Load Balancing Algorithms in Fog Computing

Mostafa Haghi Kashani, *Member, IEEE* and Ebrahim Mahdipour

Abstract—Recently, fog computing has been introduced as a modern distributed paradigm and complement to cloud computing to provide services. The fog system extends storing and computing to the edge of the network, which can remarkably solve the problem of service computing in delay-sensitive applications besides enabling location awareness and mobility support. Load balancing is an important aspect of fog networks that avoids a situation with some under-loaded or overloaded fog nodes. Quality of service parameters such as resource utilization, throughput, cost, response time, performance, and energy consumption can be improved by load balancing. In recent years, some research in load balancing algorithms in fog networks has been carried out, but there is no systematic study to consolidate these works. This article investigates the load-balancing algorithms systematically in fog computing in four classifications, including approximate, exact, fundamental, and hybrid algorithms. Also, this article investigates load balancing metrics with all advantages and disadvantages related to chosen load balancing algorithms in fog networks. The evaluation techniques and tools applied for each reviewed study are explored as well. Additionally, the essential open challenges and future trends of these algorithms are discussed.

Index Terms—Fog computing, load balancing, quality of service, Internet of Things

1 Introduction

Fog computing, which extends from the cloud and is a geographically distributed paradigm, brings networking power and computing into the network edge, closer to both end-users and IoT devices because of being supported by wide-spread fog nodes [1]. Most of the data in cloud-only architectures, requiring processing, analysis, and storage, are transmitted to the cloud servers, which may have an influence on latency, security, mobility, and reliability adversely. With the existence of location-aware and delay-sensitive applications, the cloud on its own comes across some problems to meet the extremely low latency requirements of these applications; the proximity of the fog layer to the Internet of Things (IoT) devices may remarkably decrease latency and meet the needs of extremely-low [2], [3]. Fog computing always interacts with and supports the cloud, creating a novel generation of applications and services.

Nowadays, in fog computing environments, users need applications that give quick responses whenever they want to access anything and work fast. To improve Quality of Service (QoS) factors in a fog network significantly, we can use an efficient load balancing strategy because load balancing is regarded as an important issue. Many studies have been done to balance the cloud computing load because the load on the cloud increases enormously [4]. The fog networks, which are heterogeneous and dynamic, cannot directly apply most of the load balancing algorithms of cloud computing; the goal of load balancing in a fog environment is to distribute the coming load between available fog nodes or cloud, based on one mechanism, to avoid overload or underload of fog nodes. This mechanism can maximize throughput, performance, and resource utilization while minimizing response time, cost, and energy consumption.

We have had no Systematic Literature Review (SLR) of any research on load balancing algorithms in fog computing, which makes it hard to evaluate and identify the gaps in studies, different trends, and specifically future dimensions of load balancing in the fog environment. In addition, regarding the ever-increasing need for load balancing in fog computing, we are required to investigate a research agenda for load balancing algorithms in fog computing. An SLR can identify, categorize, and synthesize a comparative review of state-of-the-art studies. It also makes knowledge transfer possible in the research community [5], [6]. This SLR is conducted with the aim of *identification, taxonomic classification, and systematic comparison of the existing research that focus on planning, executing, and validating the fog systems load balancing*. We specially aim at answering the questions below by conducting a methodological overview of the existing studies:

- What are the main practical motivations for load balancing in fog computing?
- Which kind of classification in research approaches can be applied in fog systems load balancing?
- What evaluation metrics are applied in load balancing algorithms of fog computing?
- What are popular evaluation tools applied in load balancing algorithms in fog computing?
- What measurement techniques are used to assess the load balancing in fog computing?
- What are the open issues, future trends, and challenges of load balancing in fog computing?

Guidelines in [5], [7], [8] were followed. Our purpose is to have a systematic identification and taxonomic classification of the evidence available on load balancing algorithms in fog computing and to have a holistic comparison to analyze the limitations and potentials of the existing studies. It

- *The authors are with the Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran 1477893855, Iran. E-mail: {mh.kashani, mahdipour}@srbiau.ac.ir.*

Manuscript received 10 Sept. 2020; revised 9 Jan. 2022; accepted 5 May 2022. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Ebrahim Mahdipour.)
Digital Object Identifier no. 10.1109/TSC.2022.3174475

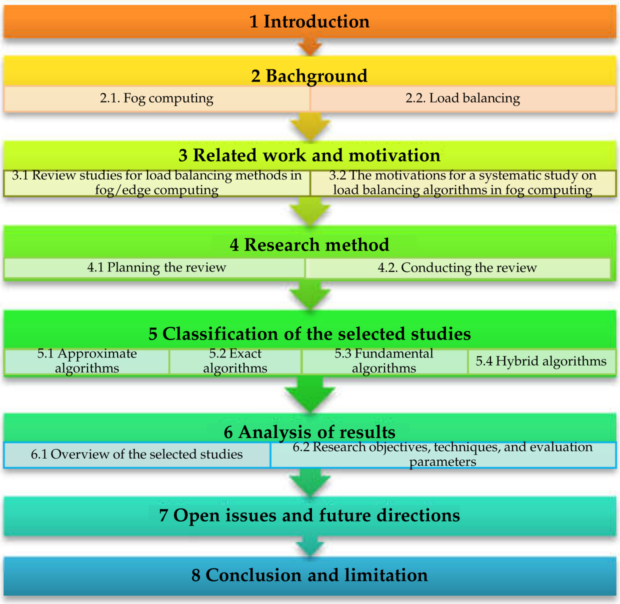

Fig. 1. The structure of this SLR.

makes an SLR of the current study by concentrating on the suggested techniques, solutions, and algorithms in load balancing in fog computing. Therefore, 49 studies are chosen, categorized, and compared by applying a characterization taxonomy. The characterization taxonomy is composed of four groups, including approximate, exact, fundamental, and hybrid, which are derived and refined by following a qualitative evaluation of the included research, some famous references [9], and our current experience with previous systematic studies [10], [11], [12], [13], [14], [15], [16], [17], [18]. The research synthesis led to a knowledge base of recent research mechanisms, techniques, algorithms, experiences, and best practices that were applied in load balancing in fog computing. Furthermore, open challenges and future trends related to load balancing algorithms in fog systems are discussed. The results related to this systematic study are beneficial for

- Scholars in fog/cloud systems who require the identification of relevant research. Presenting the research systematically procures a corpus of knowledge that is necessary for developing theories and solutions, analyzing research implications, and establishing future directions.
- Practitioners who are eager to understand the available techniques and algorithms with tool support and their limitations in supporting load balancing algorithms in fog-based environments.

The structure of the study is organized as Fig. 1: Section 2 shows the background of fog computing, load balancing, and metrics definition. In Section 3, surveys related to this study are presented. Section 4 illustrates the methodology of the research. Also, Section 5 depicts the selected load balancing algorithms in fog computing in four categories. Section 6 refers to the results and comparisons of techniques, and then, in Section 7, open issues are outlined. In the end, Section 8 shows the conclusions and limitations of this study.

2 Background

The concept and structure of fog computing (Section 2.1) and load balancing (Section 2.2) are discussed and explained in this part.

2.1 Fog Computing

Due to the unprecedented amount of data and the connection of over 50 billion devices to the Internet (based on Cisco estimation), handling that much data with traditional computing models, like cloud computing and distributed computing, is difficult [19]. Often privacy gaps, high communication delay, and related network traffic loads that connect cloud computing to end-users for unpredictable reasons with the recent expansion of services related to IoT (like smart healthcare, smart homes, smart cities, industrial scenarios, smart transportation systems [10], [12]) are some challenges that affect cloud computing performance. To refer to some of cloud computing limitations and to bring cloud service traits so much closer to “Things”, as it is referred to, including cars, mobile phones, embedded systems, and sensors, the research community has suggested the fog computing concept [1], [20].

Fog computing is regarded as a platform bringing cloud computing to end-users’ vicinity. “Fog”, as a term, has an analogy with real-life fog and was initially introduced by Cisco [1]. When the fog is nearer to the earth, clouds are up above in the sky, and, interestingly, fog computing applies this concept when the virtual fog platform is located closer to end-users, just between end-users’ devices and the cloud. In a similar definition, fog computing is suggested to make computing possible at the network edge, to send new services and applications specifically for the Internet future [21].

Fog computing is most often mistaken for edge computing, but we have major differences between the two [22]. Fog computing applications are run in a multi-layer architecture that disconnects and meshes the software and hardware functions, permitting the dynamic reconfigurations for diverse applications while executing transmission services and intelligent computing. Edge computing, on the other hand, creates a direct transmission service and manages special applications in a fixed logic location. While fog computing is hierarchical, edge computing is limited to a few peripheral devices. Besides networking and computation, fog computing deals with the control, storage, and acceleration of data processing [2], [23]. There are various use cases for both architectures. For example, fog use cases are like surveillance systems, transportation systems, and smart cities [22]. Edge use cases are similar to cloud gaming, predictive maintenance, virtualized radio networks, and content delivery [24]. An IoT client or smart end-device, to recognize fog computing from other computing standards, needs to utilize the following characteristics, but not all of them, while consuming a fog computing service [22], [25]: 1) *Low latency and contextual location awareness*, 2) *geographical distribution*, 3) *heterogeneity*, 4) *federation and interoperability*, 5) *real-time interactions*, 6) *fog-node clusters*, 7) *predominance of wireless access*, and 8) *support for mobility*.

Fog computing architecture reference model is an important study topic. Recently, a wide range of architectures has been suggested for fog computing, mostly obtained from a structure with three layers [12], [22]. A fog network expands

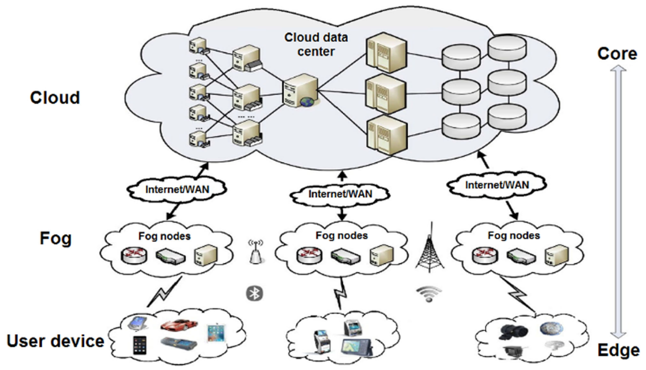

Fig. 2. The architecture of the fog network [11].

cloud services to the network edge by suggesting a fog layer between cloud and user devices. As it can be observed, Fig. 2 illustrates the fog architecture hierarchically [22], having three layers as follows:

- *Cloud layer*: The layer of cloud computing is composed of different storage devices and high-performing servers and creates several services of applications. It bears robust storage and computing abilities to back the permanent storage of a large amount of information and extended computation analysis.
- *Fog layer*: The fog layer is located at the network edge, which consists of a couple of fog nodes like access points, routers, switches, and gateways. They are spread between cloud and end devices.
- *User device*: The layer of a user device is so close to the physical environment and end-user. This layer is composed of different IoT devices, like, sensors, cellphones, smart automobiles, cards, and readers.

