
YCSB	Yahoo cloud serving benchmark

1 Introduction

With the increase in the amount of data generated due to the growing trend in applying smart things, the conventional cloud computing paradigm has failed to scale properly; to be specific, its latency and bandwidth utilization are remarkably increased, and, consequently, quality of service (QoS) is decreased [1]. Indeed, for real-time IoT applications (e.g., Internet of Vehicles (IoV), eHealth, and Industry 4.0), conducting instant communication, or at least with a neglectable delay, is critical, as the excessive delays may cause serious problems. To cope with the aforementioned limitations of conventional cloud computing, especially for large-scale IoT applications, fog computing as a novel approach in this context has emerged [2].

Fog computing is defined as a scenario in which computing and storage, as well as communication services, are conducted through cooperating various heterogeneous nodes in the network's core which are dispersed in a geographical area, instead of just being centralized at the core data centers in the conventional cloud computing schema [3]. In this architecture, users also offer parts of their devices to host these services in exchange for incentives.

As a matter of fact, expanding the computational capacity of networks communication equipment remarkably pushed the emergence of the fog computing paradigm. The architecture of the fog is to prepare computational and stored contents at the intermediate communication equipment, through which the network nodes can perform computational services or store data. - This way, services become closer to IoT devices so that the network latency in applications is reduced remarkably. Similarly, data is stored near IoT devices, and data transmitted across the network is reduced significantly. Finally, only the processed and summarized data is sent to the remote datacenters, i.e., cloud or other IoT devices.

Due to heterogeneity and large-scale, security is one of the main challenges of IoT equipment. The impact factors can be divided into two categories: heterogeneity of things and communication of things. In each category, there are different security problems. To increase security in the IoT, the parameters and mechanisms of data privacy,

confidentiality, integrity, authentication, authorization, accounting, availability of services and energy efficiency are required. Security in IoT systems many challenges due to the different special of IoT system. Jamali et al. [4] Provided the following IoT security classification:

- Application (Authorization, Authentication, Exhaustion of Resources, Trust Establishment)
- Architectural (Authorization, Authentication)
- Communication (Main-in-the Middle Attack, Eavesdropping, Denial-of- service)
- Data (Privacy, Trust)

Admittedly, although fog computing brings a lot of advantages, it is still wrestling with the accessibility challenge. This means that IoT devices might be unable to access their required data on time, especially for real-time applications like eHealth, IoV, and Industry 4.0 cases. One feasible solution is to create replicas of the data close to the IoT devices. This course of action significantly reduces data access delay and eventually increases accessibility, but at the cost of increased data redundancy. Also, it is noteworthy that fog nodes' hardware resources (i.e., storage and computation) are often limited compared to cloud computing servers. However, there are several ways to store data replicas in the fog infrastructure:

1. *No replication* in this strategy, data is only stored in one location, and there are no replicas. No replication strategies are easy to deploy and maintain but may lead to delays in functionality.
2. *Full replication* in this strategy, there are replicas of data on all nodes. Overhead for this strategy is extremely high in terms of occupied storage space during data distribution and updates. However, this duplication strategy is promising in terms of functionality because the data round trip time has almost no transmission delay.
3. *n replication* ($n < m$ where n and m are the numbers of replicas and nodes, respectively): this strategy takes advantage of both above-mentioned strategies. This strategy gains a tradeoff between the network overhead and the delay, as it enables us to adjust the number of replicas according to the system.

Although the last data management strategy, namely n replication, would be the most rational one, and since the number of fog nodes and the amount of the data is relatively high, the data replicas placement problem is settled in the category of NP-Hard problems. Indeed, data management comprises several scopes such as data acquisition, data cleaning, data pre-processing, data processing, data storage, data exchange, data placement, and data analytics. Every aspect of data management is essential to handle the immense data generated by IoT devices momentarily. Data

replica placement is one of the most imperative scopes of study in this literature in order to enhance the functionality of fog computing environments. We have categorized the existing efforts in applying replication strategies in the fog computing environment as follows: framework-based, graph-based, heuristic-based, meta-heuristic-based approaches. These approaches aim to achieve the best performance of fog infrastructure regardless of considering the benefits of providers or meeting the users' demands.

Therefore, replication strategies in such situations should ensure both the quality of service (QoS) and the benefits of the economic provider [3]. However, the existing replication methods are not compatible with the fog environments.

In spite of the importance of data replica placement strategies in fog computing, there is no comprehensive analysis on this issue to guide the researchers who work in the data management area. Therefore, this review aims to provide a comprehensive go-through and analysis of the existing data replica placement mechanisms within fog computing. Briefly, the main contribution of this study is as follows:

- Studying articles on data replica placement in fog/edge computing, and providing a comprehensive classification, as well as presenting the strengths and weaknesses per each.
- Analyzing the proposed mechanisms based on data management methods in fog computing environments to direct to gain performance improvements in future works.
- Discuss the current research challenges to discover future works focusing on promoting the evaluation mechanisms in the fog computing environment.

The rest of this paper is organized as follows: Sect. 2 provides the key background of data management and data replica placement in fog computing. In Sect. 3, we review some related survey papers. Section 4 describes the research methodology of our study. Section 5 provides a taxonomy of reviewed papers along with a brief explanation of their methods and a comparison of these methods. Several technical questions are also given out in Sect. 6. Section 7 provides some open issues as future research directions. Finally, our conclusion is presented in Sect. 8.

2 Background

Due to differences in fog and cloud infrastructures, data and task replication are different in fog and cloud, as following:

The first difference is the efficient size of a replica in fog and cloud, which is placed in this category. Computing and

storage facility in the cloud is powerfully built but fog nodes are weaker [5]. The strength of Fog devices comes from the parallelization of workloads of data analysis. There is no device that bears a heavy task, each one of them performs only a few lightweight tasks [6].

Second, the cloud model is based on the Internet. Its usage is very easy and has access to computing resources such as servers, storage, applications, and services. Cloud computing is one of the distributed and parallel systems that includes a set of VMs and computers which are connected [7]. Therefore, the cloud replica placement location is on the Internet but fog replication location is on the edge of the network.

Third, geographical distribution for cloud stratum is centralized but in fog is decentralized and distributed [5].

Fourth, when the migration occurs in cloud system, the old replica's policy must be updated according to the new demand's accessibility. So, executing algorithms must be updated to keep the access time demands but mobility for fog replication is fully supported [8].

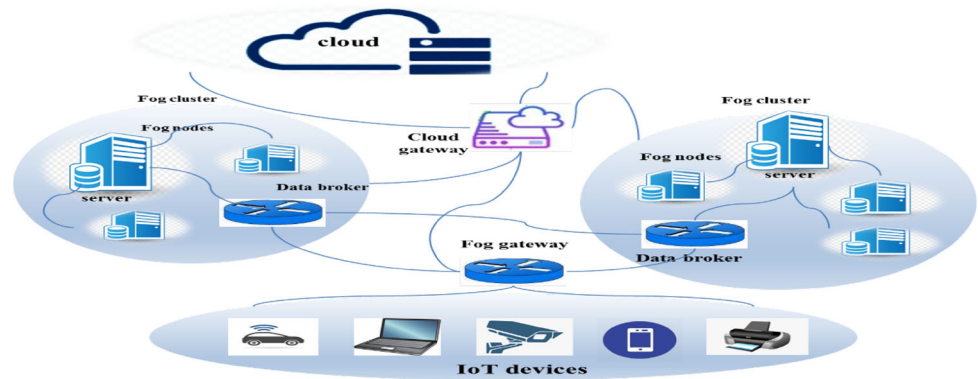
So, this section presents a quick review of data management and data replica placement in fog computing.

2.1 Data management in fog computing

Similar to the usefulness of cloud computing during the Internet utilization bursts, fog computing is highly functional in real-time data management systems within the realm of IoT. The general diagram of data management in the fog environment is shown in Fig. 1.

IoT is attracting researchers' attention more and more nowadays, although it is still struggling with the massive amount of generated data and immediately responding to the users' requests. These challenges in the IoT paradigm exist due to its infancy and lack of thorough investigation. The increasing growth rate of generating data in IoT environments is a striking issue. We can see in some studies that every day in 2012, 2500 petabytes of data, on average, was generated [9]. In a health application, with 30 million users, 25,000 records were generated per second [10]. Nevertheless, the abundance of these types of information and studies entails provisioning an effective mechanism for handling these significant loads of data, which a lot of them need an immediate reaction. Indeed, this procedure in cloud computing schema due primarily to transmit data between edge and core data centers might lead to drastic bandwidth utilization and disastrous latencies that are especially intolerable in IoT well-being applications, as delayed responses could jeopardize one's life in urgent cases. End devices and IoT sensors often generate duplicated data periodically, which might contain useless and redundant entries. So, transferring these

Fig. 1 Data management in fog computing



gigantic amounts of data would increase the error rate and cause packet loss and data interference.

In this paper, we have covered two basic concepts in data management. First is the life cycle of data, and another is the specifications and type of that data.

2.1.1 Data life cycle

Fog data life cycle is a loop that begins with data collection at the device layer where data is created and transferred to the higher layers and culminates in returning the proper response to the device layer and executing the necessary commands [2]. As it is shown in Fig. 2, we consider five general steps: data collection, pre-processing, basic analysis, collecting feedback, and command execution, which all have been explained in the following.

2.1.1.1 Data collection Data from different devices are collected to be sent to the top layers. This is realized by

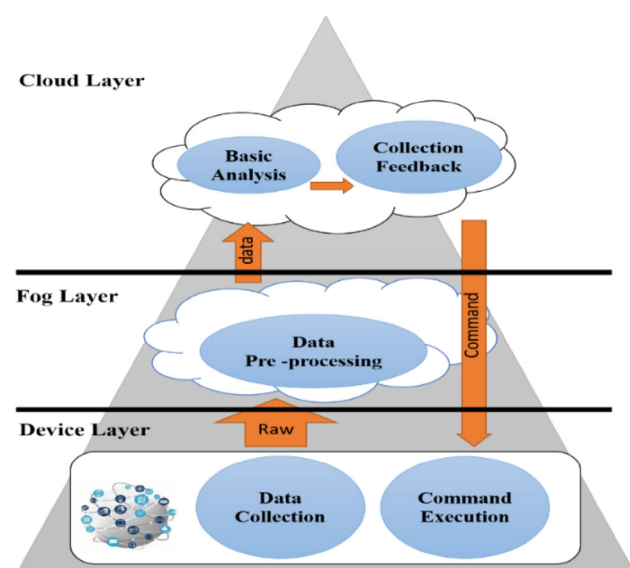


Fig. 2 Data life cycle model

employing a local port, communication node, and/or special sensors for sending data directly to the fog layer [9].

2.1.1.2 Data pre-processing At this stage, small changes are made in data, and they are processed locally. Collecting and filtering data, deleting duplicated and trivial data, cleansing data, and compressing or decompressing data, as well as exchanging data, and analyzing various patterns, are all accomplished in the pre-processing stage. Data from the previous stage is available locally on fog devices because those are stored on fog devices for a period of time and remarkably helps in data pre-processing. The data collected by the network will be transferred to the cloud layer and a report as a response is sent back to the edge device. Once the report is received by fog nodes from the cloud layer, it is transferred to the device layer, which might require decryption, or compression, or special operations before sending. Therefore, the fog layer must support this series of operations [11].

2.1.1.3 Basic analysis Network users might have access to a variety of reports or data analyses depending on their demands. Therefore, the data received in the cloud layer is stored permanently and processed according to defined demands. Various types of analyzes and processes are performed on this data, generally having a large volume, to extract valuable information. Therefore, to process this amount of data, they need to manage big data and use mapping technologies such as Hadoop distributed file system (HDFS) [12].

2.1.1.4 Collecting feedback Actuators must trigger the proper action based on the received data. In this way, appropriate feedback is sent in this situation [2].

2.1.1.5 Command execution Edge devices receive feedbacks and commands from fog nodes and execute these commands on their data [2]. The fog data life cycle of data

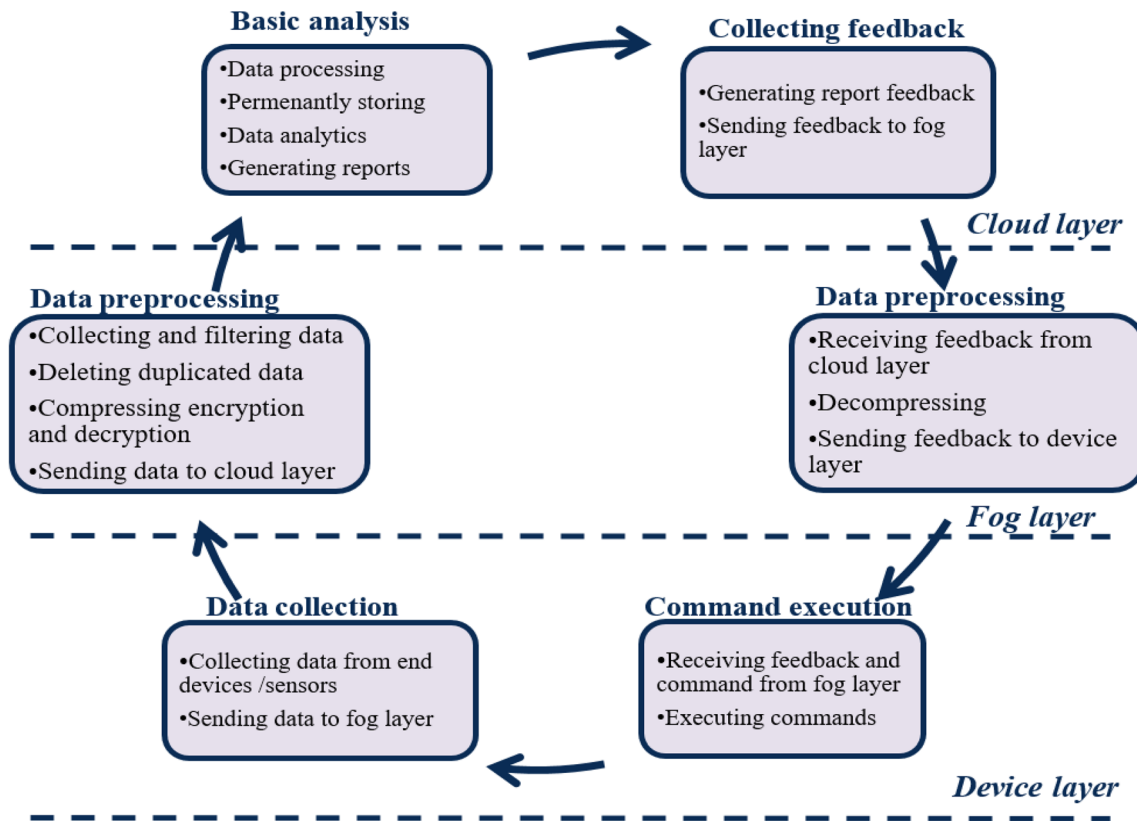


Fig. 3 Data lifecycle in fog ecosystem

management in the fog environment is shown in Fig. 3. Data lifecycle in fog ecosystem Fig. 3.