Here in the architecture, all end devices or smart objects are connected with fog nodes by technologies with wired or wireless connection access such as 3G, 4G, wireless LAN, ZigBee, Bluetooth, and Wi-Fi. Wireless or wired communication technologies to help the interconnection and intercommunications of fog nodes. IP core network helps fog nodes each to be linked with the cloud [11], [17].

2.2 Load Balancing

In a fog system, load balancing facilitates the distribution of workload on resources equally, aiming to provide services continually if the service component fails, and it is done by provisioning and de-provisioning instances of applications along with proper resource utilization. Because the computation and storage resources in fog nodes are so varied, an appropriate load balancing mechanism is needed in a fog system to improve application performance and to make optimal utilization of network resources possible. To evade any overload or under-load on resources, load balancing, as a mechanism, spreads the workload onto different resources. Load balancing, which distributes the load among different resources, is implemented either in physical equipment or software [26].

The load balancing has some goals, including throughput maximization, response time minimization, and traffic optimization. Consumption optimization in the server-side resources, request processing time minimization, and scalability improvement in the distributed environment are some other purposes of the technique of load balancing [27]. In fog networks, load balancing may have various methods that can be of static or dynamic nature or both. In static methods, with primary information about the system as a necessary feature, the rule should be programmed in the load balancer because the user's behavior is not predictable, and methods of static load balancing are not necessarily efficient in the network. Further, the dynamic methods outperform the static methods because of the dynamic distribution of load based on the pattern that is programmed in the load balancer [28]. Mechanisms of dynamic load-balancing apply the current system state to this end, and they use special policies, including [29]:

- *Transfer*: It defines the conditions based on which a task has to be sent from one node to the other. The arriving tasks that enter the transfer policy are transferred or processed based on a determining rule that relies on each node workload. The policy deals with task migration and rescheduling.
- *Selection*: It defines whether a task should be sent or not and also regards a couple of elements to select a task, like the amount of overhead needed for migration, time of task execution, and total nonlocal system calls.
- *Location*: It defines under-loaded nodes and then sends tasks to these nodes. In aimed nodes, the availability of essential services for task rescheduling or migration is checked.
- *Information*: The complete information, considering system nodes, is collected in this policy and used by other policies to make a decision. This policy determines the time at which the information should be collected. Various policies have some relationships that are mentioned below:

Transfer policy grabs incoming tasks and determines whether to transfer them to a remote node to balance load or not. If not eligible to be transferred, the task will be locally processed. When the transfer policy decides to transfer a task, the location policy would be triggered to locate a remote node to process the task. The task is locally processed when a far partner cannot be found, or the task is transferred to a remote node. The necessary information is provided by the information policy for location and transfers to help them make a decision.

Several metrics are needed to assess an algorithm of load balancing and weigh it in comparison with algorithms before showing which algorithm is better and recognizing the pros and cons related. The metrics need some qualitative paradigms. Various qualitative metrics are used in articles, like response time, cost, and energy. The essential metrics for load balancing in fog computing are stated below:

- *Response time*: This issue is described by the interval starting from the acceptance of a request (or task) to the response to a request for a server in a fog environment.
- *Cost*: It is the payment of money to ask for an action that is required to do.

- *Energy consumption*: It refers to the energy consumption amount in a fog network. Energy consumption can be decreased by an effective load balancing algorithm.
- *Scalability*: It shows how the system is capable of accomplishing a load balancing algorithm with a couple of hosts or machines.
- *Security*: It is the quality side of service that procures non-repudiation and confidentiality via authentication involving parties and message encryption.
- *Flexibility*: Fog nodes that always connect to a system pro tempore incline to leave periodically, so, to reflect both nodes, which are newly joined, and nodes revocation, this algorithm has to be flexible.
- *Resource utilization*: It represents the maximum utilization of the resources available in a cloud system.
- *Deadline*: It is the latest time when a service request in the fog system can be completed.
- *Processing time*: The duration in which a service request in the fog system is executed entirely.
- *Reliability*: It is the ability of a fog network to perform its required requests in a defined time and a specified condition.
- *Throughput*: We can refer to the maximum requested service rate that might be processed in the fog system as throughput.
- *Availability*: A rise in resource application or service requests can maintain the system performance that shows the capability of a computing system.

3 Related Work and Motivation

In this section, we study the existing review works for load balancing in fog/edge computing (Section 3.1) based on a systematic exploration (Section 3.2).

3.1 Review Studies for Load Balancing Methods in Fog/Edge Computing

No systematic review was found on load balancing algorithms in fog computing. Thus, the decision was made to investigate the review studies existing on load balancing mechanisms in fog/edge computing to conduct this systematic review. The studies were summarized in Table 1 as review studies based on surveys [30], [31], [32], [33] and SLR [34].

Chandak and Ray [30] presented a survey of load balancing techniques in fog computing. They also introduced some of the evaluation parameters and simulation tools used for load balancing methods in the fog system. In addition, Baburao, *et al.* [31] surveyed some of the techniques of service migration, load balancing, and load optimization in fog computing. Pydi and Iyer [32] reviewed load balancing methods in edge systems, including security-based, traffic load-based, optimization-based, heuristic-based, joint load-based, multi-access-based, allocation-based, dynamic load-based, and distributed-based techniques. They introduced some of the evaluation factors and tools. Also, the authors discussed some of the future trends and research gaps. Singh *et al.* [33] reviewed load balancing methods based on energy efficiency in the fog environment. Furthermore, Kaur and Aron [34] surveyed load balancing approaches systematically in a fog environment.

TABLE 1
Related Reviews of Fog/Edge Load Balancing

Study type	Ref.	Scope	Publication year	Article selection process	Open issue	Covered year
Surveys	[30]	Fog	2019	×	×	×
	[31]	Fog	2019	×	×	×
	[32]	Edge	2020	×	✓	×
	[33]	Fog	2020	×	×	×
SLRs	[34]	Fog	2021	✓	✓	2013-20
	Our study	Fog	2022	✓	✓	2013-21

Table 1 presents a summary of studied articles that depicted some parameters like the type of reviews, main ideas, year of publications, the process of article selection, open issues, and covered years of every study. As for the search string and research methodology, the work of Kaur and Aron [34] is the closest to ours; nevertheless, we specifically examined load balancing algorithms in fog computing. We also presented a different taxonomy and reviewed recently published articles up to 2021.

3.2 The Motivations for a Systematic Study on Load Balancing Algorithms in Fog Computing

Needing a systematic review leads to the *identification*, *classification*, and *comparison* of the existing evidence on the load balancing algorithms in a fog environment. It concentrates on the classification and comparison of load balancing algorithms in the fog system. In order to show that a resembling review has not been reported, we surfed the Google Scholar (on 1/9/2021) with the search strings below:

(*fog* <OR> *edge*) **[AND]**
(*load* <OR> *balancing*) **[AND]**
(*survey* <OR> *review* <OR> *overview* <OR> *challenges* <OR> *trends* <OR> *issues* <OR> *study*)

Among the obtained review studies, no one was related to any of our research questions in Section 4. Regarding the significance of load balancing algorithms in fog networks, the consolidation of the existing evidence on load balancing algorithms in the fog system is necessary.

4 Research Method

Contrary to the procedure in a non-structured review, a systematic review [5] decreases the partiality and follows an exact sequence of methodological stages to research literature. A systematic review depends on truly defined and assessed review protocols for extracting, analyzing, and documenting the results. We obtained the guidelines in [5], [35], [36] with a three-stage study procedure, including *planning, conducting, and documenting*. The study is accomplished by an external assessment of the results of each phase. A clear classification of the reviewed studies is provided, which is a foundation for a comparative analysis of research based on the dimensions of our analysis that are also subject to an external assessment. The planning and conducting stages of the methodology are summarized to

TABLE 2
Research Questions

RQ 1-What are the main practical motivations for load balancing in fog computing?
RQ 2-Which kind of classification in research approaches can be applied in fog systems load balancing?
RQ 3-What evaluation metrics are applied in load balancing algorithms of fog computing?
RQ 4-What popular evaluation tools are applied in load balancing algorithms in fog computing?
RQ 5-What measurement techniques are used to assess the load balancing in fog computing?
RQ 6-What are open issues, future trends, and challenges of load balancing in fog computing?

perform this systematic review. In terms of data summary, the results are illustrated in Section 5, and for findings and research implications the results are depicted in Section 6. Based on the proposed taxonomy, the data are collected and synthesized, as explained in Section 5.

4.1 Planning the Review

Planning begins with knowing the motivations for a systematic study and the results in a review protocol as defined below:

4.1.1 Identify the Motivations for the Systematic Review.

In Section 3.2, we identify the motivation and justify the contribution of this systematic review.

4.1.2 Specifying the Research Questions

The research questions (RQs) define our motivation, i.e., answers give us an evidence-based review of load balancing algorithms. Six research questions are defined that clarify the basis for obtaining the strategy of the search for extracting literature, as shown in Table 2. The aim of investigating each question is outlined by motivation. A comparative analysis, on the other hand, permits an analysis of the collective influence of the research, which is presented in terms of comparison features.

4.1.3 Define and Evaluate Review Protocol

A protocol was developed for a systematic study by following [5] and our experience with systematic reviews [10], [11], [12], [13], [14], [15], [18]. This protocol includes a set of RQs, the selection process of primary studies, and data extraction described in Sections 4.1.2, 4.2.1, and 4.2.2, respectively. Based on the purposes, the RQs and the study scope were specified to make the search strings for extracting literature. As proposed in [7], [35], the protocol was externally assessed before execution. An external expert was asked for feedback who was experienced in conducting systematic reviews in a field that overlapped with fog computing. The feedback given is reflected in a defined protocol. A pilot study of the systematic survey was performed, containing 20 percent of the included research. The purpose of this pilot study was primarily reducing the partiality among the researchers and improving the characterization method for data collection. The study scope was expanded, the search methods were improved, and the exclusion/inclusion criteria were refined during the experimental studies.

TABLE 3
Search Results on Digital Libraries

No	Academic database	Result
1	ACM	24
2	IEEE	217
3	ScienceDirect	151
4	Springer	42
5	Google Scholar	542
6	Taylor & Francis	7
7	Wiley	71
	Total	*1054*

4.2 Conducting the Review

The second phase is conducting that starts with article selection and leads to data extraction and information synthesis:

4.2.1 Select Primary Studies

Conducting, as the second phase of research methodology, begins with selecting articles and leads to data extraction. This subsection aim presents the procedures of searching and choosing articles in the second phase of the systematic review. We follow a two-step guideline to select the articles:

- *Initial selection.* The search strings that follow are found among academic databases to locate articles having these strings in their abstracts, titles, and keywords. Accordingly, famous online academic databases like ACM, IEEE, ScienceDirect, Springer, Google Scholar, Taylor & Francis, and Wiley are used. 1054 articles were extracted primarily (see Table 3). We also regarded the online-published articles from 2013 to August 2021. 2013 was chosen because fog computing was introduced in 2012 [1].

*fog [**AND**]*
(load <OR> balancing <OR> balanced <OR> balancer)

Final selection. We examined 1054 articles extracted from the previous step and applied the inclusion/exclusion criteria (as mentioned in Table 4). Next, we investigated the articles fully and applied quality assessment; only articles that had mentioned the assessment details and techniques explicitly were chosen; therefore, 49 related studies were finally selected to be evaluated qualitatively.