2.1.1.6 Data specifications Data quality addresses the questions of how many data specifications are in the configuration and whether it meets the consumer's needs. Data specification is essential for designing and correcting data quality and integration standards and responding appropriately to challenges during the data management process. Some of the key features of IoT data are introduced [13]. They studied validity, reliance, totality, data volume, and delay as data status specifications. Interpretability, access security and, ease of access are other additional specifications. Also, in [9], IoT data specifications are divided into three general sections: data status, data generation, and data collaboration. IoT data status covers uncertainty, redundancy, ambiguity, and incompatibility of data. Basically, after collecting data from the devices, they are stored to perform the required operations. After collecting and storing data, discrete processes begin. Therefore, new data analysis parameters would be required while data production rate and volume increase. In the following, these specifications are elaborated:

- **Heterogeneity:** The data generated by different sensors, nodes, and devices have different structures [14].

- **Inaccuracy:** There is inaccuracy and uncertainty in data collected by various sensors and IoT devices. Even, data might be incorrect [13].
- **Scalability:** In different scenarios, there is a massive amount of data generated by a large number of IoT devices [9].
- **Unacceptable semantics:** Since the format, structure, and source of collected data might vary in different cases, data processing and management would be complicated. Therefore, semantic web concepts are applied to process raw data for a better understanding of a machine. Unfortunately, most of the data collected from the devices do not carry proper semantic [9, 14].
- **Generation rate:** Data generation speed rate and frequency vary in different edge devices [9].
- **Redundancy:** Sending duplicate data by devices and sensors causes data accumulation [9].
- **Incompatibility:** Low resolution or misreading of data taken from sensors may cause incompatibility in the collected data [9].

2.1.2 Data replica placement

Data is stored on fog devices for further processing. The process of storing data requires conducting cache

management techniques to minimize the occupied spaces of end devices' cache and provide a real-time response. Decisions about the volume and duration of storing data are also made based on the underlying potential of the application and its capabilities. Another issue is to efficiently place data generated and collected in fog storage based on node specifications, geographical characteristics, and application features, as the data placement methods influence service delays. Naas et al. [15] have applied iFogStore to reduce the latency. They considered fog devices' features, as well as their heterogeneity and location. In addition, iFogStore uses storage and retrieval capabilities to reduce latency. It also benefits from sharing data by consumers who are likely to move to other locations or use devices with different infrastructural capabilities. A storage management architecture is recommended in fog computing based on a three-tier architecture for real-time decision-making. Technically, six data storage mechanisms in the management framework are data preparation, monitoring, specification list, adaptive algorithm, storage components, and intermediaries. The device layer and cloud layer are also two other layers. The former generates raw data, and the latter is in charge of storing the historical data.

The data replica placement problem mainly addresses the question of how to place delay-sensitive data in multiple replica nodes in order to reduce access delay and network bandwidth utilization significantly. In this context, data, which used to be uploaded into the centralized data centers in cloud computing schema, is stored into different replica nodes. Accordingly, different strategies are adopted in determining the number of replica nodes, including full replication and partial replication.

3 Related works

In This section, some survey articles on data replication issues in fog computing and cloud environments, as well as data storage have been investigated. Some of their advantages and shortcomings in these areas are also discussed and analyzed.

Da Silva et al. [16] have reviewed three algorithms for data placement in fog environments and evaluated them with the iFogSim simulator. Although this work appropriately compares various perspectives of the data placement problem using the simulated data, there are still some shortcomings in the reviewed paper which are as follows:

- Poor organization of the article.
- Narrow coverage of related subjects.
- Insufficient technical explanations.

Karatas et al. [17] targeted data analytics for fog-based big data. They have developed a broad taxonomy on Fog

Data Analytics (FDA), and it spreads from data collection and storage to security and privacy. However, the reviewed paper has the following shortage:

- Lack of precise details of presented subjects.

Milani et al. [18] presented a review paper that discusses approaches for the replication of data in cloud networks. They have appropriately classified these studies as static and dynamic approaches. However, the review paper has the following shortages:

- Lack of precise details of presented subjects.
- Narrow coverage of related subjects.
- Lack of presenting a systematic format to select articles.

Nikoui et al. [2] reviewed and described concepts of data management in fog domains. They have discussed the main benefits of data management in fog environments and provided a better understanding of basic processes such as the mechanisms of mining procedures, clearing, and fusion, data storage, and privacy issues. They have considered e-health application case-study in the fog environment with data management and provided a conceptual architecture for it. Nonetheless, the following shortages exist in the reviewed article:

- Narrow coverage of related subjects.
- Poor categorization of subjects.

Moysiadis et al.[19] presented a comparison between different distributed data storage systems employed in fog and edge computing systems and the importance of storage in such systems. Mansouri et al. [20] presented an overview of different data replication methods in cloud computing. In these articles, the key points of metaheuristic algorithms have been examined; furthermore, a tabular representation of the features of these algorithms is provided.

They have also concluded that there is no comprehensive meta-algorithm for replication in a cloud system, being able to meet all needs of the system. The following shortages exist in the reviewed article:

- Future directions are poorly covered and incomplete.
- Narrow coverage of related subjects.

Mazumdar et al. [21] provided an overview of big data placement and data storage methodologies in cloud networks. They have highlighted the relation between storage and data placement to improve the knowledge about the role of these two in the non-functional properties of big data management. However, the following shortages exist in the reviewed article:

- Narrow coverage of related subjects.
- Poor categorization of subjects.

Tabet et al. [3] presented a taxonomy for the replication strategy in cloud systems. They have classified this strategy based on five factors. Although their classification properly considers different aspects of replication strategies, the following shortages exist in their work:

- Lack of precise details of presented subjects.
- Insufficient technical explanations.

In [22], They used the SLR method to examine fog data management to understand the various topics and key areas in the field that have just been presented. Their purpose was to classify and analyze research related to the field of May data management, which was published from 2014 to 2019. Field-based classification is also provided for fog data management. Their job classification includes data processing, data storage, and data security. Nonetheless, the following shortages exist in the reviewed article:

- Narrow coverage of related subjects.

Rani et al. [5] have presented a detailed review of those research articles which have implemented storage techniques at the fog layer. Also, they review the benefits of Fog computing over cloud computing and how secure storage techniques are implemented at the Fog layer. They argue that cloud-level storage helps maintain user data security by reducing latency and increasing user control, which did not exist in cloud-based storage. However, the following shortages exist in the reviewed article:

- Future directions are not covered appropriately.

Islam et al. [23] presented a survey on context-aware scheduling in Fog computing. also, They have classified this strategy based on factors such as context-aware parameters, performance metrics, case studies, and evaluation tools with their advantages and limitations is given. However, the following shortages exist in the reviewed article:

- Deals only with context -aware scheduling in fog computing.

Heidari et al. [24] they used the SLR method to cloud service discovery mechanisms in three major classes: centralized, decentralized, and hybrid. However, the following shortages exist in the reviewed article:

- Narrow coverage of related subjects.

Briefly, there are several unanswered questions in the data replication strategies literature within the fog and edge contexts. Accordingly, we have provided our survey since we believe that all other works, to the best of our knowledge, either have narrowed their study only on cloud systems or have not merely concentrated on the problem of data replication and covered a broad range of issues.

According to the mentioned survey articles in the field of data replica placement in fog and edge computing, a side-by-side comparison in terms of advantages, disadvantages, open issues, future works, published year, paper selection, comparison, and the type of review of reviewed papers has been summarized into Table 1.

Unlike other surveys that we reviewed, this paper discusses data replication strategies in fog networks and provides a classification on all papers with a presented strategy in data replica placement in fog.

The previous papers suffer from some weaknesses as follows:

- Many papers [2, 16, 18–21] did not coverage of related subjects of the resource data replica placement in fog computing.
- Some papers[2, 16, 21] poor categorization of data replica placement approaches in fog computing.
- The organization of the existing works does not have a systematic format to select papers data replica placement in fog computing.
- Some Papers [19, 20] did not provide the future direction of resource data replica placement approaches in fog computing.
- Many Papers [3, 16–18] did not provide technical explanations of data replica placement approaches in fog computing.

The aforementioned reasons led us to present a review paper on the mechanisms of data replica placement in the fog computing environment in order to cover all these shortcomings.

4 Research methodology

This section elaborates on the procedure of selecting related papers in the literature of data replica placement in the fog and edge domain. This paper has been accomplished benefiting from the common Systematic Literature Review (SLR) approach.

4.1 Question formalization

According to the SLR approach, we outlined some Technical Questions (TQ) regarding the scope of our study in fog and edge computing.

- **TQ1:** What classification is applied in data replica placement methods in the fog domain?
- **TQ2:** What performance metrics are usually measured in data replica placement approaches in the fog domain?

Table 1 A side-by-side of review papers in the data management strategies in fog computing

Refs.	Type of review	Paper selection	Classification	Comparison	Open issues	Other advantages	Other disadvantages
[18]	Survey	No	Yes	Yes	Yes	Reviewing past and state-of-the-art cloud data replication	Lack of describing methods of paper selection Poor categorization of mechanisms
[3]	Survey	No	Yes	No	Yes	Presenting a well-defined classification for cloud replication Providing a solid understanding of the replication concept	Lack of presenting the detailed result as they claim
[19]	Survey	No	No	Yes	Yes	Presenting a well-defined introduction about distributed data storage mechanisms in the cloud and fog environment	Lack of providing any classification of methods or procedure of selecting papers Lack of presenting a comprehensive conclusion
[17]	Survey	No	Yes	Yes	No	Presenting a general and comprehensive classification for fog data analysis	Lack of suggesting any open issue for future research Lack of mentioning the procedure of choosing papers
[2]	Review	No	No	No	Yes	Using a case study for a better understanding of data management in the fog domain	Lack of well-organized classification or comparison between papers in this field
[21]	Survey	Yes	Yes	Yes	Yes	Providing a holistic overview of the state-of-the-art methods related to both storing and placement of big data in the Cloud ecosystem	Lack of mentioning and classifying different approaches of data placement
[16]	Review	No	No	Yes	Yes	Evaluating data placement algorithms papers in fog environment via simulation	Lack of describing how to classify papers and providing any classification of papers
[20]	Survey	No	Yes	Yes	Yes	Classifying data replication strategies in the cloud domain through meta-heuristic approaches	Lack of evaluating papers based on their claims Lack of presenting how to find related papers to review
[22]	SLR	Yes	Yes	Yes	Yes	Providing an inclusive comprehension of the fog data management Presenting a thorough description of open issues of this field and related challenges	Lack of any evaluation of classified papers An overfit focus on data management
[5]	SLR	yes	yes	yes	yes	Presenting motivation behind the research and procedure followed Providing the steps followed for review	Many research articles were not considered due to the absence of a search string in the article's abstract and title
[23]	Survey	Yes	Yes	Yes	Yes	Presenting a thorough description of open issues of this field and related challenges Presenting a comprehensive classification on context-aware scheduling in fog computing	Deals only with context -aware scheduling in fog computing
[24]	SLR	No	Yes	Yes	Yes	Presenting a description of open issues of this field and related challenges This paper is utilized to detect, evaluate and combine findings from related investigations	Narrow coverage of related subjects

- **TQ3:** What case studies are considered in data replica placement approaches in the fog domain?
- **TQ4:** What evaluation tools are applied for assessing the data replica placement approaches in the fog domain?
- **TQ5:** What techniques are applied in data replica placement approaches in the fog domain?
- **TQ6:** What are the future research directions and open issues in data replica placement methods in the fog domain?

Sections 6 and 7 are dedicated to addressing the above questions.

4.2 Data exploring and article selection

To select the most relevant papers in the field of data replica placement in fog computing, we have used the following academic article databases, as shown in Table 2

We have searched the following combination of keywords and their synonyms to find all papers in the area of data replica placement in fog systems.

- (“Data Replica” OR “Replica Placement” OR “Data Replication” OR “Replica management” OR “Data placement”) AND (“Fog”) OR (“Fog computing”) OR (“Edge computing”).

Figure 4 is a visualization of keywords used in these articles. This chart provides a better intuition about what the most important keywords of this topic are.

Figure 5 shows a framework that describes how we have selected and evaluated articles for the proposed taxonomy. This was accomplished in October 2021, by setting a constraint on the time range from 2017 to 2021. The search result was the total number of 465 articles. By studying the main sections of these articles, we found that 257 articles are irrelevant to our topic, and those were discarded just at the beginning. In the next step, by studying the main part of the remaining articles, we found that 59 articles did not have the desired quality, and 5 articles were similar, and 9 articles were reviewed; thus, 73 articles were excluded totally.

Table 2 Academic databases

Database	URL	Works
Springer	http://link.springer.com	[1, 20, 25–31]
Science Direct	http://www.sciencedirect.com	[17, 32–39]
IEEE	http://ieeexplore.ieee.org	[15, 40–46]
ACM	http://www.acm.org	[47, 48]
Google Scholar	http://Scholar.google.com	[49–51]

Also, 105 papers were not particularly discussed data replica placement and discarded. Finally, the remaining 30 articles that were related to data replica placement have been included in the review.

In this study, due to lack of resources, we did not review any non-English research and we used “since 2017” as our filter.

Figure 6 shows research variation per year demonstrated by the publisher. As can be observed, Elsevier, which indexes in the Science Direct database, has the most number of papers on this subject.

4.3 Data replica placement approaches in fog computing

Data replica placement approaches are distinguished based on some criteria, design configuration, and deployment strategies. In this paper, we have focused on the four following aspects of data replica placement methods.

Almost all replica placement approaches in the fog and edge domain are investigated, among which some applied a framework or graph as a solution to place data replicas on nodes, and others utilized heuristic and meta-heuristic approaches. We categorized them into four main categories.

Figure 7 demonstrates the taxonomy of data replica placement approaches based on a framework, graph, heuristic, and meta-heuristic properties.

However, there are also other perspectives for categorizing data replica placement approaches such as placement controller, placement dynamicity, and placement events, as shown in Fig. 8.

Replica placement controller is the first perspective. Indeed, all prior studies have controlled their data replica placement mechanism either by a central controller or in a distributed manner by distributing the controlling tasks throughout some or all of networking and/or edge devices; also, some of them conducted a hybrid approach that controls a proportion of tasks in the central controller and the rest in networking and/or edge devices. The two centralized and distributed controlling strategies in the placement controller category have their advantages and disadvantages. A centralized control requires global knowledge about the whole system information, applications on demands, and network infrastructure for making decisions. However, it can find an optimum strategy for replica placement in contrast. However, another centralized controller’s downside is its scalability challenge due to the computational complexity. On the contrary, the decentralized approach is more flexible and scalable, more complex for design, yet needs less computational complexity. It can present an acceptable solution without any global knowledge about the whole network infrastructure and

Fig. 4 Keywords frequency in reviewed papers

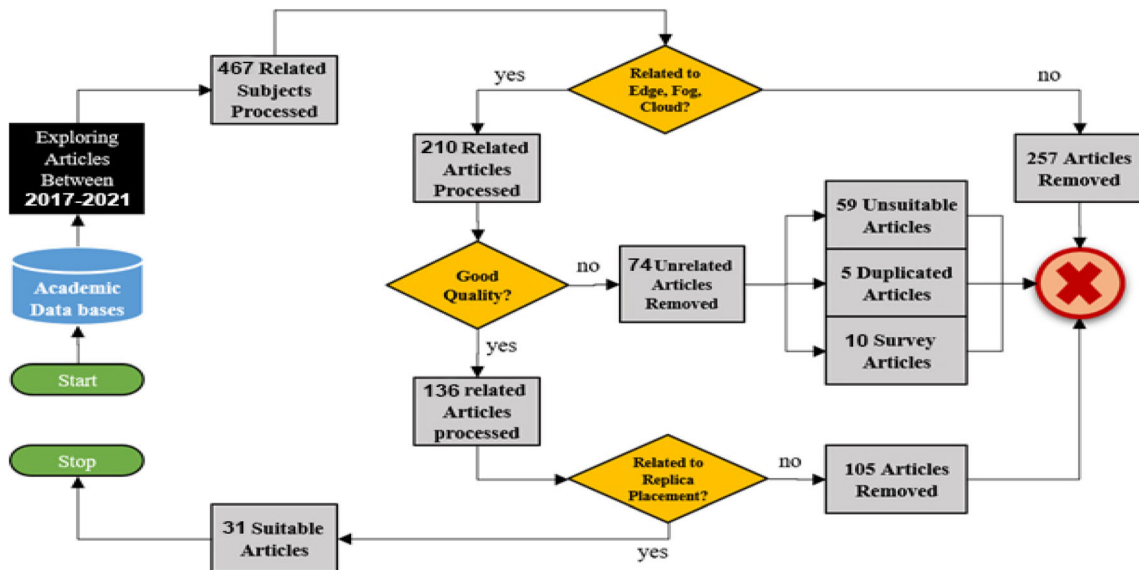


Fig. 5 Paper selection process

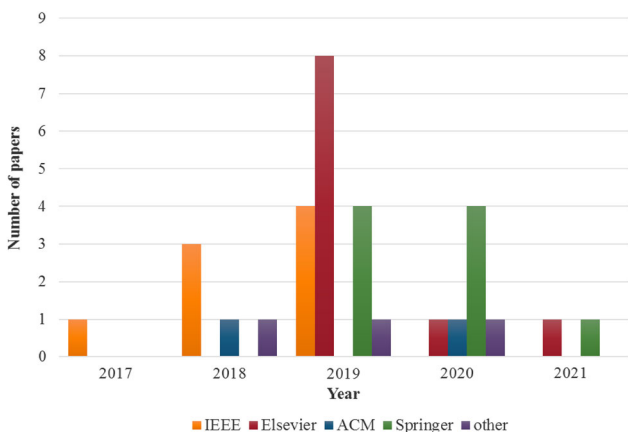


Fig. 6 Number of papers classifies with publisher and year of publication

application demands, but it is often unable to provide a globally optimum solution.

Dynamicity is the next category that determines whether the number of replicas is determined statically or dynamically. In the static approach, as the name reveals, the

number of replicas is fixed during the runtime. In contrast, in the dynamic placement strategy, the number of replicas is flexible and adjusted in different stages over time. In a large-scale system with a lot of IoT devices, fog nodes, and applications, nodes leave and join the system due to many reasons such as nodes failure because of the instability of networks, adding new devices, or the user decides to send or receive data at any time.

To deal with such behavior, we need to define a strategy to dynamically change the location and number of replicas for each data. So, in order to guarantee an acceptable level of quality of service, we need to realize when adaptation is required and deploy a mechanism to dynamically add or remove some resources such as data.

Events category discusses enhancing the data availability through triggering some events in the system. In the placement events category, the reactive replication makes data more available by dynamically replacing the non-existence and failure of data replicas. In contrast, the proactive approach offers a mechanism that calculates the mean of available data replicas for a more extended period, e.g.,

Fig. 7 Proposed taxonomy of data replica placement approaches in fog computing

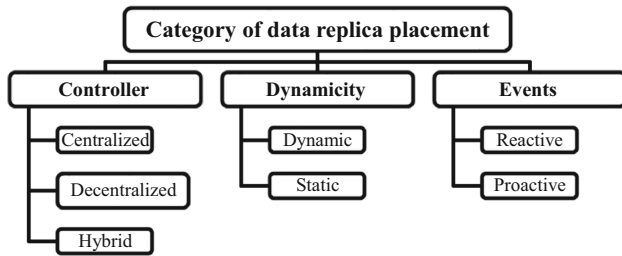
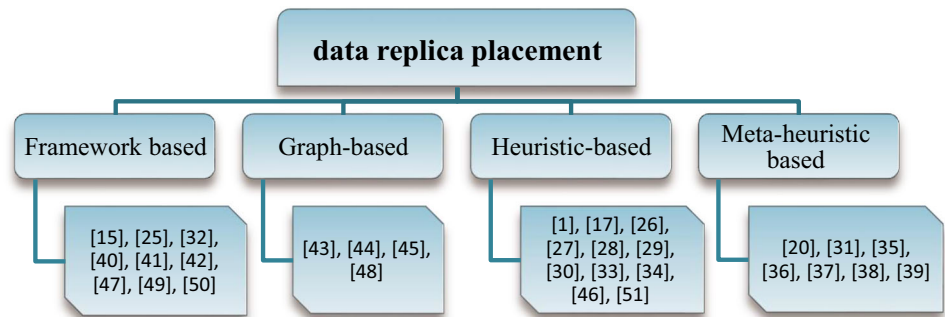


Fig. 8 Proposed category of data replica placement approaches in fog computing

providing three available replicas on average for one month. Some papers have not devised any solution for replicating failed data and have not even mentioned it.

4.4 Overview on the data replica placement approaches

Data replica placement is one of the key issues in data management in fog computing. This section discusses different data replica placement methods in fog environments. We have categorized all data replication approaches into four main groups as follows: framework-based, graph-based, heuristic-based, and meta-heuristic-based methods.

4.4.1 Framework-based data replica placement mechanism

The framework-based solutions create and use frameworks for solving the data replica placement problem.

Vales et al. [32] proposed a hybrid system of controlling the placement of data replicas. They proposed an approach that uses the storage of fog nodes and IoT devices for storing and serving data to the consumer. To do so, they applied an adaptive distance metric that manages the edge service to replicate data for node clustering. In evaluating this distance metric, three important parameters are considered: Spatio-temporal data popularity, distance from consumer to data, and self-ruled battery-powered node. Their results confirm that this hybrid system provides end-users more desirable data access due to the reduction of file

transfer time; also, the backhaul links become less congested because of the replication of data in the edge.

Naas et al. [15] presented a strategy for data replica placement that considers heterogeneity and location of fog nodes. Their approach focuses on latency for storing and restoring data in fog systems. They solve this as a Generalized assignment problem (GAP) and reached two solutions. The first is an integer programming solution, and the second is a heuristic solution based on dividing different geographical zones. Monga et al. [40] proposed Elfstore, a federated data storage service, that uses both p2p architecture and HDFS like distributed storage systems. They applied a federated method for indexing blocks of data with help of a bloom filter. They used reliable fog nodes for managing and monitoring edge resources.

Mayer et al. [41] used existing distributed data stores (DDSs) and then designed and managed them so as to be used properly in current fog systems. They proposed a fog-aware replication strategy and evaluated their work using the yahoo cloud serving benchmark. Hasenburger et al. [49] developed a middleware framework for programmers that facilitate data movement and data replica placement across the fog nodes. They presented a replication service which is an abstraction for programming and named it Fbase. In [47], the authors determined a set of requirements that a replication service in a data-intensive fog system would have needed. They also used an abstraction for programmers to handle how to distribute data.

In [42], a general introduction to data streaming in fog infrastructure has been proposed. They have presented a framework to use the processing power of fog nodes for data streaming. Gupta et al. [50] developed a data management system at the network edge, designed for geodistributed and heterogeneous fog networks. Their system accomplishes data replica placement both in fog nodes and between fog and cloud layers. Moreover, their data replica placement approach considers geographic distribution. In [25], They considered the use of fog-based computing resources for data processing purposes in the Internet of Things. To this end, they introduced a practical scenario in the field of industry and provided a framework for flow

processing in fog. Breitbach et al. [42] proposed a data management system and a scheduler for both data and tasks that combines these two and adapts data replica placement at runtime.

We provide a comparison between these framework-based solutions in Tables 3 and 4.

According to Fig. 9, in framework-based placement approaches, the decentralized methods, with 56%, are more popular than centralized methods, with 33%, in terms of dynamicity. Likewise, in the controller category, the share of applying dynamic methods is 1.27 times more than that for static ones. Finally, the figure for reactive methods is double as much as proactive ones in the event category.

4.4.2 Graph-based data replica placement mechanisms

A graph-based algorithm creates a graph of fog nodes and divides the domain of the problem into smaller ones to tackle the problem.

Nass et al. [48] presented a heuristic divide-and-conquer method for placing data across the fog environment. They follow the strategy of breaking the problem space down into smaller sections by means of graph modeling. In this case, they divide the fog environment into some sectors. They claim that solving this problem with this heuristic method reduces the problem-solving time by almost 450 times.

Confais et al. [43] used a method based on a physical topology, called tree-based, instead of the common Distributed Hash Table (DHT). In their proposed method, servers seek the location of an object by repeatedly sending requests to their roots in the tree. New location records, which have been logged to reduce network traffic during the process of request for an object in placement, also improve access time.

Lera et al. [44] proposed a balanced distribution of the distance between the fog device and the data sources in order to store the generated data by the sensors as close as possible to IoT devices. They consider the Centrality Complex Weighted Networks (CCWNs) criterion for identifying devices that are even close to the sensors. They assumed that selecting those devices to store data in fog would reduce network utilization.

Confais et al. [45] applied the Domain Name System (DNS) protocol, which intrinsically uses a tree-based method to place data. In this protocol, servers seek the best location for storing an object by repeatedly requesting their root in the tree. The tree is created and modified by employing a version of the Dijkstra algorithm over the physical topology.

We have presented a side-by-side comparison of different metrics of graph-based scientific papers in Tables 5 and 6.