4.2.2 Data Extraction and Synthesis

As mentioned in Section 4, a structured format was designed and followed [5] based on characterization dimensions to record the obtained data from chosen articles. By investigating the limitations and potentials of current studies and the reflections on future studies, an organized comparative analysis was formed, which provided us with an investigation of the collective research impact.

TABLE 4
Inclusion/Exclusion Criteria

Inclusion	• Research articles that present algorithms or innovative solutions on load balancing in fog computing • Peer-reviewed articles in conferences and JCR-indexed journals • Articles published between 2013 and August 2021
Exclusion	• Review articles, editorial articles, short articles (less than six pages), white articles, and non-English articles • Research articles that do not mention solutions and algorithms to improve load balancing in fog computing explicitly • Books, book chapters, and dissertations

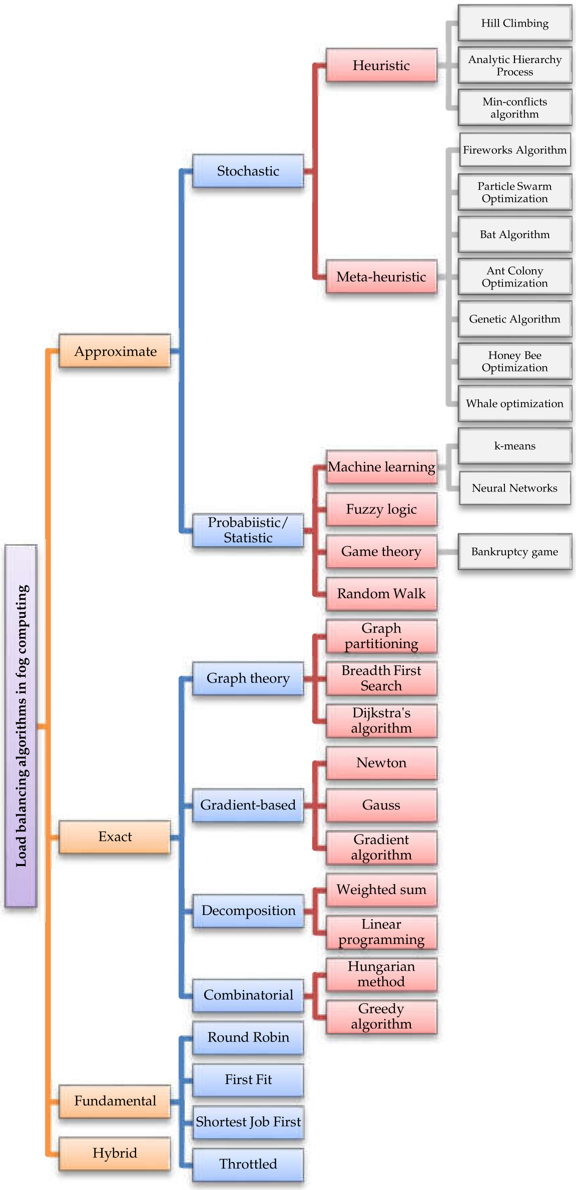

Fig. 3. Classification of load balancing algorithms in fog computing.

5 Classification of the Fog Load Balancing Studies

A structured classification of the related literature is defined here. Because of the diverse literature on load balancing algorithms of fog computing, it is not easy to structure the related works systematically. Based on the available literature, Fig. 3 shows the framework of the suggested classification scheme. Four main categories are recognized: approximate, exact, fundamental, and hybrid algorithms. It is a natural move to review the literature from these four perspectives since most studies in this domain deal with the issues related to one of these four perspectives, which allows the reviewed articles to be categorized under common umbrellas. However, other taxonomies are also possible (e. g. *centralized*, *semi-distributed*, and *distributed*, or *system state* and *who initiated the process*). This section reviews 49 selected articles based on the criteria mentioned before, and their main features, differences, evaluation parameters, tools, pros and cons, and evaluation techniques are defined.

5.1 Approximate Algorithms

In this part, studies related to approximate algorithms, including stochastic, probabilistic, and statistic techniques, are performed. In Sections 5.1.1 and 5.1.2, we investigate stochastic algorithms, including heuristic and meta-heuristic related to the research field, respectively. Then, in Section 5.1.3, probabilistic/statistic algorithms are reviewed.

5.1.1 Heuristic Algorithms

Heuristic algorithms are totally made by "experience" for special optimization problems, intending to find the best solution to the problem through "trial-and-error" in an optimal amount of time [37]. The solutions in heuristic approaches might not be the best or optimal; however, they can be much better than an educated guess. Heuristic approaches make use of the problem particularities. As exact approaches consume a huge amount of time to get the optimal solution, heuristic approaches are preferable, gaining near-optimal solutions in an optimal amount of time [37]. Some of the heuristic methods in the literature reviewed include Hill Climbing [38] Min-conflicts [39], and Analytic Hierarchy Process [40]. In this section, selected heuristic-based algorithms are discussed.

For three-layered architecture, Zahid *et al.* [38] proposed a framework that was composed of a distributed fog layer, a centralized cloud layer, and a consumer layer. A hill-climbing load-balancing method, decreasing the processing time (PT) and response time (RT) of fogs to consumers, was suggested here. There was a tradeoff, though, in RT and microgrid cost, so this article aims to procure the request load balancing on fog nodes. In addition, Kamal *et al.* [39] presented a load balancing scheduling algorithm as Min-conflicts scheduling. This algorithm makes use of a heuristic method to solve a constraint satisfaction problem. The architecture proposed is composed of cloud, fog, and end-users in three layers.

In order to reduce the delay of data streams from IoT devices to the applications, Banaie, *et al.* [40] proposed multiple vacation-based queuing systems to model the performance of a fog system. To speed up user's access to sensor data, they leveraged multi-gateways architecture along with a resource caching policy in the IoT domain. A load-balancing scheme based on the analytic hierarchy process (AHP) method was also employed to provide global load fairness among the network entities. Further, Oueis, *et al.* [41] focused on improving users' quality of experience by referring to load balancing in a fog environment. They considered multiple users that require computation offloading, in which the whole requests have to be processed by local computation cluster resources. Also, Xu *et al.* [42] suggested a virtual machine (VM) scheduling mechanism to balance the load in the cloud-fog system.

5.1.2 Meta-Heuristic Algorithms

A meta-heuristic method, as a higher-level heuristic method, is problem-independent and can be applied to a wide range of problems. Today's "Meta-heuristics" indicate all modern higher-level methods [37]. We have two major parts in modern meta-heuristics: diversification and intensification [43]. It is important to have a balance between diversification and intensification to gain an influential and effective meta-heuristic method. A meta-heuristic method investigates the whole solution space; a different set of solutions should be produced, and the search has to be heightened near the neighborhood of the optimal or near-optimal solutions. Some of the meta-heuristic algorithms in the literature reviewed include Particle Swarm Optimization (PSO) [44], [45], [46], Fireworks Algorithm [47], Bat Algorithm [48], Whale Optimization Algorithm [49], and Hybrid metaheuristic [50].

He, *et al.* [44] presented the fog and software-defined network (SDN) to tackle the challenges. For the sufficient use of the SDN and cloud/fog architecture on the Internet of vehicle, they presented an SDN-based modified constrained optimization particle swarm optimization method. Also, based on the fog network, Wan *et al.* [45] suggested an energy-aware load balancing and scheduling approach. Firstly, an energy consumption model was proposed on the fog node that was related to the workload, and then an optimization function was formulated to balance the manufacturing cluster load. Then, they applied an improved PSO method to gain a good solution and achieve tasks; they had to prioritize the manufacturing cluster. In [46], considering load balancing, the authors proposed a resource allocation method based on PSO in a fog environment.

Shi, *et al.* [47] proposed fog and SDN architecture to the cloud-based mobile face recognition architecture for solving the delay problem. They also formulated the load balancing in the SDN and fog/cloud system as an optimization problem and proposed using a firework algorithm (FWA) based on SDN centralized control for solving the load balancing problem. Furthermore, Yang [48] proposed a three-layered architecture based on a fog/cloud network and big medical data, including cloud, fog, and medical devices. In the proposed architecture, the bat algorithm used the load balancing strategy to perform the initial setup of bat population data, which improved the quality of the solution in the initial sample. In [49], regarding load balancing, the authors suggested a whale optimization algorithm in a microgrid-connected wireless sensor network and fog environment. In addition, Qun and Arefzadeh [50] presented a load balancing approach using a hybrid metaheuristic algorithm in fog-based vehicular ad hoc networks.

5.1.3 Probabilistic/statistic Algorithms

In this section, load balancing algorithms based on probabilistic/statistic algorithms, including machine learning [51], fuzzy logic [52], game theory [53], and Random Walk [54], are discussed.

Li, *et al.* [51] examined the fog infrastructure runtime features and proposed a self-similarity-based load balancing (SSLB) technique for large-scale fog systems. They proposed an adaptive threshold policy and a corresponding scheduling method to guarantee SSLB efficiency successfully. Further, Singh, *et al.* [52] introduced a load balancer based on fuzzy logic using different levels of tuning and designing of fuzzy controls in fog networks. The proposed fuzzy logic model was used to conduct link analysis as interconnects for managing traffic. Abedin, *et al.* [53] formulated a fog load balancing problem to minimize the load balancing cost of a fog environment empowered by narrow-band Internet of things (NB-IoT). Firstly, the time resource scheduling problem in NB-IoT was modeled as a Bankruptcy game. Subsequently, the transportation problem was solved by applying Vogel's approximation technique that locates a feasible load balancing solution to guarantee optimal assignment of jobs in the fog environment. In [54], the authors introduced a distributed approach for fog load balancing based on random walk method. The classification of the above-mentioned articles and essential factors, in analyzing the approximate load balancing algorithms in fog computing is depicted in Table 5.

5.2 Exact Algorithms

Exact algorithms can optimally solve optimization problems. Each optimization problem might be solved by applying the exact search, but the bigger the size of the instances, the more time it takes forbiddingly to get the optimal solution. The exhaustive search is considerably slower than the exact algorithms [55]. Some of the exact algorithms in literature reviewed include graph theory [56], [57], [58] gradient-based [59], [60], [61], decomposition [62], [63], [64], [65], and combinatorial [66], [67]. In this section, the studied articles that are based on exact algorithms are summarized below:

The graph partitioning theory was used by Ningning, *et al.* [56] to make the fog computing load balancing method on the basis of dynamic graph partitioning. The authors showed that, after cloud atomization, the fog computing framework could flexibly build the system network, and the dynamic load balancing mechanism is capable of configuring the system and reducing node migration caused by system changes. In addition, a load balancing technique was suggested by Puthal *et al.* [57] to validate the edge data centers (EDCs) and find less loaded EDCs for the allocation of tasks. This technique is more useful than other techniques in locating less-loaded EDCs for the allocation of tasks. It not only increases the efficacy of load balancing but also improves security via destination EDCs authentication. In [58], the authors proposed Dijkstra's algorithm for the load balancing problem in vehicular fog computing.