According to Fig. 10, it is observable that, in graph-based placement mechanisms, only decentralized approaches have been applied in the dynamicity category. Likewise, applying the dynamic approaches is three times more than static ones in the controller category, and in the event category, the share for reactive and proactive approaches is relatively equal.

4.4.3 Heuristic-based data replica placement mechanisms

Heuristic algorithms are designed to solve problems, which conventional algorithms are unable to solve, efficiently and/or in an acceptable time. These algorithms fairly sacrifice optimality, precision, or completeness for better execution time. Heuristic algorithms are often applied for solving NP-complete problems, including data replica placement. In this section, we have, firstly, investigated prior works in the area of data replica placement applying heuristic-based approaches.

Naas et al. [26] considered heterogeneity in data replica placement problems. Because fog infrastructures vary in size, they have introduced two heuristic methods, iFogStoreS, and iFogStoreP. Both methods for managing data placement in fog use the consistency parameter to reduce data replica sync delays. iFogStoreS provides more accurate performance for fewer nodes (up to tens of nodes) but requires more processing, and iFogStoreP is dedicated to a large number of nodes (up to thousands of nodes) that perform poorly compared to the previous strategy, but calculations are much lighter. Their experiments showed that when using the proposed strategies, service delays could be reduced by 30% in the case of small fog infrastructure and up to 13% in case of large-scale fog infrastructure compared to the iFogStore base method.

Karatas et al. [17] proposed an IoT-based hierarchical and geographically distributed architecture in cloud and fog environments. They also developed a technique for placing data in cloud and fog stations. Authors categorized data into different types according to their functionality in different applications. Finally, they attempted to find an optimal algorithm by modeling the data replica placement as an optimization problem. Accordingly, they proposed two heuristic algorithms and compared their work results with other heuristic and meta-heuristic optimization algorithms by conducting some simulation efforts.

Guerrero et al. [51] introduced decrements in latency and increment in availability as one of the chief reasons for emerging fog and cloud platforms. Unlike other data replica placement methods, which store only a single replica of data, their proposed system maintains multiple replicas to reduce latency and increase availability. Their proposed system is modeled through complex weighted networks and topological features, such as Centrality

Table 3 A side-by-side comparison of framework-based data replica placement mechanisms

Refs.	Utilized technique	Evaluation tools	Placement				Performance Metrics								
			Dynamicty	Controller	Event	Network	Average latency	Availability	Scalability	Workload	Number of replicas	Number of users			
													Latency	Bandwidth	
[15]	Generalized Assignment Problem (GAP)	iFogSim	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*	*	*	*
CPLEX MILP															
[25]	Framework-based	Theory	Decentralized	Static	Proactive	*	*	*	*	*	*	*	*	*	*
[32]	Framework-based	Testbed	Hybrid	Dynamic	Reactive	*	*	*	*	*	*	*	*	*	*
[40]	Place the first replica locally	Testbed	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*	*	*	*
[41]	Context-Sensitive Differential Consistency	FogStore	Decentralized	Dynamic	Reactive	*	*	*	*	*	*	*	*	*	*
[42]	Framework-based	Testbed	Decentralized	Dynamic	Reactive	*	*	*	*	*	*	*	*	*	*
[47]	Framework-based	Design	Decentralized	Static	None	*	*	*	*	*	*	*	*	*	*
[49]	Framework-based	JAVA	Centralized	Static	None	*	*	*	*	*	*	*	*	*	*
[50]	Framework-based	Apache Cassandra	Decentralized	Static	Proactive	*	*	*	*	*	*	*	*	*	*

Table 4 A side-by-side comparison of main idea, advantage, and disadvantage of framework-based solutions

Refs.	Main idea	Advantages	Disadvantages
[15]	Using POPs (Point of Presence): LPOPs and RPOPs, local and regional POPs	Testing in a large-scale system	Lack of mentioning replication mechanisms
[25]	Presenting a VISIP ecosystem with a runtime module and computational resources	Providing a framework for data steaming in the fog domain to reduce overhead and latency and use of bandwidth in the network	Lack of san implemented version for validation of the framework
[32]	Controlling the number of data replicas based on popularity	Designing a hybrid system that uses edge devices as storage	Lack of discussing the number of users and tested in a small environment
[40]	Using Bloom's hierarchical filters on fixed metadata features for fast possible scale search, it uses a federation index to search for 2-hop blocks	Performing automatic block repetition in edge failures to maintain the required reliability	It is failed in large scale and concurrent tests
[41]	Proposing a solution for storing distributed data in fog computing	Discussing a solution for distributed data storage in fog	Inconsistency in distributed data replicas
[42]	Context-based data and task placement in fog computing systems	Discussing a multi-level scheduler and runtime data replica placement	Lack of assessing on a reasonable scale and did not discuss network topology and user preferences
[47]	Controlling the data flow and replication across geographically distributed fog nodes	Identifying a set of requirements for a replication service	Lack of any proof of concept
[49]	Using Kotlin to program abstraction for data-intensive fog systems	An abstraction framework for using as middleware in fog systems	Lack of provisioning any data store for storing and retrieving data
[50]	A hash function for locality indexing and a load balancing region composed from a hash prefix	A data management system that replicates data for reducing latency and tolerated failures	Lack of conducting a quantitative evaluation

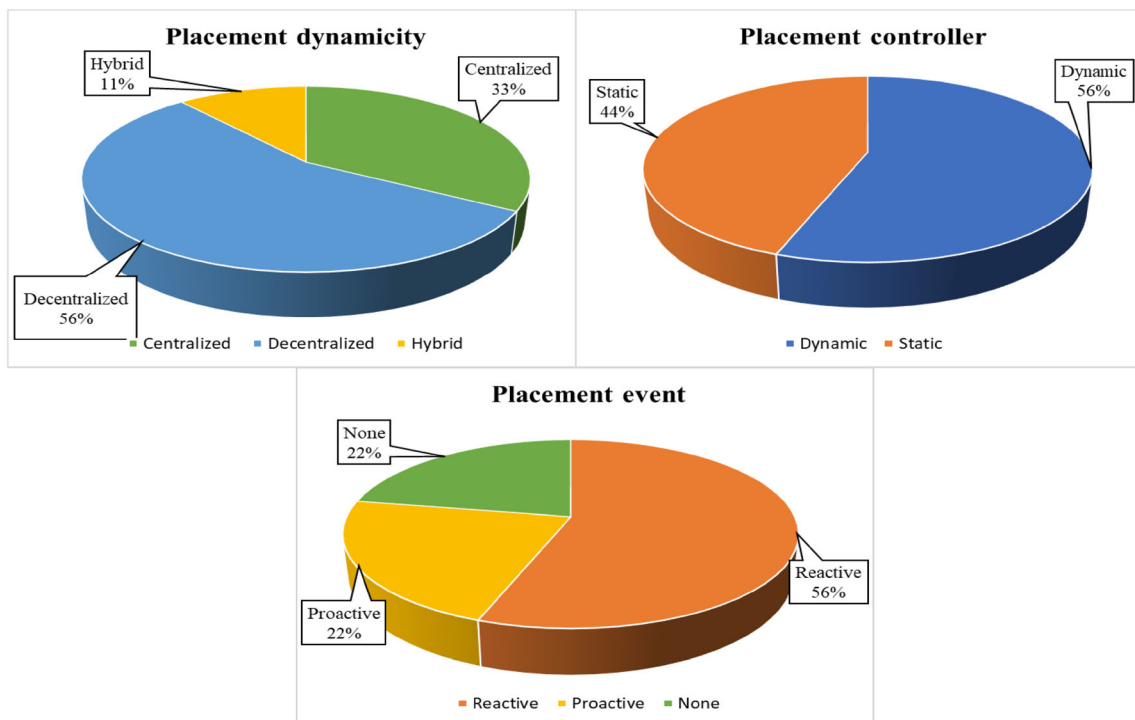
**Fig. 9** Percentage of different framework-based data replica placement mechanisms

Table 5 A side-by-side comparison of different aspects of graph-based technical studies

Refs.	Utilized technique	Evaluation tools	Placement				Performance metrics					
			Dynamicty	Controller	Event	Network	Average latency	Availability	Scalability	Workload	Number of replicas	Number of users
[43]	Graph-based	Testbed	Decentralized	Dynamic	Proactive	*	*	*	*	*	*	*
[44]	Centrality indices	YAFS	Decentralized	Dynamic	None	*	*	*	*	*	*	*
[45]	Tree-based	Testbed	Decentralized	Static	None	*	*	*	*	*	*	*
[48]	Graph-based	iFogSim	Decentralized	Dynamic	Reactive	*	*	*	*	*	*	*

indicators. Thus, graph partitioning algorithms are employed to select the best fog devices for storing data.

Huang et al.[27] claimed a single replica fails to meet the needs of reducing network latency according to differences in the topology of data users. Thus, they proposed the multi replica data replica placement model, called iFogStorM, for fog environment, benefitting from which they successfully solved the problem of finding the optimal number of data replicas and the optimization problem of placement of data on fog nodes. Also, they proposed the greedy MultiCopyStorage algorithm to reduce the overall latency of the data replica placement strategy.

Li et al. [28] presented a strategy for creating data replicas and finding the optimal data location in the cloud and fog environments. Their proposed data replica creation algorithm is dynamic and based on access heat (DRC-AH), and the data replica selection algorithm is based on node service capability (DRS-NSC). The DRC-AH utilizes data blocks for partial replication and Gray Markov chain for dynamic adjustment of the number of replicas. After receiving the user request from the client, DRS-NSC selects the best node to respond to the user.

Guo et al. [1] considered features such as dynamicity, speed, recovery, and consistency in the data replica placement problem. They proposed a fast recovery method in data replica placement to cope with the data inconsistency problem caused by frequent updates of different versions. Dynamic Replica Creation Based on Domain Structure (DRC-DS) module identifies the number and location of replicas based on regional structure, data access frequency, and average response time. This method leads to a reduction or balance on the edge server load, being considered in their article.

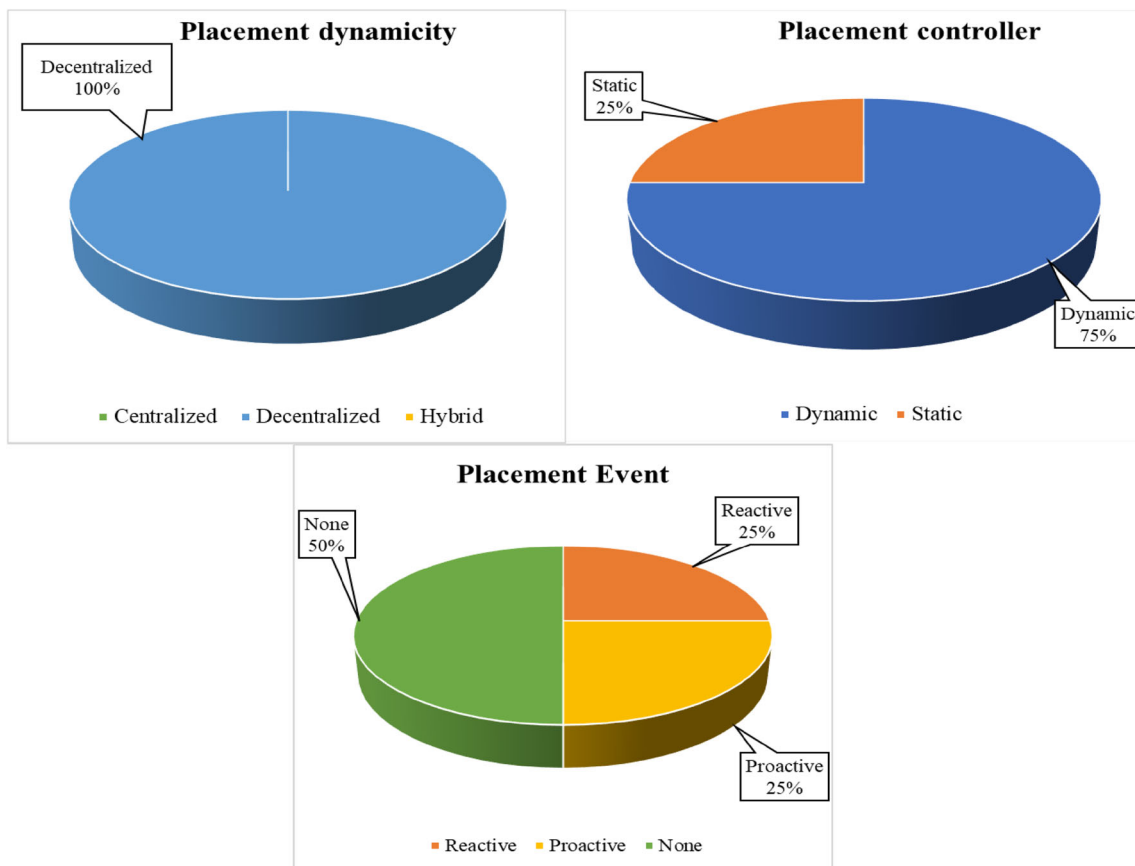
Saranya et al.[29] proposed a strategy to randomly replicate data on Mobile Edge Computer and then evaluated their strategy with a varying number of mobile devices, resulting in improved latency and bandwidth. Their experimental results confirm the superiority of applying a random algorithm for replica placement instead of a simple algorithm to reduce the amount of bandwidth utilized by data in the network.

Li et al.[33] attempted to shrink computational latencies and response time by proposing an optimal replica placement and data blocks to enhance the user experience in the edge computing environment. In their method, the popularity of data blocks, data storage capacity, and replacement ratios of an edge server, which stores data blocks, are considered, and data blocks are ranked based on these parameters for replica placement. Furthermore, the cost of placing data replicas for each block is taken into account.