Moreover, a workload balancing model was presented by Fan and Ansari [59] in a fog network for minimizing the data flow latency in processing procedures and communications through the association of IoT devices to appropriate base stations. Barros, *et al.* [60] used fog computing as a means of reducing the logical distance between consumption spot and central distribution. IoT devices in the network edge have higher efficacy and lower cost to manage

TABLE 5
Approximate Load Balancing Algorithms in Fog Computing and Their Properties
(In the Evaluation Column, S = >Simulation, P = >Prototype, and N = >Not-mentioned)

Method		Article	Main idea	Evaluation	Tool(s)	Advantage(s)	Disadvantage(s)
Stochastic	Heuristic	[38]	Hill climbing load balancing algorithm based on fog system	S	CloudAnalyst	• Low response time • Low processing time	• Low scalability • Low security
		[39]	Load balancing in heuristic Min-conflicts optimizing method	S	CloudAnalyst	• Low response time • Low cost • Low latency	• High complexity • Requires experience and • knowledge
		[40]	A load-balancing scheme based on the AHP method for Multiple Gateways in a fog network	S	MATLAB	• Low response time • Low energy consumption	• High complexity • Low availability
		[41]	Resource management based on load distribution for fog clustering	S	Not-mentioned	• Customizable design • Low energy • Low complexity • Low latency	• Inability to recover from database corruption
		[42]	A heuristic VM scheduling mechanism to provide load balancing	S	CloudSim	• Avoid bottleneck • High resource utilization • Avoid overload	• Not supporting the positive and negative impacts of VM migration • Imbalances of the positive and negative impacts in service migration
	Meta-heuristic	[44]	Load balancing mechanism based on SDN in cloud/fog network	S	Not-mentioned	• Low response time • High mobility • Improve the QoS • Low latency	• Low security • Low scalability
		[45]	Scheduling and load balancing for fog-based smart factory	P	Work robots	• Optimal scheduling • Low energy	• The efficiency of broadcast mode is low
		[46]	Resource allocation and load balancing by particle swarm optimization in fog environment	S	OpenSSH, Arduino	• Low latency • High resource utilization	• Low security • High complexity
		[47]	Load balancing based on SDN in fog/cloud system	S	MATLAB	• Low response time • Low cost • Low latency	• Low performance
		[48]	A fog/cloud system and big medical data based on bat algorithm considering load balancing	S	MATLAB	• Low latency	• High complexity • The possibility of a bottleneck • Low scalability • Low reliability
		[49]	Whale optimization algorithm in fog and microgrid-connected wireless sensor networks considering load balancing	S	NS2	• Low energy • High throughput • Low delay	• High complexity
		[50]	Energy-aware load balancing by hybrid metaheuristic method in fog-based vehicular ad hoc networks	S	NS2	• Low energy • High scalability	• High complexity
Probabilistic/Statistic	Machine learning	[51]	Load balancing method in large-scale fog environment	S	iFogSim	• High scalability • Low response time • High flexibility • High resource utilization	• Low performance • High overhead
	Fuzzy logic	[52]	A load balancer based on fuzzy logic in fog computing	N	Not-mentioned	• Low energy consumption • Low latency	• Low reliability • Low security
	Game theory	[53]	Load balancing in fog network for great machine-type communications	S	Not-mentioned	• Low response time • Low energy • Low execution time	• High complexity
	Random Walk	[54]	Fog load balancing algorithm based on Random Walk	S	MATLAB, OMNeT++	• Low response time	• Low scalability

the power flow information. They evaluated the Gauss-Seidel and Newton-Raphson method performance aiming to develop computations in real-time of the load flow problem with the assistance of fog. Beraldi and Alnuweiri [61] studied load balancing among fog nodes and addressed the special challenges resulting from the fog system. Particularly, they applied randomized-based load balancing protocols that leveraged the power-of-random choice property. Based on parallel exploring, they proposed sequential probing contrasting the classical randomization protocols.

Chen and Kuehn [62] considered the downlink of the cache-enabled fog-radio access network (F-RAN) and investigated minimizing the consumption of power to communicate green. Based on channel states, an efficient load balancing algorithm was suggested. With the proposed algorithm, the increase in cache memory for greater content hitting rate was considered an economical method for achieving greener networks.

Further, Maswood, *et al.* [63] presented a Mixed-Integer Linear Programming (MILP) model in the fog/cloud environment to improve the bandwidth cost in routing, link utilization of network, and server resource utilization. They considered load balancing strategy jointly at the server level and the network both.

Also, Sthapit, *et al.* [64] proposed solutions for some situations in which the cloud or fog is unavailable. Firstly, by

TABLE 6
Exact Load Balancing Algorithms in the Fog System and Their Properties
(In the Evaluation Column, S = >Simulation and F = >Formal verification)

Method		Article	Main idea	Evaluation	Tool(s)	Advantage(s)	Disadvantage(s)
Graph theory	Graph partitioning	[56]	Dynamic load balancing technique in the fog network	S	JMeter	• Low cost • High flexibility	• High complexity • Extra overhead at execution time
	Breadth First Search	[57]	Load balancing technique in edge data centers and task allocation in fog environment	S, F	Scyther. MATLAB	• High security • Low latency	• Not proposing lightweight security solutions and not improving load balancing performance
	Dijkstra Algorithm	[58]	Load balancing of an unmanned surface vehicle in a fog system	S	C#	• Low response time • Reduce blocking probability	• Weakness in responding to complex tasks
Gradient_based	Gradient algorithm	[59]	Workload balancing scheme in fog/IoT model	S	Not-mentioned	• Low response time • Low energy	• Bottleneck • Low scalability • Low availability
	Newton-Raphson, Gauss-Seidel	[60]	Dynamic Load Flow in fog-based SGs	S	MATLAB	• High reliability • Low latency • Low cost	• High complexity • High execution time
	Fixed point algorithm	[61]	Load balancing method based on Sequential Randomization in fog nodes	S, F	Custom simulator with Python	• Low cost	• Low reliability
Decomposition	Weighted sum	[62]	Load balancing and radio unit selection in the downlink of F-RAN	S	Not-mentioned	• Low energy • Low cost	• Deactivation should be considered for lower operational energy consumption • Increasing the cache memory for higher content hitting rate
		[63]	A MILP model to improve resource utilization in a fog network considering load balancing	S	AMPL/ CPLEX	• High Resource utilization • Low cost	• Low scalability • Low availability
	Linear programming	[64]	Load balancing for computation in an edge system	S	NS-3	• High performance • Low energy	• The performance boost also comes at a similar energy cost
		[65]	Load balancing for minimizing total runtime and deadline in fog-based vehicle systems	S	Not-mentioned	• Low energy • Low response time • Low latency	• Low reliability • Priorities cannot be set
Combinatorial	Hungarian Method	[66]	Resource balancing scheme in F-RAN	S	Not-mentioned	• High reliability • Low response time • High throughput	• Low performance • In different network conditions, the amount of service migration is not considered
	Greedy algorithm	[67]	Load balancing mechanism for F-RAN	S	Not-mentioned	• Low response time • Low energy	• High execution time • Low reliability

using a network of queues, the sensor network was modeled, then, considering the load balancing, a linear programming technique was applied to make scheduling decisions. Chen, *et al.* [65], by using connected car systems as an evocative application, showed that mobility patterns of vehicles could be utilized for performing periodic load balancing in fog servers. A task model was presented by them for solving the scheduling problem at the server level, not at the device level.

Also, Dao, *et al.* [66] presented an adaptive resource balancing (ARB) model to maximize serviceability in F-RANs in which the resource block (RB) utilization within remote radio heads (RRHs) are balanced by applying the Hungarian method and backpressure technique, considering a time-varying network topology issued by potential RRH mobility. Further, a load-balancing scheme was proposed by Mukherjee, *et al.* [67] to define the tradeoff between computing delays and transmission in F-RANs.

Table 6 describes the classification of the aforementioned studies and the influential elements for analyzing the exact load balancing algorithms in fog computing.

5.3 Fundamental Algorithms

In the existing literature, some research on load balancing strategy in fog computing is based on simple algorithms without complex computations that are classified in the fundamental algorithms. They include such algorithm as Shortest Job First [68], Throttled, Round Robin (RR), First Fit [69], [70], and other algorithms [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81]. In this part, the selected fundamental algorithms are reviewed.

Nazar *et al.* [68] suggested a load balancing algorithm that modified the shortest job first (MSJF) to manage user's request load between VMs at the fog level aiming to optimize the suggested cloud and fog performance based on integrated architecture. In addition, Ahmad *et al.* [69] proposed an integrated cloud and fog-based platform to manage energy effectively in smart buildings. The first fit (FF) method was applied for load balancing that chooses VMs based on partitioning memory blocks. In the cloud/fog-based model, smart buildings, having many apartments, consist of IoT devices that were regarded. Also, a decentralized scheduling architecture was presented by Chekired, *et al.* [71] for energy management of electric vehicles (EVs) based on the fog system paradigm, where, by applying a priority-queuing model, an optimal load balancing algorithm (LBA) was performed.

Neto, *et al.* [72] proposed a Multi-tenant Load Distribution approach for Fog networks (MtLDF) regarding specific multi-tenancy needs like latency and priority. Further, they presented case studies to illustrate the proposed method applicability compared to a latency-driven load distribution mechanism.

TABLE 7
Fundamental Load Balancing Algorithms in the Fog System and Their Properties
(In the Evaluation Column, E = >Example Application, F = >Formal Verification, and S = >Simulation)

Method	Article	Main idea	Evaluation	Tool(s)	Advantage(s)	Disadvantage(s)
MSJF	[68]	The shortest job first-based method for providing load balancing	S	CloudAnalyst	• Low response time • Low cost	• Low performance • Longer processes have a more waiting time
Throttled, RR, First Fit	[69]	Resource allocation in fog/cloud system considering load balancing	S	CloudSim	• Low response time • Low energy	• High cost
LBA algorithm	[71]	Load balancing method in distributed fog architecture	S	NS-2 MATLAB	• High scalability • Low energy • Low response time • Low latency	• Not stable at evening peak hours • Standard case without scheduling is not stable at peak hours
MtLDF algorithm	[72]	A load-balancing method for fog Environment	E	Java platform	• Low energy • Low latency • Low cost	• Not improve the load balancing across the fog-cloud layers • Not consider some features such as disk I/O operations • Low resource utilization
FoT Load Balancing algorithm	[73]	Load balancing method in FoT System	S	Mininet	• Low response time • High performance in IoT	• Low scalability • Not developed a solution for IoT scenarios • The solution was not compared with traditional load-balancing algorithms
Throttled, RR	[70]	Load balancing based on priority in fog/cloud systems	S	CloudAnalyst	• Low response time • Low latency • Low energy • High flexibility	• Results compared to throttled are not much better
ELB algorithm	[74]	Effective data replication considering load balancing	S	CloudSim	• Low cost • Low response time • Low processing time	• Low security • Low privacy • Low reliability
LBSSA	[75]	Service scheduling and load balancing in fog/cloud environment	S	CloudSim	• High reliability • Low running time • Improve resource utilization	• Low security
LBS	[76]	Fog load balancing scheme in health monitoring systems	S	iFogSim	• Low latency • High scalability • High resource utilization	• Low reliability • Low security
LFA	[77]	Trust-aware offloading and load balancing in a fog network	S	Python	• Low latency • High security • Low complexity	• Low scalability
PriorityFogLB	[78]	Priority-based load balancing in fog nodes	S	Python	• Low response time	• Bottleneck • Low reliability
SFA and AFA algorithms	[79]	Distributed load balancing for fog-based smart cities	S, F	OMNeT++	• Low response time • Low loss rate	• Different dropping scenarios are not considered
DEER	[80]	Resource allocation scheme in fog system considering load balancing	S	CloudSim	• Low energy • Low cost	• Low reliability
Pricing-based workload distributor algorithm	[81]	The fog-federation system based on the pricing model in the vehicular network to provide workload balancing	S	AnyLogic	• Low energy • Low delay	• Low reliability

Furthermore, an approach was presented by Batista *et al.* [73] based on executing load balancing needs for the fog of things (FoT) platforms by programmability for distributed IoT environments by applying SDN. The authors addressed the problems by applying the FoT load balancing mechanism and assessed response time and lost samples as two metrics.

Tariq *et al.* [70] designed a fog-based environment to cover a vast area of six regions in the world, each regarded as a single region with many consumers that send requests on fog to access the needed resources. A load-balancing method was proposed to select VMs efficiently inside a fog system for the consumers to receive a fast response with minimum latency. As well as for fog-cloud-based architecture, Verma *et al.* [74] suggested an efficient load balancing (ELB) technique. It used the technique of information replication to maintain those data in fog networks that reduced the total dependency on massive data centers.

Many other research suggested several fundamental algorithms to provide load balancing in fog environments [75], [76], [77], [78], [79], [80], [81]. Table 7 describes the classification of the aforementioned studies and the influential elements for analyzing the fundamental load balancing algorithms in fog computing.

5.4 Hybrid Algorithms

To accomplish load balancing in fog networks, hybrid algorithms apply such various algorithms as approximate, exact, and fundamental [82], [83], [84], [85], [86], [87], [88]. Studies with hybrid algorithms are reviewed in this section.

Naqvi *et al.* [82] introduced fog computing to increase cloud computing processing speed, a cloud computing complement. Fog nodes, each having four to nine VMs, use the service broker policies for request processing.

The use of the Ant Colony Optimization (ACO) load balancing method, throttle, and RR balances VMs load. Also, Abbasi *et al.* [83] concentrated on fog computing application to a smart grid (SG) consisting of a distributed generation environment recognized as a microgrid. The aim of this study was to improve delay time, response time, and resource

TABLE 8
Hybrid Load Balancing Algorithms in the Fog System and Their Properties (In the Evaluation Column, S = >Simulation)

Method	Article	Main idea	Evaluation	Tool	Advantage(s)	Disadvantage(s)
RR, Throttle, ACO	[82]	Metaheuristic method in the fog-cloud system for load balancing	S	Java platform	• Low cost • Low response time	• High complexity • Low security
Throttled, RR, Particle Swarm Optimization	[83]	Load balancing method for load stabilization in fog environment	S	CloudAnalyst	• Low energy • Low response time	• The limitation of this article is that the results of the presented broker policy do not outperform the results of existing broker policies.
RR, Honey Bee Optimization	[84]	State-based load balancing method for SG energy management in fog	S	CloudAnalyst	• High performance • Low cost • Low response time • Low latency	• There are some changes of values in different fogs, but overall the average cost remains the same
RR, Throttled, Hybrid GA	[85]	Integration of fog-cloud based system for load balancing by applying bin packing and genetic algorithms	S	CloudAnalyst	• Low response time • Low cost • High security • Low latency	• High complexity • Difficult to show branching and looping
Weighted RR, Q-learning, GA	[86]	A load balancing strategy based on genetic algorithm and Q- learning in the fog-based healthcare system	S	MATLAB	• High Resource utilization • Low response time	• High complexity • Low scalability
Fuzzy logic, Probabilistic Neural Networks	[87]	Load balancing method in real-time fog system based on neural network and Fuzzy logic algorithms	S	iFogSim	• Low latency • Low cost • Low response time • Low energy	• QoS can be measured in terms of reduced IoT service delay • Can make it more distributed
Greedy and coalitional game-based algorithms	[88]	Task offloading and load balancing in a fog network	S	NS-3	• Low energy • Low delay	• Low scalability

utilization. The proposed VM load balancing technique performed better than the other techniques mentioned in the results.

Ali *et al.* [84] proposed a four-layered SG-based architecture to improve communication between consumers and an electricity company, and this model covers a huge area of residents. Three load balancing algorithms were applied for the allocation of VM, and the service broker policies applied for simulations were dynamically reconfigurable and the closest to data centers. For resource allocation, Zubair *et al.* [85] used Genetic Algorithm (GA), throttle, and RR for a load balancing mechanism by applying bin pack techniques. In this study, an SG was integrated with fog, and the cloud-based model and three places with some buildings were considered.

Moreover, based on the Q-learning and GA, Talaat *et al.* [86] presented a load balancing scheme using the dynamic resource allocation approach in a fog-based healthcare environment. The load balancing scheme continuously monitors network traffic, collects information about each server load, and controls incoming requests and distributes them among the existing servers applying a dynamic resource allocation method.

In addition, Talaat, *et al.* [87] presented an influential load balancing strategy (ELBS) for a fog system appropriate for applications in healthcare. ELBS attempted to attain important load balancing in a fog environment through caching algorithms and real-time scheduling. The authors presented ELBS for the fog environment appropriate for applications of the healthcare system. In [88], network considering load balancing, the authors suggested a task offloading approach based on greedy and coalitional game algorithms in a fog network.

Table 8 shows the classifications of the above-mentioned studies and the influential elements in analyzing the hybrid algorithms of load balancing in fog networks.

6 Analysis of Results

This section analyzes the results of the systematic review. We give a review of the selected articles in Section 6.1 and then introduce a comparison of the articles in Section 6.2 by answering the RQs.

6.1 Overview of the Selected Studies

The following complementary questions (CQs) are considered to examine the state of research on load balancing in fog networks:

- *CQ1*: What is the status of research on load balancing in fog networks?
- *CQ2*: When did research on load balancing in fog networks become active in the computing community?

6.1.1 *Temporal Overview of Studies*

As we can see in Fig. 4, 55% of the articles are associated with IEEE, 35% to Springer, 6% to ScienceDirect, 2% to ACM, and 2% to Wiley. Fig. 5 illustrates the overtime classification of the articles in every category, including IEEE, Springer, ScienceDirect, ACM, and Wiley published between 2013 up to August 2021. The work on load balancing in fog networks started with solutions with load balancing for small cell fog/cloud computing [41] in 2015. It should be noted that just 1 article was published in 2015, 4 articles in 2016, 3 articles in 2017, 21 articles in 2018, 2 articles in 2019, 11 articles in 2020, and 7 articles in 2021.

6.1.2 *Active Research Communities*

After selecting the studies, we looked at the authors' affiliations and identified 75 communities related to the research

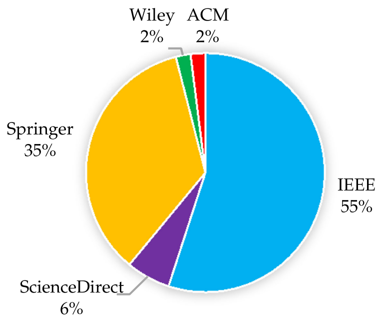

Fig. 4. The percentage of articles per publisher.

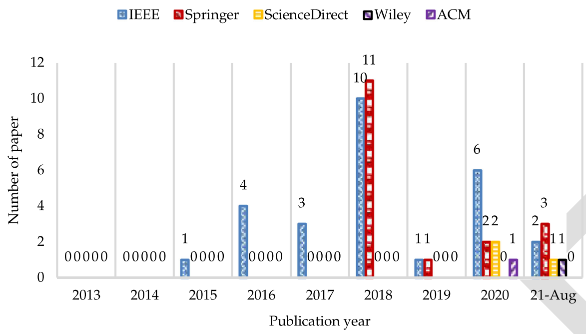

Fig. 5. The distribution of articles over time-based on the publisher.

TABLE 9
Active Communities on Load Balancing Algorithms in Fog Computing

Affiliation	Study ID	Research Focus
COMSATS University, Pakistan	[38], [39], [68], [69], [70] [82], [83], [84], [85], [76]	Approximate, Fundamental, and Hybrid algorithms
The University of Sydney, Australia	[53], [57]	Approximate and Exact algorithms
Sapienza University of Rome, Italy	[41], [61]	Approximate, Exact, Fundamental, and algorithms
Xidian University, China	[44], [47]	Approximate algorithms
Mansoura University, Egypt	[86], [87]	Hybrid algorithms
University of Modena and Reggio Emilia, Italy	[54], [79]	Approximate and Fundamental algorithms
National University of Sciences and Technology, Pakistan	[76], [81]	Fundamental algorithms
Kafrelsheikh University, Egypt	[86], [75]	Fundamental and Hybrid algorithms
Guangdong University of Petrochemical Technology, China	[67], [88]	Approximate and Exact algorithms
Indian Institute of Information Technology, India	[67], [77]	Exact and Fundamental algorithms

topic; however, as shown in Table 9, only active communities that presented at least two articles, along with their primary research focuses, are listed. A remarkable number of articles that are associated with approximate algorithms have been published in the COMSATS University in Pakistan, The University of Sydney in Australia, the Sapienza University of Rome in Italy, the Xidian University in China, the University of Modena and Reggio Emilia in Italy, and the Guangdong University of Petrochemical Technology in China. Further, The University of Sydney in

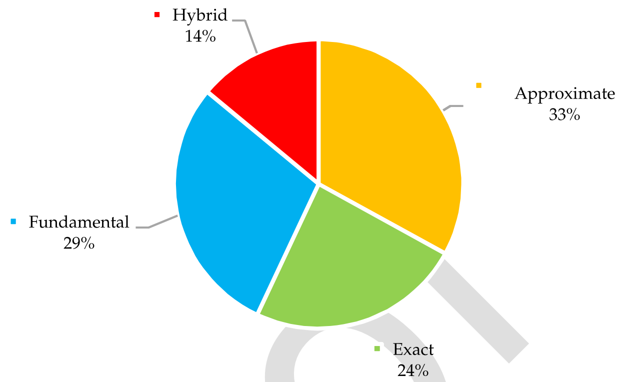

Fig. 6. The percentage of load balancing algorithms in fog computing.

Australia, the Sapienza University of Rome in Italy, the Guangdong University of Petrochemical Technology in China, and the Indian Institute of Information Technology in India studied exact algorithms. Besides, researchers in the COMSATS University in Pakistan, the University of Modena and Reggio Emilia in Italy, the National University of Sciences and Technology in Pakistan, the Kafrelsheikh University in Egypt, and the Indian Institute of Information Technology in India have focused on fundamental algorithms. Also, at the Mansoura University in Egypt, the COMSATS University in Pakistan, and Kafrelsheikh University in Egypt, researchers concentrated more on hybrid algorithms.