Qureshi et al.[30] developed the Wireless IoT Edge-enabled Block Replica Strategy (WIEBRS) platform, which stores data replicas on edge nodes based on in-place,

Table 6 A side-by-side comparison of the main idea, advantage, and disadvantage of graph-based solutions

Refs.	Main Idea	Advantages	Disadvantages
[43]	A tree-based solution for finding the location of data replicas in fog systems	Replacing DHT by a tree-based approach to physical topology and using Dijkstra's algorithm to find data replica location	Assuming IoT devices are fixed, while the dynamicity in the topology in such systems is imminent
[44]	Generating a graph from nodes and using graph algorithms	Optimizing the network usage	Just focusing on locating data, and lack of producing any replica from the data
[45]	Tree-based algorithm using Dijkstra	Using simple methods like Dijkstra and evaluating that	Lack of considering dynamicity of network topology
[48]	A heuristic method based on partitioning mechanisms	Scalable in large scale network and divide problem into small pieces	Despite the improvements in the proposed method, latency and accessibility time is high

**Fig. 10** Percentage of different graph-based data replica placement mechanisms

partition-based, and multi-homing blocks. This leads to shrinking the latency of accessing datasets for aggregate MapReduce and increasing the performance of the job in the smart grid. Big data structures and large volumes of data in many advanced systems are managed through the MapReduce model.

Li et al.[34] proposed an energy-aware clustering strategy to reduce the system's energy consumption and achieve energy efficiency by turning data nodes off and on

according to the system load status. They solved the data placement problem using multi-objective optimization, which jointly minimizes response time and guarantees the network load balance through the genetic algorithm. They applied this strategy to some delay- and privacy-sensitive applications, such as smart farming, social media, transportation, and the military.

Aral and Ovatman [46] proposed a decentralized method to monitor data transmitted from edge nodes and apply

creation/replacement/removal functions for placing them dynamically. Their effort is based on the demand from users, the geographical proximity of data users, localization of storage, and storage capacity costs to minimize access delays.

Then, a comparison between these works in terms of their performance metrics, utilization techniques, evaluation tools, and advantages and disadvantages of each applied method is presented in Tables 7 and 8.

According to Fig. 11, it is observable that, unlike the two aforementioned framework-based and graph-based mechanisms, in heuristic-based placement mechanisms, the share of centralized approaches, with 60%, is approximately two times more than that for decentralized ones in the dynamicity category. Likewise, in the controller category, both dynamic and static approaches hold an equal share, and in the event category, the percentages of applying reactive and proactive approaches are 40% and 30%, respectively.

4.4.4 Meta-heuristic based data replica placement mechanisms

Meta-heuristic algorithms combine standard heuristic methods with higher-level frameworks to select proper heuristics for finding the problem solutions in the search space. This technique is one of the common techniques for finding almost every solution because they are broadly applicable, can combine with traditional techniques, implement easily, and can solve a problem in an acceptable time. Meta-heuristic techniques have been widely applied in scientific papers (i.e., applications, algorithms, analysis, and comparisons) in this literature due to their flexibility, and simplicity [20].

Li et al. [35] divided the fog space into clusters and mainly concentrated on reducing the energy cost of each cluster server. Then, they have presented a scalable strategy for clusters based on dynamic placements and energy cost optimization, called energy-aware. Their experiments confirm that their proposed algorithm effectively reduces the average operating time, network bandwidth utilization, and storage space occupation.

Shao et al. [36] concentrated on the collaborative edge and cloud environment. They set the data replica placement model in a system called collaborative-aware, in which they have presented the data replica placement problem as a binary integer model. By doing so, in addition to guaranteeing data reliability, they minimized the cost of data access and designed a framework for cooperative processing.

Chen et al. [31] firstly, developed the Data-Intensive Service Edge Genetic Algorithm (DSEGA) and formed the edge server model based on the graph theory algorithm.

Then, they employed five different algorithms such as Genetic Algorithm (GA), Simulated Annealing (SA), Ant Colony Algorithm (ACO), Optimized Ant Colony Algorithm (ACO v), and Hill Climbing to obtain an optimal placement scheme for the data replicas, respectively. Comparing these algorithms shows that their proposed algorithm outperforms other algorithms in terms of response time and latency.

Li et al. [37] proposed Dynamic Replica Creation Algorithm (DRC-GM) and the Fast Non-dominated Sorting Genetic replica placement (RP-FNSG) algorithm in the edge computing environment. The DRC-GM algorithm considers the data in smaller blocks as granular and comprehensive, and the system environment is deemed to be real-time. The number of replicas is adjusted dynamically and according to the user's demands. Experimental results show that DRC-GM and RP-FNSG algorithms in the edge computing environment improve the system performance significantly.

Shao et al. [38] investigated the number and placement of data replicas to optimize workload in edge and fog computing systems and proposed a deadline-driven scheduling strategy. They utilized the data blocks and fog nodes to obtain the number of replicas and their locations. They have also designed a system monitor and a number of data security tools for cloud-edge collaboration.

Li et al. [39] proposed the Fast Non-dominated Sorting Genetic (FNSG) algorithm and the retrieval of failed data node data in a cloud-edge system. Their research paper also presents a delayed synchronization schema for data replica placement and a load balancer based on placement recovery strategies. Finally, they proved the effectiveness of their method through their experimental evaluations.

Accordingly, the side-by-side comparison of meta-heuristic-based approaches in data replica placement has been presented in, Tables 9 and 10.

According to Fig. 12, it is observable that, in meta-heuristic-based placement mechanisms, the centralized approaches, with 86%, hold the major contribution of the dynamicity category compared to the decentralized solutions. Similarly, it is noteworthy that the margin between dynamic and static approaches in the controller category is exactly the same for the dynamicity category, with the dynamic solutions having 86% of the contribution. In the event category, the popularity of applying reactive approaches is 2.3 times more than proactive ones.

5 Summary and discussion

As stated by previous works of literature, we identified some solutions to solve the data replica placement problem concerning utilized techniques. We discuss framework-

Table 7 A side-by-side comparison of different aspects of heuristic-based data replica placement mechanisms

Refs.	Utilized technique	Evaluation Tools	Placement			Performance metrics							
			Dynamism	Controller	Event	Network Latency	Bandwidth	Average latency	Availability	Scalability	Workload	Number of replicas	Number of users
[1]	Knapsack algorithm	Testbed	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*	*
[17]	Heuristic Linear optimization model	Testbed	Centralized	Static	None	*	*	*	*	*	*	*	*
[26]	consistency protocol: Floyd-Warshall algorithm for iFogStoreS and Cplex MILP for iFogStoreP	iFogSim	Decentralized	Dynamic	Proactive	*	*	*	*	*	*	*	*
[27]	CPLEX	iFogSim	Centralized	Static	Proactive	*	*	*	*	*	*	*	*
[28]	Dynamic Replica Creation based on Access Heat (DRC-AH) Replica Selection based on Node Service Capability (DRS-NSC) Grey Markov chain	Testbed	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*	*
[29]	Heuristic-based	NS-2	Centralized	Static	Proactive	*	*	*	*	*	*	*	*
[30]	MapReduce	Hadoop	Decentralized	Dynamic	Proactive	*	*	*	*	*	*	*	*
[33]	Heuristic-based	JAVA	Centralized	Static	None	*	*	*	*	*	*	*	*
[34]	Genetic algorithm	Hadoop	Hybrid	Dynamic	Reactive	*	*	*	*	*	*	*	*
[46]	Mixed-integer programming and multiple heuristic solutions	Amazon S3	Decentralized	Dynamic	Reactive	*	*	*	*	*	*	*	*
[51]	Centrality indices	YAFS	Decentralized	Static	None	*	*	*	*	*	*	*	*

Table 8 A side-by-side comparison of main idea, advantage, and disadvantage of heuristic-based papers

Refs.	Main Idea	Advantages	Disadvantages
[1]	Fast recovery of replica for load balancing	Preserving replica consistency	Lack of considering scalability for bigger networks
[17]	Distributed data service based on fog network for IoT devices	Presenting a hierarchical architecture within fog and cloud and using metaheuristic methods	Lack of comparing results evaluation with meta-heuristic algorithms like GA and PSO
[26]	Data Replication with Consistency Management in Fog structure	Separation of small fog infrastructure algorithms from large fog infrastructure	the replicas synchronization process may add a latency overhead to the system
[27]	MultiCopyStorage heuristic algorithm	Presenting a good solution for storing data in real-time	Lack of considering data integration and data confidentiality when replicating and storing data
[28]	Select scalable replicas based on node service capability to improve data accessibility in the fog computing system	Proposing a solution to reduce bandwidth and accessibility latency in fog computing	Reviewing just a selected dataset and lack of investigating real datasets
[29]	Random placement of replica	Testing random algorithm one for all time	Applying a random algorithm which always exists better algorithms than random
[30]	Aggregate MapReduce	Using HDFS as a file system	Lack of using intermedia replica placement
[33]	A collaborative model for data placement and task scheduling	Applying a combination of the task and data scheduling creatively	Lack of using servers in the edge which is sub-optimal
[34]	Fast non-dominated sorting genetic algorithm	Applying their method to different scenarios	Lack of considering cost and energy consumption
[46]	D-ReP (Distributed—Replica Placement) algorithm and FLP (Facility Location Problem)	Presenting a cost-efficient and latency-aware algorithm	Requiring to propose an interface for programming
[51]	Complex weighted network and graph partitioning algorithm	Improving the network latency by adding some replica	Increasing the number of transmitted packages

based, Graph-based, Heuristic-based, and Meta-heuristic based solutions below. The framework-based solutions [15, 25, 32, 40–42, 47, 49, 50] use frameworks for solving the data replica placement problem. The graph-based solutions [43–45, 48] often utilize mathematical techniques such as Mixed-Integer Linear Programming (MILP), Integer Programming (IP), and Linear Optimization Model, while the heuristic [1, 17, 26–30, 33, 34, 46, 51] and meta-heuristic-based [31, 35–39] solutions apply heuristic/meta-heuristic algorithms such as MultiCopyStorage algorithm, artificial bee colony, ant colony, genetic algorithm, and Hill Climbing optimization. Most studies have examined the performance metrics such as the network latency, network bandwidth, and availability metrics due to the exchanged data during the data replica placement. However, the number of users, the number of replicas, and workload metrics for data replica placement have not been considered much. Moreover, it has been figured out that most recent research studies apply testbed environments according to their assessment tools. For example, those who applied small-scale testbed environments [1, 17, 28, 32, 37, 40, 42, 43, 45] and medium-scale

evaluation studies applied iFogSim toolkit [15, 26, 27, 48], or YAFS [44, 51], and other studies, which considered large-scale environments, used Hadoop data file system [30, 34, 35, 39] or apache Casandra [50].

6 Discussion

This section discusses and examines the existing papers in the literature on replica and data placement in edge and fog computing systems. This analytical examination is based on our Technical Question in Sect. 4.

- **TQ1:** What taxonomy is applied in data replica placement methods in the fog domain?

A statistical comparison between data replica placement papers in fog computing is presented based on the taxonomy proposed in Sect. 5. Figure 13 demonstrates the classification of replica papers in the fog domain according to their placement methods. Heuristic-based and meta-heuristic-based algorithms are contributed in more than half of the papers. The share of Framework-based methods

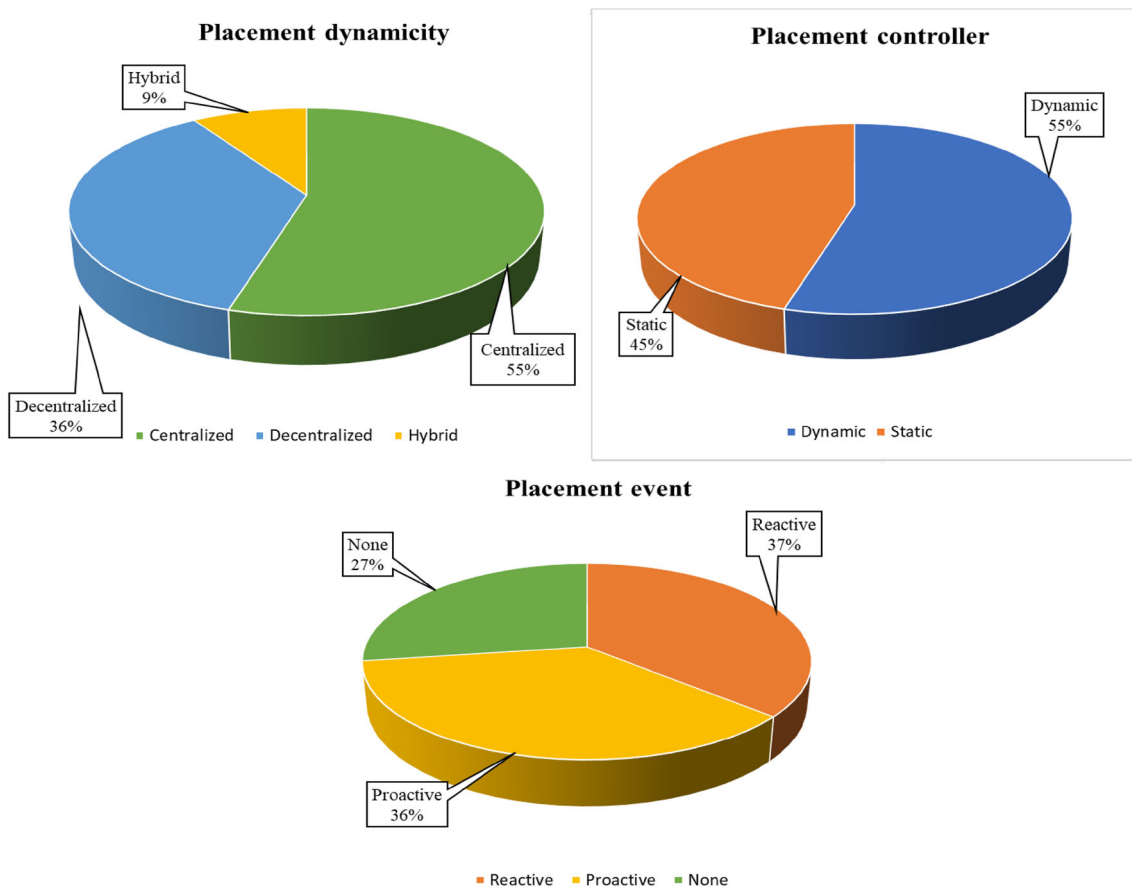


Fig. 11 Percentage of different heuristic-based data replica placement mechanisms

is 31, and graph-based algorithms ranked the last with only 14%.