6.2 Research Objectives, Techniques, and Evaluation Parameters

The review process of the chosen articles in load balancing algorithms in fog computing was discussed in Section 5 in four main categories, including approximate, exact, fundamental, and hybrid algorithms. Moreover, now, the analytical and statistical reports of the research questions are presented based on the plan in Section 4.1.2 as follows:

- RQ 2: Which kind of classification in research approaches can be applied in fog systems load balancing?

The used algorithms in the load balancing of fog computing fall into four classes, and their statistic percentages are shown in Fig. 6. It was seen that the highest percentage of studies is conducted in approximate algorithms with 33%, while exact algorithms have 24%, fundamental algorithms have 29%, and hybrid algorithms have 14% of all types of used algorithms.

- RQ 3: What evaluation metrics are applied in load balancing algorithms of fog computing?

In order to answer RQ 3 and to consider the literature, researchers have used various evaluation metrics.

We applied (1) and illustrated factors and their percentages in Figs. 7 and 8. To obtain the percentage in each factor, we counted the number of a factor separately and divided it by the sum of the number of all factors, then multiplied the result by 100. Fig. 7 displays the percentage of evaluation metrics in load balancing algorithms of fog computing, with 31% for response time evaluation metrics, having the highest percentage, and energy with 18%, cost with 15%, processing time with 13%, resource utilization with 8%, scalability with 7%, security with 4%, reliability with 2%, and finally, throughput and availability each

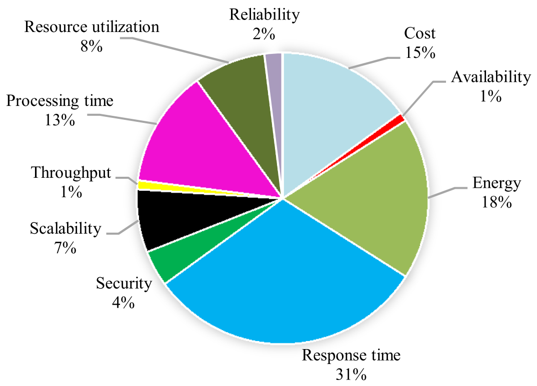

Fig. 7. The percentage of evaluation metrics in load balancing algorithms of fog computing.

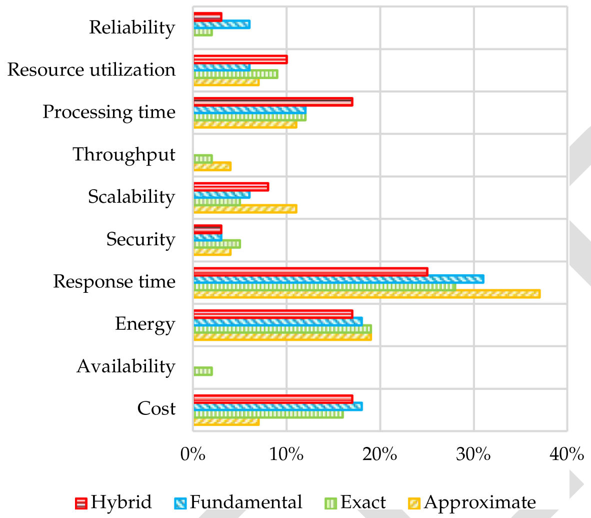

Fig. 8. The percentage of evaluation metrics in each classification.

having 1% of all evaluation metrics used in articles. As defined in Fig. 8, in approximate algorithms, 37% of the articles have made efforts to improve the response time, and 16% have improved the factor of energy. In exact algorithms, 28% of the studies have pushed for decreasing the response time, and 19% of the articles have tried to improve energy. In fundamental algorithms, 31% of studies have made attempts to improve response time, and 18% of the articles have improved two factors of cost and energy. Finally, in hybrid algorithms, 25% of the articles have made efforts to decrease the response time, and 17% have improved all cost factors, energy, scalability, and processing time.

$$Eval\_metric(i) = \frac{metric(i)}{\sum_{j=1}^{metric\_no} metric(j)} * 100 \quad (1)$$

• RQ 4: What popular evaluation tools are applied in load balancing algorithms in fog computing?

Fig. 9 defines the results provided by the comparisons in Tables 5, 6, 7, and 8, which illustrates the percentage of evaluation tools in the load balancing algorithms in fog computing. Also, statistically, Fig. 9 shows the percentage of evaluation tools used for the review of the literature here in this article. The CloudAnalyst and MATLAB have 16% each, CloudSim and NS-2/3 have 12% each, iFogSim, Java platform, JMeter, AMPL/CPLEX, Scyther, Mininet, work

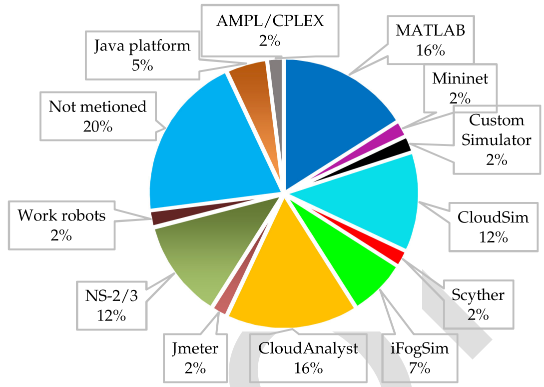

Fig. 9. The percentage of evaluation tools for load balancing algorithms in fog computing.

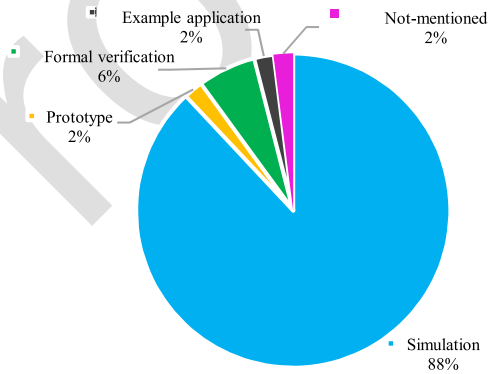

Fig. 10. The percentage of measurement environments in load balancing algorithms of fog computing.

robots come next, and custom simulator is the evaluation tools applied for these literature reviews.

• RQ 5: What measurement techniques are used to assess the load balancing in fog computing?

On that basis of Tables 5, 6, 7, and 8, Fig. 10 displays the percentage of measurement environments in the load balancing algorithms in fog computing. Simulation is the most used measurement environments with 88%, applied in four categories because actual environments have not yet been provided for fog computing; researchers mostly used simulation environments for measurement. Also, the formal verification used for measurement environments in some articles is in the exact and fundamental categories. Finally, the prototype and example application applied for measurement environments in some studies are in the approximate and fundamental categories.

7 Open Issues and Future Directions

Based on the review, there are some key cases not having been investigated in the load balancing of the fog network. So, in this section, the following answers to RQ 6 are presented.

• RQ 6: What are the open issues, future trends, and challenges of load balancing in fog computing?

- *Energy consumption and green fog*: In most of the techniques studied, the challenges of greenhouse gas and carbon emission were not considered by researchers. In the topic of load balancing and fog computing, green fog computing and energy consumption can be important [89]; however, only 18% of the reviewed articles focused on energy, which can optimize the popularity and efficacy of existing load balancing algorithms. So load balancing algorithms, based on energy consumption and greenhouse gas and carbon emissions in fog environments, are up-and-coming and can be a direction for open issues.

- *Multi-objective optimization*: It is obvious that there is no mechanism, in particular, to define most of the QoS parameters to decide for load balancing in a fog network. Some of the algorithms, for example, just regard energy, cost, or response time and ignore such parameters as scalability, reliability, and security. Therefore, multi-objective optimization in load balancing decision making needs to be expanded to bring some QoS parameters into consideration, and establishing a tradeoff between different parameters might be a considerable open issue.
- *Optimal solutions*: In literature, most fog-based approaches to load balancing, such as scheduling, and resource allocation, are in terms of complexity in the NP-hard and NP-complete groups. Some meta-heuristic and heuristic algorithms have been used to solve them, but other optimization algorithms such as bacterial colony optimization [90], memetic [91], [92], [93], Artificial Immune System [94], grey wolf Optimizer [95], simulated annealing [96], firefly algorithm [97], artificial bee colony [98], and lion optimizer algorithm [99] are good directions for future works.
- *Implementation challenges*: Regarding that fog computing is under investigation, the real testbed is not yet available to most researchers, and it was found that 88% of the articles used simulator-based tools for their assessments. Because the results of cases like scheduling in the real environment can be different from the simulation environment, it can be concluded that implementing the discussed algorithms in the real testbed is really challenging.
- *Context-aware computing*: As fog nodes and IoT devices can be mobile, it is important to predict the future location of nodes and devices to balance the load on them. Predicting the future location of nodes and devices can be provided by capturing mobility and behavioral patterns. With available contextual information and semantic support, load balancing solutions can be improved [100], [101]. Using context-aware computing to develop fog networks' load balancing techniques can be an exciting case for future trends.
- *SDN/NFV*: Considering the constraint of fog resources, the use of SDN in fog computing can provide load balancing more easily in fog network management. Applying network function virtualization (NFV) with fog computing can provide flexibility and speed in the management, construction, and deployment of novel applicant-based services. Using SDN/NFV [102] technologies for supporting load balancing techniques in fog computing is an interesting case to be studied.
- *Scalability*: Some approaches in fog computing must be able to act on large scales. The validation of these approaches on small scales does not guarantee some nodes, devices, and related operations. Despite its importance, only 7% of the literature has focused on the scalability factor. The related articles have been defined in small-scale scenarios, so it is an open challenge for future research.
- *Big data analytics*: The growing number of IoT devices in recent years has led to significant growth in big data. Recently, big data analytics has been used in decision-making and recommending systems for various smart environments, such as healthcare systems, vehicles, homes, and cities. Such criteria as latency, mobility, scalability, and localization are important in the big data analysis process in distributed environments, and fog computing can be used to meet such criteria. These metrics can be met by offloading computation and storing data to close fog nodes in the network [48], [103]. In addition, the use of load balancing algorithms can improve the QoS factors in big data analytics. Therefore, using the fog load balancing algorithms to perform big data analytics can be considered a future direction.
- *Interoperability*: Because of the variety and distribution of fog nodes and sources, interoperability can be regarded as a key success element in the load balancing in IoT/fog domains. As consumers would not like to use just one service provider, they often look for their favorites and some important factors like cost and functionality. Interoperability gives consumers the chance to move from one IoT/fog-based product to another or apply a combination of services and products to build smart environments, as they wish, in a customized way with load balancing [104]. So regarding interoperability as a key factor in combing the load balancing in IoT/fog-based services, it will be an interesting aspect for future studies.