- **TQ2:** Which case studies are considered in data replica placement approaches in the fog domain?

The applied case studies of the data replica placement methods in fog computing are shown in Fig. 14. Among the reviewed papers, case studies that have had an experimental result are as follows: machine learning (face recognition), Yahoo Cloud Serving Benchmark (YCSB), VIoLET: A Large-scale Virtual Environment for the Internet of Things, vehicle tracking, IoT information, mobile devices, smart cities, and smart industry. We categorized all examples either randomly generated or tree-based topology in a general category. Without considering the general category, we can see mobile cases, smart city, and the smart industry is more popular than other cases with 13%. Also, as we have reviewed prior papers, we can see the popularity of these subjects is considerably increased.

- **TQ3:** What evaluation factors are usually applied in data replica placement approaches in the fog domain?

We have studied some specifications, analyzed and compared them in each paper. Each paper had multiple factors of our study. This analysis has been illustrated in Fig. 15, showing that the network latency is the most considered factor in the studied papers with 21%, and the number of users is the least considered factor with only 1%. This indicates that a number of users are not a good candidate as an evaluation factor. Although when we scrutinized all of them, we found out that almost all of the metrics obtained a limited range between 10 and 20%.

- **TQ4:** What validation tools are used for determining the data replica placement approaches in the fog domain?

According to Fig. 16, 31% of the research papers used a testbed for their proposed model. Also, JAVA and Hadoop tools showed an equal share of 14% of studied papers. iFogSim has been used in 10% of papers, and the contribution of the remaining tools is insignificant in this literature, between 3 to 7%.

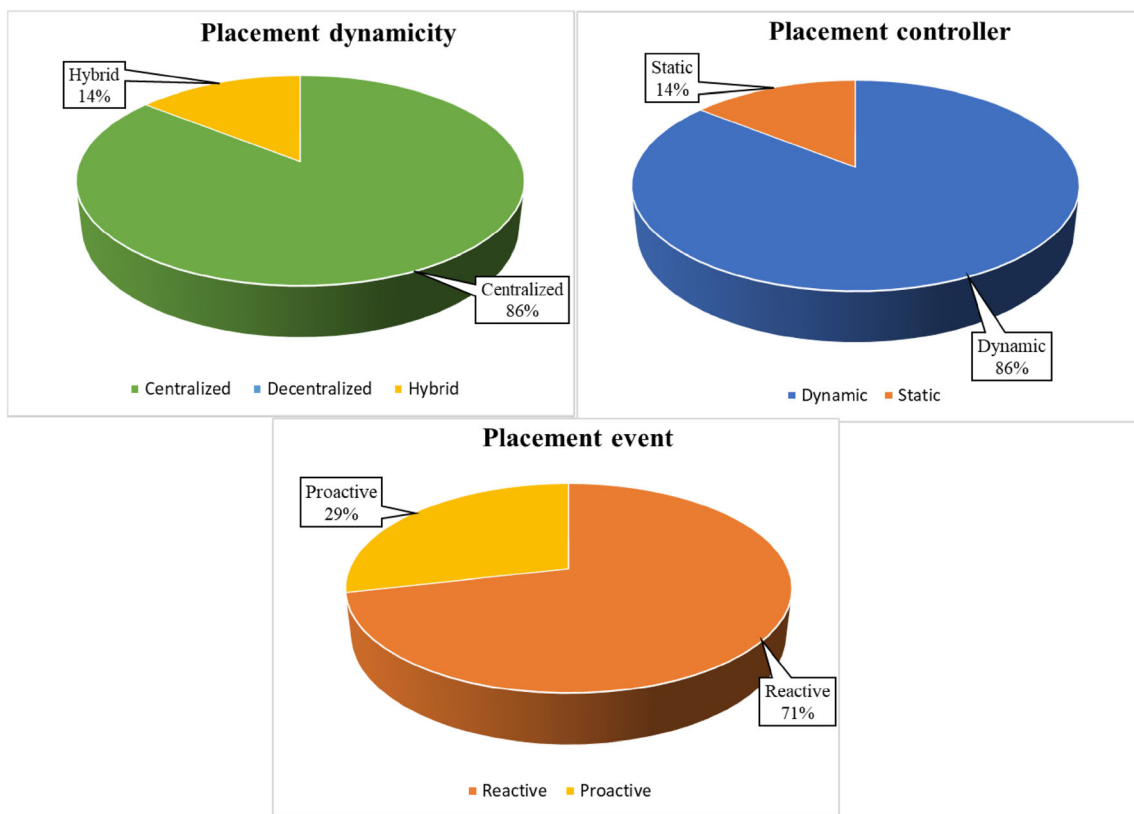
- **TQ5:** What utilized techniques are applied in data replica placement approaches in the fog domain?

Table 9 A side-by-side of different aspects for meta-heuristic-based data replica placement solutions

Refs.	Utilized technique	Evaluation Tools	Placement			Performance metrics						
			Dynamicty	Controller	Event	network	Average latency	Scalability	Workload	Number of replicas	Number of users	
						Latency	Bandwidth					
[31]	Heuristic Elite-guided Multi objective Artificial Bee Colony	Python 2	Centralized	Static	Proactive	*	*	*	*	*	*	*
[35]	Gray Markov chain	Hadoop	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*
[36]	Metaheuristic approach based on particle swarm optimization	JAVA	Centralized	Dynamic	Proactive	*	*	*	*	*	*	*
[37]	Ant colony	Testbed	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*
[38]	Integer programming Biomimetic optimization Particle swarm	JAVA	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*
[39]	Pareto optimal solution	Hadoop	Centralized	Dynamic	Reactive	*	*	*	*	*	*	*

Table 10 A side-by-side comparison of main idea, advantage, and disadvantage of meta-heuristic-based papers

Refs.	Main idea	Advantages	Disadvantages
[31]	Data-intensive Service Edge deployment scheme based on Genetic Algorithm (DSEGA)	Studying service and data placement together	Their defined problem is not close enough to the real world
[35]	A replica management policy for reducing energy in edge computing systems	Proposing a solution for reducing energy consumption	Lack of comparing performance with dynamic algorithms (only comparing results with static algorithms)
[36]	Zero-one integer programming problem with heuristic ITO algorithm	Considering workflow deadline and data reliability	Lack of properly managing cluster resources and lack of considering data dependency
[37]	The dynamic replica creation algorithm (DRC-GM) and the data replica placement algorithm (RP-FNSG)	Determining optimal node placement by considering file availability, network transmission cost, node relative load, cluster load, and node performance	Testing in a small environment
[38]	Dynamic Replica Creation Algorithm (DRCA), Data Replica Scheduling Algorithm (DRSA)	Presenting a collaborative edge and cloud computing strategy	Lack of presenting any computational scheduling
[39]	Gray Markov chain, sorting genetic algorithm	Presenting a delay-adaptive replica recovery algorithm	Their experimental environment is not large enough

**Fig. 12** Percentage of different meta-heuristic based data replica placement mechanisms

According to Fig. 17, 24% of the research papers have not specified a utilization technique for their proposed model. Heuristic-based methods ranked second as a

utilization technique, with 18% of total papers. And other methods are just used in one or two reviewed papers.

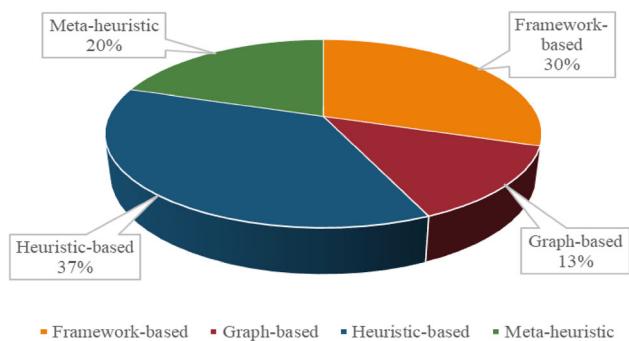


Fig. 13 Classification of data replica placement solutions

7 Open issues

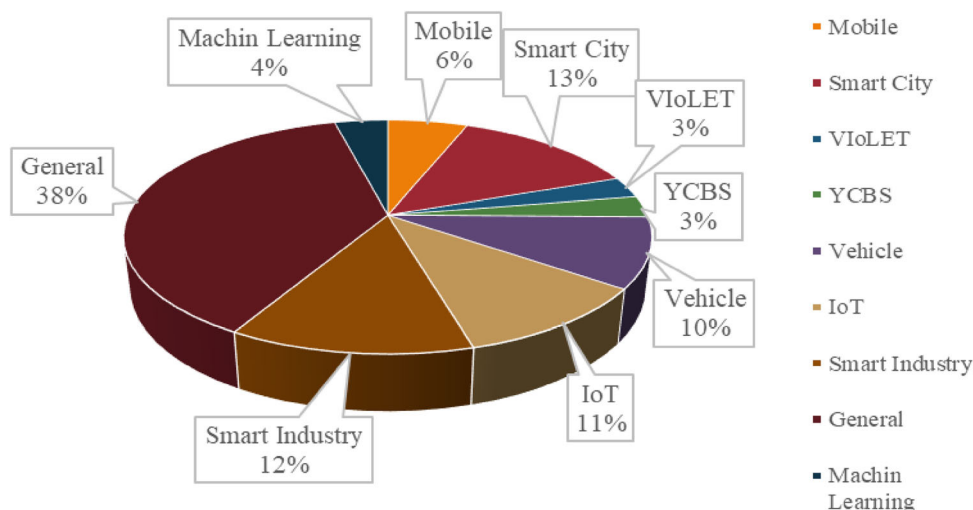
In order to minimize network latency and bandwidth utilization, one of the best approaches is to replicate data in different places. This section is dedicated to discussing open issues in the literature of resource management in fog computing concerning algorithmic and architectural challenges.

- **TQ6:** What are the future research directions and open perspectives of data replica placement methods in the fog domain?

We have extracted four different aspects, including privacy and security, scalability and mobility, energy efficiency, and heterogeneity as major future directions. Privacy and security-related features of data like authorization, authentication, and external attacks are required to be guaranteed within the data replica placement procedure for fulfilling the requirements and increasing the performance parameters [19].

Figure 18 represents an illustration of open issues in the field of data replica placement in fog computing. In this section, open issues, which are extracted through applying

Fig. 14 Percentage of different case studies that used in selected papers



the SLR method in research papers in the area of fog data management, are discussed, and future directions related to TQ6 response are presented.

7.1 Privacy and security

In fog data replica placement, the user’s data is outsourced and user’s control over data is handed over to the fog node, which introduces the same security threats as it is in cloud computing. First, it is hard to ensure data integrity, since the replicated data could be lost or incorrectly modified. Second, the replicated data could be abused by unauthorized parties for other interests. Therefore, there are new challenges in designing a secure replica placement system to achieve low latency, support dynamic operation and deal with the interplay between fog and cloud.

The existence of fake fog nodes while replica placement process will be a big threat to user data security and privacy. This problem is hard to address in fog computing due to the complex trust situation the calls for different trust management schemes and dynamic creating, deleting of virtual machine instances make it hard to maintain a blacklist of the rogue node.

In fog computing data management, location privacy mainly refers to the location privacy of the fog clients. As a fog client usually replicates its tasks to the nearest fog node, the fog node, to whom the tasks are replicated, can infer that the fog client is nearby and farther from other nodes. Furthermore, if a fog client utilizes multiple fog services at multiple locations, it may disclose its path trajectory to the fog nodes, assuming the fog nodes collude. As long as such a fog client is attached to a person or an important object, the location privacy of the person or the object is at risk.

Also, due to Fog nodes locations, (usually close to IoT devices, which means relatively weak protection and

Fig. 15 Percentage of different evaluation factors of articles

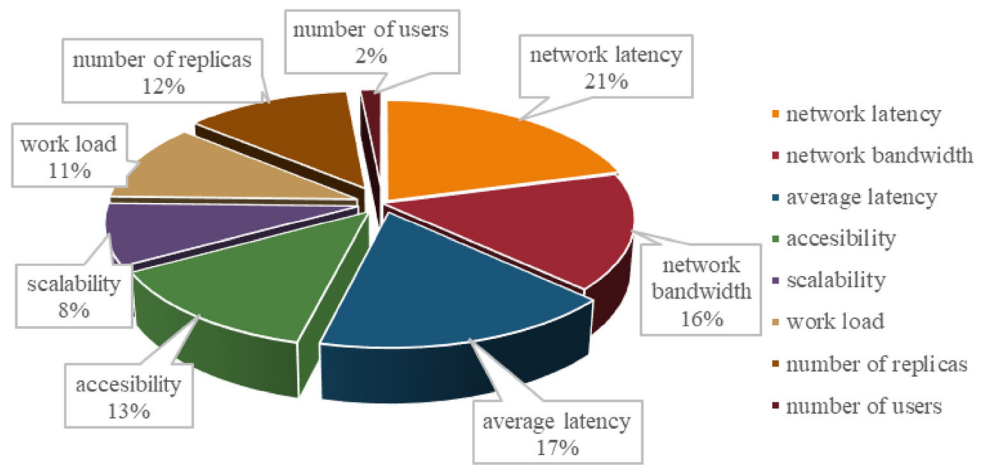


Fig. 16 Percentage of validation tools used in replica management papers

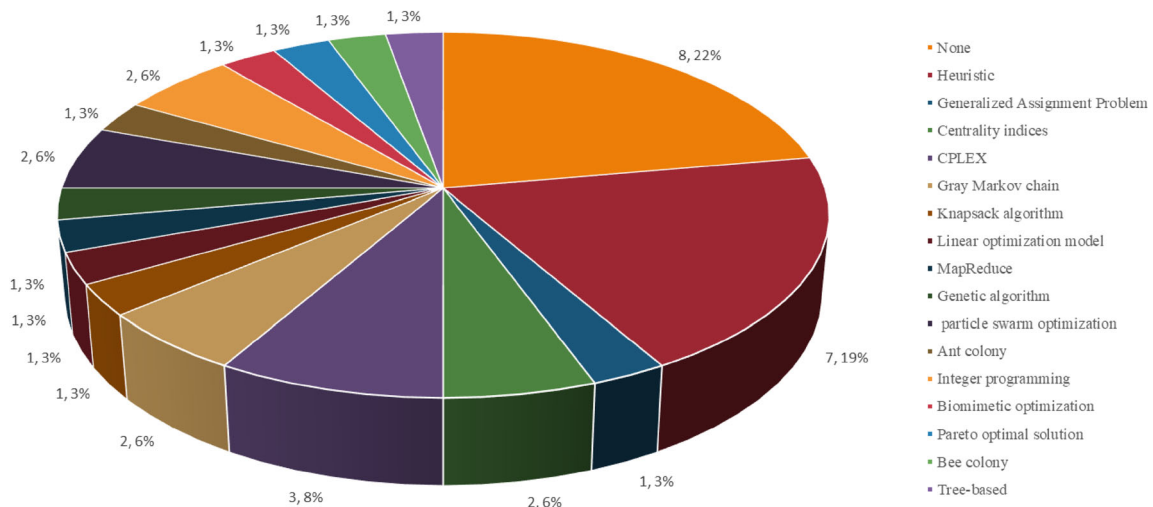
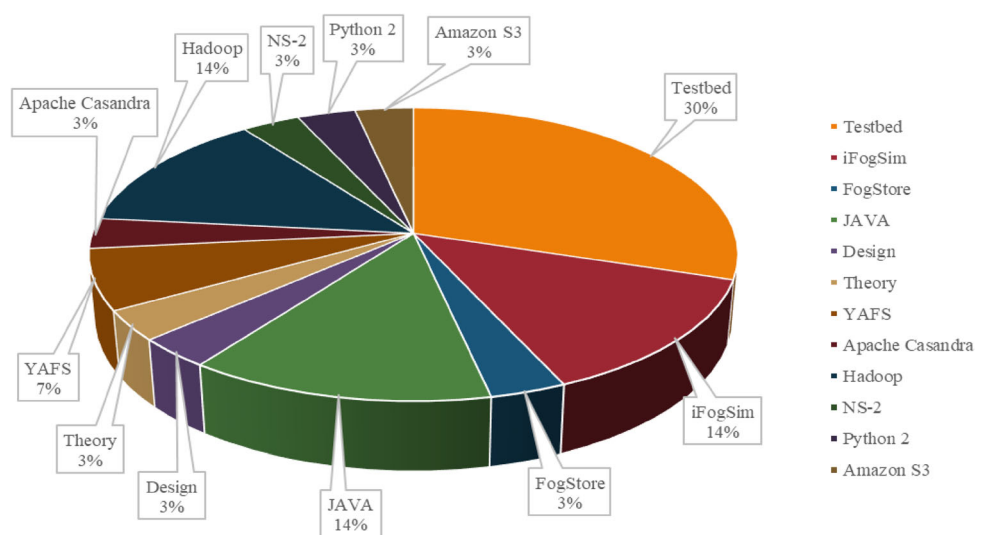


Fig. 17 Percentage of different utilization techniques used in selected papers

monitoring) fog will be easier and more accessible than the cloud, which increases the likelihood of attacks [5].

Security issues become even more challenging for the user applications, being used at the edge of the network (i.e., IoT or device layer). For this reason, these programs need to provide approved and limited access. Protecting these applications from unauthorized access is an open issue. Among the reviewed papers, only Xiao et al. have addressed the security issue, which protected modeling against programmable radio devices [50].

Despite the importance of the security issue, there are not enough articles on which adequately address this, so this issue is considered an important open issue.

7.2 Scalability and mobility

The data replica placement needs to quickly scale to meet workload demands, thus providing horizontal and vertical scalability. Horizontal scalability refers to the ability to increase capacity by adding more fog nodes or setting up a new cluster or a new distributed environment. Vertical scalability, on the other hand, refers to the increase of capacity by adding more resources to a machine, server, or a fog node (e.g., more memory or an additional CPU).

One of the major challenges in this area is the dynamic placement of computational tasks and the development of online methods for allocating computing resources. Also, work with the IoT-cloud fusion virtual machine to determine if the IoT virtual machine decides to place processes in the CPU unit based on predicted decisions or to act differently [52].

Modern networks like fog and edge are characterized by their mobility capacity since there probably are the nodes being able to move in any direction, or the nodes, being able to be added or removed at any time, and, as a result, the network topology frequently changes [15]. Therefore, this entails studying data replication in mobile and scalable environments.

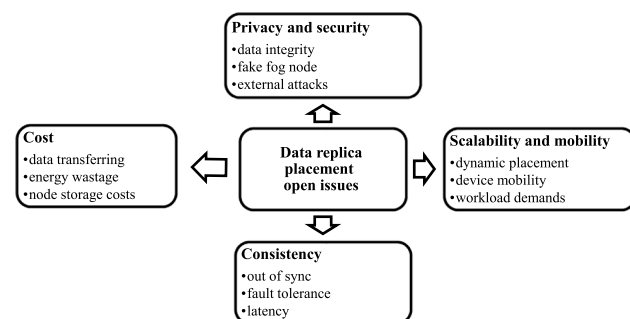


Fig. 18 The main challenges of data replica placement in fog computing

Accordingly, various essential concerns such as bandwidth, routing, and protocols for such environments should be taken into account. Moreover, there are various decision-making methods for data delivery such as Markov Decision-making Process (MDP) and game theory, most of which yield different results depending on the study.

In the computational paradigm, protocols are parts of computational codes along with some additional recognition data. Since these units (i.e., moveable nodes) are transferred by one/more hops and in an iterative manner towards the destination, applied protocols are required to include essential parts and capabilities to consider factors such as load and cost of the path, mobility, and bandwidth to deliver the downloadable content with regard to QoS constraints successfully.

Specifically for load factor, it is noteworthy that, in addition to selecting the shortest path to the goal with the lowest cost of the path, maintaining the balance of the load is essential. Since bandwidth is reserved for mobile device users, this can be rendered as a resource for them to compete for access to their downloadable content. Therefore, inspecting bandwidth is a major part of load balancing.

Routing algorithms are designed to find the shortest possible path from mobile devices to the landing-place (i.e. remote devices) for performing calculations. Nevertheless, since unloading the network and servers' workload are not deterministic, mobility management using reinforcement learning techniques would be highly applicable.

As the location of edge servers is fixed, this fixed equipment might become overloaded periodically and lead to inefficiency in the computing environment. In order to gain a tradeoff between the large volumes of the workload of fixed servers in the computing environment and achieving an acceptable response time, the idea of applying Unmanned Aerial Vehicles (UAV) has gained more attention among researchers. Regarding the inherent mobility of UAVs, some inefficiencies regarding resource management of UAVs might occur in the system, which requires presenting appropriate routing solutions.

7.3 Consistency

Data replica placement over a number of fog nodes can potentially cause inconsistencies. If you're replicating data at different times and only on certain nodes, the chance out-of-sync data is high, and it can be difficult to get every location back on the same page. Admins should create a customized replication process and always check on each server or node location to ensure consistency across the world. So the main problem to be addressed is how to ensure consistency of data across distributed copies. In essence, ensuring whenever a client connects to a new

replica, that replica is up to date according to the previous accesses of the client to the same data in the other replicas on different sites is an open issue.

Fault tolerance: The data replication must be able to recover in case of failure, e.g., by providing a backup instance of the application that will be ready to take over without disruption.

Low latency: The data replication must handle latency issues by measuring and testing the network latency before it copies the data that an application changed and before it makes such data available to other applications.

7.4 Cost

This study showed that most of the articles did not pay much attention to the cost optimization problem in data replication scenarios. Therefore, efficient solutions to achieve energy efficiency along with cost optimization in data processing and transmission are still needed.

There is a huge lack of papers to propose algorithms in order to maximize the objective function of the cost replication problem, the nodes must estimate the cost of storing replicas as well as the latency expected to make a decision to migrate or replicate to one of the neighbors. Also, such algorithms decide to delete local replicas or data copies. Creating more replicas may increase data availability, but may result in higher costs due to inefficient use of resources. We need the algorithms that allow the user to control the balance between cost optimization and latency optimization. In addition, there is a lack of research for a replica discovery method where concerned nodes are notified of nearby replicas. Some experimental results showed us that these distributed replica placement algorithms offer significant advantages in terms of cost, and latency compared to non-replicated and client-side caching approaches.

Also, **energy efficiency** in the environment of interference computing, data compression, and caching methods are new concerns in the literature of data replica placement in fog computing. Data compression is a method for improving data replication performance and consequently reducing energy consumption by reducing data size during the computation procedures. Data compression can be achieved with techniques like eliminating unnecessary or redundant data. Since the data replica placement is supposed to be run on the destination device, the destination operating system should also support data decompression methods. Interestingly, data compression can either be done entirely between available computing layers, i.e., fog layer and cloud layer, or entirely in the edge/fog layer.

In these cases, one part of the data replica can be compressed remotely and another one locally. Merging divided sections is the responsibility of the end device.

These three steps (e.g., compression, compression pressure, and integration) cause complications and overload in the system, leading to further delays for real-time applications. Therefore, aspects of the problem of minimizing delays should be taken into account in this regard.

8 Conclusion

In this paper, we present an investigation and analysis of data replica placement in the fog computing environment. This study was conducted in October 2021 with restrictions in the period from 2017 to 2021. Search results were 464 articles in total. By studying some key sections (i.e., abstracts, contributions, and conclusions) in the first stage, and then the remaining sections of articles, 30 articles related to data replica placement have been included in the review. Statistics based on the year of publication in data replica placement show the analysis done in this research. The number of research studies in the fog computing scope has an upward trend from 2017 to 2021. This growth rate shows the importance of data replica placement in this literature recently. According to TQ1, the research studies in data replica placement in fog environments were classified into four categories. The framework-based solutions create and use frameworks for solving the data replica placement problem. A graph-based algorithm creates a graph of fog nodes and divides the problem's domain into smaller sections to tackle the problem. The heuristic-based algorithms solve problems faster and more efficiently by fairly sacrificing accuracy, completeness, and optimality. In this way, we reach a proper execution time with acceptable accuracy. Heuristic algorithms are often used to solve NP-complete problems, including data replica placement. Meta-heuristic-based algorithms combine standard heuristic strategies with higher-level frameworks to explore a search space effectively. These techniques can be combined with conventional algorithms, implemented easily, and solved in an acceptable time. We may not have reviewed all available research papers in the case of the SLR-based process. As a result, non-peer-reviewed articles, non-English articles, editorial articles, book chapters, and survey articles were dismissed. Also, there are no recent efforts for comparing the discussed mechanisms to the best of our knowledge. Also, by comparing the methods discussed, there is no other way to locate the data replicas. So, a comprehensive approach to solving all of these issues is an important challenge and an interesting direction for future research and work.

Author contributions ET, AS, MGA conducted this research. ET: Methodology, Software, Validation, Writing original draft. AS:

Conceptualization, Supervision, Writing review & editing, Formal analysis, Project administration. MGA: Investigation, Resources, Data curation, Visualization.

Funding This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest We certify that there is no actual or potential conflict of interest in relation to this article.