8 Conclusion and Limitation

In this study, a qualitative and quantitative piece of research was conducted based on an SLR method on load balancing algorithms in fog computing. Applying 1054 studies published recently, between 2013 and 2021, the authors offered the SLR-based method in this literature by using the exploration query. In addition, 49 studies focusing on load balancing algorithms in fog computing were examined. According to RQ 2, the used algorithms in the load balancing of fog computing were categorized into four groups, with the highest percentage of research done in approximate algorithms with 33%, fundamental algorithms with 29%, exact algorithms with 24%, and hybrid algorithms with 14% of all types of used algorithms. According to RQ 3, statistically, the percentage of evaluation metrics presented that the response time metric has the highest application in the assessment of the load balancing algorithms by 31%, and energy comes next with 18%. Based on RQ 4, for evaluation tools, it was observed that CloudAnalyst and MATLAB have the highest percentage of applying the simulation environment of case studies in load balancing algorithms, with 16% each. Also, with respect to RQ 5, it was observed that 88% of studies applied a simulation environment for measuring the suggested load balancing algorithms in fog computing because actual environments have not yet been provided for fog computing. Finally, based on RQ 6, the important open challenges and future directions of load balancing algorithms in fog computing were described in detail.

Because of systematic review-based methods, all existing studies might not have been evaluated. These types of review articles usually have limitations [5], but the systematic review results are mainly reliable [105]. The main limitations of this systematic review may include the following:

- Although several well-known databases, such as ScienceDirect, IEEE, ACM, Springer, Wiley, and Taylor & Francis, were used as reliable sources, it is impossible to guarantee that all the relevant studies are selected. So, some articles were neglected due to the mechanism described in Section 4.2.
- We did not consider non-peer-reviewed and non-English articles, chapters of books, thesis, review articles, short articles, and editorial papers.
- This article was divided into four groups: approximate, exact, fundamental, and hybrid. However, other taxonomies could be possible.
- We presented six research questions to find their answers by investigating related papers. Other scholars may define some other research questions.

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. 1st ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[2] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, 2019.

[3] N. Auluck, A. Azim, and K. Fizza, "Improving the schedulability of real-time tasks using fog computing," *IEEE Trans. Serv. Comput.*, vol. 15, no. 1, pp. 372–385, Jan./Feb. 2022.

[4] S. Aslam and M. A. Shah, "Load balancing algorithms in cloud computing: A survey of modern techniques," in *Proc. Nat. Softw. Eng. Conf.*, 2015, pp. 30–35.

[5] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007.

[6] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 142–157, Jul.–Dec. 2013.

[7] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.

[8] C. Jatoth, G. R. Gangadharan, and R. Buyya, "Computational intelligence based qos-aware web service composition: A systematic literature review," *IEEE Trans. Serv. Comput.*, vol. 10, no. 3, pp. 475–492, May/Jun. 2017.

[9] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: A proposal and a discussion," *Requirements Eng.*, vol. 11, no. 1, pp. 102–107, 2005.

[10] M. Haghi Kashani, M. Madanipour, M. Nikravan, P. Asghari, and E. Mahdipour, "A systematic review of IoT in healthcare: Applications, techniques, and trends," *J. Netw. Comput. Appl.*, vol. 192, 2021, Art. no. 103164.

[11] M. Haghi Kashani, A. M. Rahmani, and N. Jafari Navimipour, "Quality of service-aware approaches in fog computing," *Int. J. Commun. Syst.*, vol. 33, no. 8, pp. 1–34, 2020.

[12] M. Rahimi, M. Songhorabadi, and M. H. Kashani, "Fog-based smart homes: A systematic review," *J. Netw. Comput. Appl.*, vol. 153, 2020, Art. no. 102531.

[13] Z. Ahmadi, M. Haghi Kashani, M. Nikravan, and E. Mahdipour, "Fog-based healthcare systems: A systematic review," *Multimedia Tools Appl.*, vol. 80, pp. 36 361–36 400, 2021.

[14] S. Bazzaz Abkenar, M. Haghi Kashani, M. Akbari, and E. Mahdipour, "Twitter spam detection: A systematic review," 2020, *arXiv:2011.14754*.

[15] M. Fathi, M. Haghi Kashani, S. M. Jameii, and E. Mahdipour, "Big data analytics in weather forecasting: A systematic review," *Arch. Comput. Methods Eng.*, vol. 29, pp. 1247–1275, 2022.

[16] M. Songhorabadi, M. Rahimi, A. M. M. Farid, and M. H. Kashani, "Fog computing approaches in smart cities: A state-of-the-art review," 2020, *arXiv:2011.14732*.

[17] M. Sheikh Sofla, M. Haghi Kashani, E. Mahdipour, and R. Faghih Mirzaee, "Towards effective offloading mechanisms in fog computing," *Multimedia Tools Appl.*, vol. 81, pp. 1997–2042, 2021.

[18] Y. Karimi, M. Haghi Kashani, M. Akbari, and E. Mahdipour, "Leveraging big data in smart cities: A systematic review," *Concurrency Computation, Pract. Experience*, vol. 33, 2021, Art. no. e6379.

[19] M. Sharma and L. Ahuja, "A data mining approach towards healthcare recommender system," in *Proc. Int. Conf. Next Gener. Comput. Technol.*, 2017, pp. 199–210.

[20] M. Maadani *et al.*, "EDCA delay analysis of spatial multiplexing in IEEE 802.11-based wireless sensor and actuator networks," *Int. J. Inf. Electron. Eng.*, vol. 2, no. 3, pp. 318–322, 2012.

[21] E. Marín-Tordera *et al.*, "Do we all really know what a fog node is? Current trends towards an open definition," *Comput. Commun.*, vol. 109, pp. 117–130, 2017.

[22] O. C. A. W. Group, "OpenFog reference architecture for fog computing," *OPFRA001*, vol. 20817, 2017, Art. no. 162.

[23] C. Mouradian *et al.*, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surv. Tut.*, vol. 20, no. 1, pp. 416–464, Jan.–Mar. 2017.

[24] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Commun. Surv. Tut.*, vol. 21, no. 1, pp. 260–288, Jan.–Mar. 2019.

[25] M. Iorga *et al.*, *Fog Computing Conceptual Model, Recommendations of the National Institute of Standards and Technology*. Gaithersburg, MD, USA: NIST, 2018, pp. 500–325.

[26] A. S. Milani and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," *J. Netw. Comput. Appl.*, vol. 71, pp. 86–98, 2016.

[27] M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," *Concurrency Computation, Pract. Experience*, vol. 29, no. 12, 2017, Art. no. e4123.

[28] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load balancing mechanisms in the software defined networks: A systematic and comprehensive review of the literature," *IEEE Access*, vol. 6, pp. 14 159–14 178, 2018.

[29] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 50–71, 2017.

[30] A. Chandak and N. K. Ray, "A review of load balancing in fog computing," in *Proc. Int. Conf. Inf. Technol.*, 2019, pp. 460–465.

[31] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, "Survey on service migration, load optimization and load balancing in fog computing environment," in *Proc. IEEE 5th Int. Conf. Convergence Technol.*, 2019, pp. 1–5.

[32] H. Pydi and G. N. Iyer, "Analytical review and study on load balancing in edge computing platform," in *Proc. 4th Int. Conf. Comput. Methodol. Commun.*, 2020, pp. 180–187.

[33] S. P. Singh, R. Kumar, A. Sharma, and A. Nayyar, "Leveraging energy-efficient load balancing algorithms in fog computing," *Concurrency Computation, Pract. Exp.*, vol. 34, no. 13, 2020 Art. no. e5913.

[34] M. Kaur and R. Aron, "A systematic study of load balancing approaches in the fog computing environment," *J. Supercomputing*, vol. 77, no. 8, pp. 9202–9247, 2021.

[35] B. Kitchenham, "Procedures for Performing Systematic Reviews," Keele Univ., Keele, U.K., Tech. Rep. 33, 2004, pp. 1–26.

[36] S. Bazzaz Abkenar, M. Haghi Kashani, E. Mahdipour, and S. M. Jameii, "Big data analytics meets social media: A systematic review of techniques, open issues, and future directions," *Telematics Inform.*, vol. 57, pp. 10 1517–11 0555, 2020.

[37] X.-S. Yang, "Harmony search as a metaheuristic algorithm," in *Music-Inspired Harmony Search Algorithm*, Berlin, Germany: Springer, 2009, pp. 1–14.

[38] M. Zahid *et al.*, "Hill climbing load balancing algorithm on fog computing," in *Proc. Int. Conf. P2P Parallel Grid Cloud Internet Comput.*, 2018, pp. 238–251.

[39] M. B. Kamal *et al.*, "Heuristic min-conflicts optimizing technique for load balancing on fog computing," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, 2018, pp. 207–219.

[40] F. Banaie, M. H. Yaghmaee, S. A. Hosseini, and F. Tashtarian, "Load-balancing algorithm for multiple gateways in fog-based internet of things," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7043–7053, Aug. 2020.

[41] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. IEEE 81st Veh. Technol. Conf.*, 2015, pp. 1–6.

[42] X. Xu *et al.*, "A heuristic virtual machine scheduling method for load balancing in fog-cloud computing," in *Proc. IEEE 4th Int. Conf. Big Data Secur. Cloud*, 2018, pp. 83–88.

[43] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm Evol. Computation*, vol. 16, pp. 1–18, 2014.

[44] X. He, Z. Ren, C. Shi, and J. Fang, "A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles," *China Commun.*, vol. 13, no. Supplement2, pp. 140–149, 2016.

[45] J. Wan *et al.*, "Fog computing for energy-aware load balancing and scheduling in smart factory," *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4548–4556, Oct. 2018.

[46] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, "Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method," *Appl. Nanosci.*, 2021.

[47] C. Shi, Z. Ren, and X. He, "Research on load balancing for software defined cloud-fog network in real-time mobile face recognition," in *Proc. Int. Conf. Commun. Netw. China*, 2016, pp. 121–131.

[48] J. Yang, "Low-latency cloud-fog network architecture and its load balancing strategy for medical big data," *J. Ambient Intell. Humanized Comput.*, 2020.

[49] S. S. Karthik and A. Kavithamani, "Fog computing-based deep learning model for optimization of microgrid-connected WSN with load balancing," *Wireless Netw.*, vol. 27, no. 4, pp. 2719–2727, 2021.

[50] R. Qun and S. M. Arefzadeh, "A new energy-aware method for load balance managing in the fog-based vehicular ad hoc networks (VANET) using a hybrid optimization algorithm," *IET Commun.*, vol. 15, no. 13, pp. 1665–1676, 2021.

[51] C. Li, H. Zhuang, Q. Wang, and X. Zhou, "SSLB: Self-similarity-based load balancing for large-scale fog computing," *Arabian J. Sci. Eng.*, vol. 43, pp. 7487–7498, 2018.

[52] S. P. Singh, A. Sharma, and R. Kumar, "Design and exploration of load balancers for fog computing using fuzzy logic," *Simul. Modell. Pract. Theory*, vol. 101, 2020, Art. no. 102017.

[53] S. F. Abedin, A. K. Bairagi, M. S. Munir, N. H. Tran, and C. S. Hong, "Fog load balancing for massive machine type communications: A game and transport theoretic approach," *IEEE Access*, vol. 7, pp. 4204–4218, 2019.

[54] R. Beraldi *et al.*, "Randomized load balancing under loosely correlated state information in fog computing," in *Proc. 23rd Int. ACM Conf. Model. Anal. Simul. Wireless Mobile Syst.*, 2020, pp. 123–127.

[55] G. J. Woeginger, "Exact algorithms for NP-hard problems: A survey," in *Combinatorial Optimization—Eureka, You Shrink!*, Berlin, Germany: Springer, 2003, pp. 185–207.

[56] S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Commun.*, vol. 13, no. 3, pp. 156–164, 2016.

[57] D. Puthal *et al.*, "Secure and sustainable load balancing of edge data centers in fog computing," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 60–65, May 2018.

[58] K. Cui, W. Sun, B. Lin, and W. Sun, "Load balancing mechanisms of unmanned surface vehicle cluster based on marine vehicular fog computing," in *Proc. 16th Int. Conf. Mobility Sens. Netw.*, 2020, pp. 797–802.

[59] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered ioT," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 253–262, Jan.–Mar. 2018.

[60] E. Barros, M. Peixoto, D. Leite, B. Batista, and B. Kuehne, "A fog model for dynamic load flow analysis in smart grids," in *Proc. IEEE Symp. Comput. Commun.*, 2018, pp. 1–6.

[61] R. Beraldi and H. Alnuweiri, "Sequential randomization load balancing for fog computing," in *Proc. 26th Int. Conf. Softw. Telecommun. Comput. Netw.*, 2018, pp. 1–6.

[62] D. Chen and V. Kuehn, "Adaptive radio unit selection and load balancing in the downlink of fog radio access network," in *Proc. IEEE Glob. Commun. Conf.*, 2016, pp. 1–7.

[63] M. M. S. Maswood, M. R. Rahman, A. G. Alharbi, and D. Medhi, "A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment," *IEEE Access*, vol. 8, pp. 11 3737–11 3750, 2020.

[64] S. Sthapit, J. Thompson, N. M. Robertson, and J. Hopgood, "Computational load balancing on the edge in absence of cloud and fog," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1499–1512, Jul. 2019.

[65] Y.-A. Chen, J. P. Walters, and S. P. Crago, "Load balancing for minimizing deadline misses and total runtime for connected car systems in fog computing," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl.*, 2017, pp. 683–690.

[66] N.-N. Dao *et al.*, "Adaptive resource balancing for serviceability maximization in fog radio access networks," *IEEE Access*, vol. 5, pp. 14 548–14 559, 2017.

[67] M. Mukherjee, Y. Liu, J. Lloret, L. Guo, R. Matam, and M. Aazam, "Transmission and latency-aware load balancing for fog radio access networks," in *Proc. IEEE Glob. Commun. Conf.*, 2018, pp. 1–6.

[68] T. Nazar *et al.*, "Modified shortest job first for load balancing in cloud-fog computing," in *Proc. Int. Conf. Broadband Wireless Comput. Commun. Appl.*, 2018, pp. 63–76.

[69] N. Ahmad *et al.*, "Fog-cloud based platform for utilization of resources using load balancing technique," in *Proc. Int. Conf. Netw.-Based Inf. Syst.*, 2018, pp. 554–567.

[70] S. Tariq *et al.*, "Priority based load balancing in cloud and fog based systems," in *Proc. Int. Conf. Broadband Wireless Comput. Commun. Appl.*, 2018, pp. 725–736.

[71] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Queuing model for evs energy management: Load balancing algorithms based on decentralized fog architecture," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[72] E. C. P. Neto, G. Callou, and F. Aires, "An algorithm to optimise the load distribution of fog environments," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2017, pp. 1292–1297.

[73] E. Batista *et al.*, "Load balancing in the fog of things platforms through software-defined networking," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.*, 2018, pp. pp. 1785–1791.

[74] S. Verma *et al.*, "An efficient data replication and load balancing technique for fog computing environment," in *Proc. 3rd Int. Conf. Comput. Sustain. Glob. Develop.*, 2016, pp. 2888–2895.

[75] F. Alqahtani, M. Amoon, and A. A. Nasr, "Reliable scheduling and load balancing for requests in cloud-fog computing," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 4, pp. 1905–1916, 2021.

[76] A. Asghar, A. Abbas, H. A. Khattak, and S. U. Khan, "Fog based architecture and load balancing methodology for health monitoring systems," *IEEE Access*, vol. 9, pp. 96 189–96 200, 2021.

[77] N. Mazumdar, A. Nag, and J. P. Singh, "Trust-based load-offloading protocol to reduce service delays in fog-computing-empowered ioT," *Comput. Elect. Eng.*, vol. 93, 2021, Art. no. 107223.

[78] E. P. Pereira, E. L. Padoin, R. D. Medina, and J. F. Méhaut, "Increasing the efficiency of fog nodes through of Priority-based load balancing," in *Proc. IEEE Symp. Comput. Commun.*, 2020, pp. 1–6.

[79] R. Beraldi, C. Canali, R. Lancellotti, and G. P. Mattia, "Distributed load balancing for heterogeneous fog computing infrastructures in smart cities," *Pervasive Mobile Comput.*, vol. 67, 2020, Art. no. 101221.

[80] A. U. Rehman *et al.*, "Dynamic energy efficient resource allocation strategy for load balancing in fog environment," *IEEE Access*, vol. 8, pp. 19 9829–19 9839, 2020.

[81] Z. Sharmin, A. W. Malik, A. U. Rahman, and R. M. Noor, "Toward sustainable micro-level fog-federated load sharing in internet of vehicles," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3614–3622, 2020.

[82] S. A. A. Naqvi *et al.*, "Metaheuristic optimization technique for load balancing in cloud-fog environment integrated with smart grid," in *Proc. Int. Conf. Netw.-Based Inf. Syst.*, 2018, pp. 700–711.

[83] S. H. Abbasi *et al.*, "Load stabilizing in fog computing environment using load balancing algorithm," in *Proc. Int. Conf. Broadband Wireless Comput. Commun. Appl.*, 2018, pp. 737–750.

[84] M. J. Ali *et al.*, "State based load balancing algorithm for smart grid energy management in fog computing," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, 2018, pp. 220–232.

[85] M. Zubair *et al.*, "Integration of cloud-fog based platform for load balancing using hybrid genetic algorithm using bin packing technique," in *Proc. Int. Conf. P2P Parallel Grid Cloud Internet Comput.*, 2018, pp. 279–292.

[86] F. M. Talaat *et al.*, "A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 4951–4966, 2020.

[87] F. M. Talaat, S. H. Ali, A. I. Saleh, and H. A. Ali, "Effective load balancing strategy (ELBS) for real-time fog computing environment using fuzzy and probabilistic neural networks," *J. Netw. Syst. Manage.*, vol. 27, pp. 1–47, 2019.

[88] J. Yan, J. Wu, Y. Wu, L. Chen, and S. Liu, "Task offloading algorithms for novel load balancing in homogeneous fog network," in *Proc. IEEE 24th Int. Conf. Comput. Supported Cooperative Work Des.*, 2021, pp. 79–84.

[89] S. Conti, G. Faraci, R. Nicolosi, S. A. Rizzo, and G. Schembra, "Battery management in a green fog-computing node: A reinforcement-learning approach," *IEEE Access*, vol. 5, pp. 21 126–21 138, 2017.

[90] B. Niu and H. Wang, "Bacterial colony optimization," *Discrete Dyn. Nat. Soc.*, vol. 2012, pp. 1–28, 2012.

[91] M. Kashani and M. Jahanshahi, "A new method based on memetic algorithm for task scheduling in distributed systems," *Int J Simul. Syst. Sci. Technol.*, vol. 10, pp. 26–32, 2009.

[92] K. Mostafa Haghi, S. Raheleh, and J. Mahdi, *A New Distributed Systems Scheduling Algorithm: A Swarm Intelligence Approach*. Bellingham, WA, USA: SPIE, 2012.

[93] M. H. Kashani, H. Zarrabi, and G. Javadzadeh, "A new metaheuristic approach to task assignment problem in distributed systems," in *Proc. IEEE 4th Int. Conf. Knowl.-Based Eng. Innov.*, 2017, pp. 0673–0677.

[94] L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Berlin, Germany: Springer, 2002.

[95] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[96] M. H. Kashani and M. Jahanshahi, "Using simulated annealing for task scheduling in distributed systems," in *Proc. Int. Conf. Comput. Intell. Modell. Simul.*, 2009, pp. 265–269.

[97] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," 2010, *arXiv:1003.1409*.

[98] R. Sarvizadeh, M. H. Kashani, F. S. Zakeri, and S. M. Jameii, "A novel bee colony approach to distributed systems scheduling," *Int. J. Comput. Appl.*, vol. 42, no. 10, pp. 1–6, 2012.

[99] M. Yazdani and F. Jolai, "Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm," *J. Comput. Des. Eng.*, vol. 3, no. 1, pp. 24–36, 2016.

[100] B. Gu, X. Wang, Y. Qu, J. Jin, Y. Xiang, and L. Gao, "Context-aware privacy preservation in a hierarchical fog computing system," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.

[101] Q. T. Minh, E. Kamioka, and S. Yamada, "CFC-ITS: Context-aware fog computing for intelligent transportation systems," *IT Professional*, vol. 20, no. 6, pp. 35–45, 2018.

[102] R. Bruschi *et al.*, "An SDN/NFV platform for personal cloud services," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 1143–1156, Apr. 2017.

[103] W. Zhang, Z. Zhang, and H. Chao, "Cooperative fog computing for dealing with Big Data in the Internet of Vehicles: Architecture and hierarchical resource management," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 60–67, Dec. 2017.

[104] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in iot-enabled healthcare solutions," in *Proc. 19th Int. Conf. Distrib. Comput. Netw.*, 2018, pp. 1–10.

[105] H. Zhang and M. A. Babar, "Systematic reviews in software engineering: An empirical investigation," *Inf. Softw. Technol.*, vol. 55, no. 7, pp. 1341–1354, 2013.

Mostafa Haghi Kashani (Member, IEEE) received the BSc degree in computer engineering from the Kashan Branch of IAU, Iran, in 1999, and the MSc degree in computer engineering from the South Tehran Branch of IAU, Iran, in 2002. He is currently working toward the PhD degree in computer engineering-software systems with IAU Science and Research Branch, Tehran, Iran. He is a researcher and lecturer with the Department of Computer Engineering, IAU University. His research interests include fog/edge computing, IoT, evolutionary computation, and smart environments. He serves as an editor for Frontiers in Big Data and Frontiers in Artificial Intelligence. He has acted as a reviewer in several international journals, including the *IEEE Transactions on Industrial Informatics*, *IEEE Internet of Things*, *Sustainable Computing: Informatics and Systems* (Elsevier), the Journal of Supercomputing (Springer), Multimedia Tools and Applications (Springer), the International Journal of Communication Systems (Wiley), and other top journals in the field.

Ebrahim Mahdipour received the BS degree in computer engineering, specialized in hardware engineering from Dezful Branch, Islamic Azad University, Dezful, Iran, in 2003, the MS degree in computer engineering, specialized in computer architecture, and the PhD degree in computer engineering, specialized in computer architecture, from Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2006 and 2012, respectively. He is the founding director of Cyber Security. He is currently an assistance professor with the Department of Computer Engineering, Science and Research Branch, Islamic Azad University. His research interests include cyber security, blockchain, and Big Data.