References

- Guo, J., Li, C., Luo, Y.: Fast replica recovery and adaptive consistency preservation for edge cloud system. *Soft Comput.* (2020). <https://doi.org/10.1007/s00500-020-04847-2>
- Nikoui, T.S., Rahmani, A.M., Tabarsaied, H.: Data management in fog computing. In: *Fog and Edge Computing*, Hoboken: Wiley, 2019, pp. 171–190
- Tabet, K., Mokadem, R., Laouar, M.R., Eom, S.: Data replication in cloud systems. *Int. J. Inf. Syst. Soc. Chang.* **8**(3), 17–33 (2017). <https://doi.org/10.4018/IJISSC.2017070102>
- Jamali, M.A.J., Bahrami, B., Heidari, A., Allahverdizadeh, P., Norouzi, F.: IoT architecture BT. *Towards Internet Things* **21**, 9–31 (2020)
- Rani, R., Kumar, N., Khurana, M., Kumar, A., Barnawi, A.: Storage as a service in Fog computing: a systematic review. *J. Syst. Archit.* **116**, 102033 (2020). <https://doi.org/10.1016/j.sysarc.2021.102033>
- Fersi, G.: *Fog Computing and Internet of Things in One Building Block: A Survey and an Overview of Interacting Technologies*, vol. 4. Springer, New York (2021)
- Heidari, A., Navimipour, N.J.: A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm. *PeerJ Comput. Sci.* **7**, 1–21 (2021). <https://doi.org/10.7717/PEERJ-CS.539>
- Shakarami, A., Ghobaei-Arani, M., Shahidinejad, A., Masdari, M., Shakarami, H.: *Data replication schemes in cloud computing: a survey*. Springer, New York (2021)
- Qin, Y.: When things matter: a survey on data-centric Internet of Things. *J. Netw. Comput. Appl.* **64**, 137–153 (2016)
- Buyya, R., Dastjerdi, A.: Fog computing: helping the internet of things realize its potential. *Computer (Long. Beach. Calif)* **49**(8), 112–116 (2016)
- Aberer, K., Sathe, S., Papaioannou, T.G., Jeung, H.: A survey of model-based sensor data acquisition and management. In: Aggarwal, C.C. (ed.) *Managing and Mining Sensor Data*. Springer, Boston (2013)
- Azad, K.M., Pramanik, I., Lau, R., Demirkan, H.: Smart health :Big data enabled health paradigm within smart cities. *Expert Syst. Appl.* **87**, 370–373 (2017)
- Noel, T., Karkouch, A., Mousannif, H., Al-Moatassime, H.: Data quality in Internet of Things: a state-of-the-art survey. *J. Netw. Comput. Appl.* **73**, 57–81 (2016)
- Sharma, S.K., Wang, X.: Live data analytics with collaborative edge and cloud processing in wireless IoT networks. *IEEE Access* **5**, 4621–4635 (2017)
- Naas, M.I., Parvedy, P.R., Boukhobza, J., Lemarchand, L.: IFogStor: an IoT data placement strategy for fog infrastructure. In: *2017 IEEE 1st International Conference on Fog and Edge Computing. IC FEC 2017*, pp. 97–104, 2017, <https://doi.org/10.1109/ICFEC.2017.15>.
- da Silva, D.M.A., Asamoosing, G., Orrillo, H., Sofia, R. C., Mendes, P.M.: An analysis of fog computing data placement algorithms. *arXiv Comput. Sci.*, (2020), arXiv:2005.11847v1.
- Karatas, F., Korpeoglu, I.: Fog-based data distribution Service (F-DAD) for Internet of Things (IoT) applications. *Futur. Gener. Comput. Syst.* **93**, 156–169 (2019). <https://doi.org/10.1016/j.future.2018.10.039>
- Milani, B.A., Navimipour, N.J.: A comprehensive review of the data replication techniques in the cloud environments: major trends and future directions. *J. Netw. Comput. Appl.* **64**, 229–238 (2016). <https://doi.org/10.1016/j.jnca.2016.02.005>
- Moysiadis, V., Sarigiannidis, P., Moscholios, I.: Towards distributed data management in fog computing. *Wirel. Commun. Mob. Comput.* (2018). <https://doi.org/10.1155/2018/7597686>
- Mansouri, N., Javidi, M.M.: A review of data replication based on meta-heuristics approach in cloud computing and data grid. *Soft Comput.* (2020). <https://doi.org/10.1007/s00500-020-04802-1>
- Mazumdar, S., Seybold, D., Kritikos, K., Verginadis, Y.: A survey on data storage and placement methodologies for Cloud-Big Data ecosystem. *J. Big Data* **6**(1), 15 (2019). <https://doi.org/10.1186/s40537-019-0178-3>
- Sadri, A.A., Rahmani, A.M., Saberikamarposhti, M., Hosseinzadeh, M.: Fog data management: a vision, challenges, and future directions. *J. Netw. Comput. Appl.* **174**, 102882 (2021). <https://doi.org/10.1016/j.jnca.2020.102882>
- Islam, M.S.U., Kumar, A., Hu, Y.-C.: Context-aware scheduling in Fog computing: a survey, taxonomy, challenges and future directions”. *J. Netw. Comput. Appl.* **180**(1), 103008 (2021). <https://doi.org/10.1016/j.jnca.2021.103008>
- Heidari, A., Navimipour, N.J.: Service discovery mechanisms in cloud computing: a comprehensive and systematic literature review. *Kybernetes* (2021). <https://doi.org/10.1108/K-12-2020-0909>
- Hiebl, T., Hochreiner, C., Schulte, S.: Towards a framework for data stream processing in the fog. *Inform. Spektrum* **42**(4), 256–265 (2019). <https://doi.org/10.1007/s00287-019-01192-z>
- Naas, M.I., Lemarchand, L., Raipin, P., Boukhobza, J.: IoT data replication and consistency management in fog computing. *J. Grid Comput.* **19**(3), 1–25 (2021). <https://doi.org/10.1007/s10723-021-09571-1>
- Huang, T., Lin, W., Li, Y., He, L.G., Peng, S.L.: A latency-aware multiple data replicas placement strategy for fog computing. *J. Signal Process. Syst.* **91**(10), 1191–1204 (2019). <https://doi.org/10.1007/s11265-019-1444-5>
- Li, C., Tang, J., Luo, Y.: Scalable replica selection based on node service capability for improving data access performance in edge computing environment. *J. Supercomput.* **75**(11), 7209–7243 (2019)
- Saranya, N., Geetha, K., Rajan, C.: Data replication in mobile edge computing systems to reduce latency in internet of things. *Wirel. Pers. Commun.* **112**(4), 2643–2662 (2020). <https://doi.org/10.1007/s11277-020-07168-7>
- Qureshi, N.M.F., et al.: An aggregate MapReduce data block placement strategy for wireless IoT edge nodes in smart grid. *Wirel. Pers. Commun.* **106**(4), 2225–2236 (2019). <https://doi.org/10.1007/s11277-018-5936-6>
- Chen, Y., Deng, S., Ma, H., Yin, J.: Deploying data-intensive applications with multiple services components on edge. *Mob. Netw. Appl.* **25**(2), 426–441 (2020). <https://doi.org/10.1007/s11036-019-01245-3>
- Vales, R., Moura, J., Marinheiro, R.: Energy-aware and adaptive fog storage mechanism with data replication ruled by spatio-

- temporal content popularity. *J. Netw. Comput. Appl.* **135**(351), 84–96 (2019). <https://doi.org/10.1016/j.jnca.2019.03.001>
33. Li, C., Bai, J., Tang, J.H.: Joint optimization of data placement and scheduling for improving user experience in edge computing. *J. Parall. Distrib. Comput.* **125**, 93–105 (2019). <https://doi.org/10.1016/j.jpdc.2018.11.006>
 34. Li, C., Wang, Y.P., Tang, H., Luo, Y.: Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud. *Futur. Gener. Comput. Syst.* **100**, 921–937 (2019). <https://doi.org/10.1016/j.future.2019.05.003>
 35. Li, C., Wang, Y.P., Chen, Y., Luo, Y.: Energy-efficient fault-tolerant replica management policy with deadline and budget constraints in edge-cloud environment. *J. Netw. Comput. Appl.* **143**(152–166), 2019 (2018). <https://doi.org/10.1016/j.jnca.2019.04.018>
 36. Shao, Y., Li, C., Tang, H.: A data replica placement strategy for IoT workflows in collaborative edge and cloud environments. *Comput. Netw.* **148**, 46–59 (2019). <https://doi.org/10.1016/j.comnet.2018.10.017>
 37. Li, C., Wang, Y.P., Tang, H., Zhang, Y., Xin, Y., Luo, Y.: Flexible replica placement for enhancing the availability in edge computing environment. *Comput. Commun.* **146**, 1–14 (2019). <https://doi.org/10.1016/j.comcom.2019.07.013>
 38. Shao, Y., Li, C., Fu, Z., Jia, L., Luo, Y.: Cost-effective replication management and scheduling in edge computing. *J. Netw. Comput. Appl.* **129**, 46–61 (2019). <https://doi.org/10.1016/j.jnca.2019.01.001>
 39. Li, C., Song, M., Zhang, M., Luo, Y.: Effective replica management for improving reliability and availability in edge-cloud computing environment. *J. Parall. Distrib. Comput.* **143**, 107–128 (2020). <https://doi.org/10.1016/j.jpdc.2020.04.012>
 40. Monga, S.K., Ramachandra, S.K., Simmhan, Y.: ElfStore: A resilient data storage service for federated edge and fog resources. *2019 IEEE International Conference on Services Computing*, pp. 336–345, 2019, <https://doi.org/10.1109/ICWS.2019.00062>.
 41. Mayer, R., Gupta, H., Saurez, E., Ramachandran, U.: FogStore: toward a distributed data store for fog computing. *2017 IEEE Fog World Congr. FWC 2017*, pp. 1–6, 2018, <https://doi.org/10.1109/FWC.2017.8368524>
 42. Breitbach, M., Schafer, D., Edinger, J., Becker, C.: Context-aware data and task placement in edge computing environments. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom, Mar. 2019*, pp. 1–10, <https://doi.org/10.1109/PERCOM.2019.8767386>.
 43. Confais, B., Parrein, B., Lebre, A.: A tree-based approach to locate object replicas in a fog storage infrastructure. *2018 IEEE Global Communications Conference*, pp. 1–6, (2018), <https://doi.org/10.1109/GLOCOM.2018.8647470>.
 44. Lera, I., Guerrero, C., Juiz, C.: Comparing centrality indices for network usage optimization of data placement policies in fog devices. *2018 3rd International Conference on Fog and Mobile Edge Computing FMEC 2018*, pp. 115–122, 2018, <https://doi.org/10.1109/FMEC.2018.8364053>.
 45. Confais, B., Parrein, B., Lebre, A.: Data location management protocol for object stores in a fog computing infrastructure. *IEEE Trans. Netw. Serv. Manag.* **16**(4), 1624–1637 (2019). <https://doi.org/10.1109/TNSM.2019.2929823>
 46. Aral, A., Ovatman, T.: A decentralized replica placement algorithm for edge computing. *IEEE Trans. Netw. Serv. Manag.* **15**(2), 516–529 (2018). <https://doi.org/10.1109/TNSM.2017.2788945>
 47. Hasenburg, J., Grambow, M., Bermbach, D.: Towards a replication service for data-intensive fog applications. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 267–270, <https://doi.org/10.1145/3341105.3374060>.
 48. Naas, M.I., Lemarchand, L., Boukhobza, J., Raipin, P.: A graph partitioning-based heuristic for runtime IoT data placement strategies in a fog infrastructure. In: *Proceedings of the Symposium on Applied Computing*, pp. 767–774, 2018, <https://doi.org/10.1145/3167132.3167217>.
 49. Hasenburg, J., Grambow, M., Bermbach, D.: FBase: a replication service for data-intensive fog applications. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing* pp. 267–270, 2019, <https://doi.org/10.1145/3341105.3374060>.
 50. Gupta, H., Xu, Z., Ramachandran, U.: DataFog: towards a holistic data management platform for the IoT age at the network edge. *USENIX Work. Hot Top. Edge Comput. HotEdge 2018, co-located with USENIX ATC 2018*, 2018.
 51. Guerrero, C., Lera, I., Juiz, C.: Optimization policy for file replica placement in fog domains. *Concurr. Comput.* **9**(1–20), 2019 (2018). <https://doi.org/10.1002/cpe.5343>
 52. Taghizadeh, J., Ghobaei-Arani, M. & Shahidinejad, A. An efficient data replica placement mechanism using biogeography-based optimization technique in the fog computing environment. *J Ambient Intell Human Comput* (2021). <https://doi.org/10.1007/s12652-021-03495-0>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Esmaeil Torabi received the B.S. degree in software engineering from Shahid Sattari Aeronautical University of Tehran, Iran in 2006, the M.S. degree in software engineering from Payame Noor University of Tehran, Iran, in 2017. He is currently a Ph.D candidate of computer engineering in Islamic Azad University (IAU), Qom branch. His research interests include edge/fog computing, replica in fog computing, internet of things.



Mostafa Ghobaei-Arani received the PhD Degree in Software Engineering from Islamic Azad University, Science and Research Branch, Tehran, Iran. He is Assistant Professor of Computer Engineering Department, Qom Branch, Islamic Azad University, Qom, Iran. He has published more than 70 journal and conference papers in the area of distributed computing. His research interests include distributed computing, cloud computing, autonomic computing, edge/fog computing, serverless computing, soft computing, and the IoT.



Ali Shahidinejad received the B.S. degree in computer hardware engineering from Islamic Azad University of Kashan, Iran in 2008, the M.S. degree in computer architecture from Islamic Azad University of Arak, Iran, in 2010 and Ph.D. degree in Computer Networks at the Universiti Teknologi Malaysia/RWTH Aachen University, Malaysia/Germany, in 2015. He joined the Department of Computer Engineering, Islamic Azad University of

Qom, as an Assistant Professor. His research interests include cloud

computing, fog computing, edge computing, internet of things, optical wireless communications.