Ali and Alubady *Iraqi Journal of Science, 2024, Vol. 65, No. 7, pp: 4008-4029*
DOI: 10.24996/ijs.2024.65.7.37

ISSN: 0067-2904

A Survey of Fog Computing-Based Resource Allocation Approaches: Overview, Classification, and Opportunity for Research

Samah Ali*, Raaid Alubady
Department of Information Network, College of Information Technology, University of Babylon, Babylon, Iraq

Received: 9/12/2022 Accepted: 24/6/2023 Published: 30/7/2024

Abstract
Recently, cloud computing has affected a large part of the computer industry, including software companies and internet service providers. It has proven efficient in managing tasks for applications. Despite its popularity, cloud computing does not meet the requirements of applications because it faces many limitations, such as high latency and bandwidth bottlenecks. These limitations will significantly affect applications sensitive to delays. To meet this challenge, fog computing is introduced as an extension to cloud computing. It improves quality of service (QoS) for applications that suffer from latency by keeping resources and services close to the end-user. How to efficiently and fairly allocate the available resources, e.g., CPU, bandwidth, and memory, between different requested tasks is a complex challenge. The main goal of this paper is to study the concepts of fog computing, architecture, environment, and metrics that affect resource allocation in fog computing. It also summarizes the classification of modern resource allocation approaches based on QoS metrics (2017–2023). On the other hand, highlighting the pros and cons of these studies as well as future research directions to develop different approaches.

Keywords: Cloud Computing, Fog Computing, Resource Allocation, Fog Environments, Latency, Quality of service.

مسح لنهج تخصيص الموارد القائمة على الحوسبة الضبابية: نظرة عامة، تصنيفها والفرصة البحثية

سماح علي*, رائد العبيدي
قسم شبكة المعلومات، كلية تكنولوجيا المعلومات، جامعة بابل، بابل، العراق

الخلاصة
في الآونة الأخيرة ، أثرت الحوسبة السحابية على جزء كبير من صناعة الحاسوب، بما في ذلك شركات البرمجيات ومقدمي خدمات الإنترنت حيث أثبتت كفاءتها في إدارة مهام التطبيقات. على الرغم من شعبيتها، إلا أن الحوسبة السحابية لا تفي بجميع بمتطلبات التطبيقات لأنها تواجه العديد من القيود، مثل طول وقت الاستجابة والاختناقات في شبكات النطاق العريض. ستؤثر هذه القيود بشكل كبير على التطبيقات الحساسة للتأخير. لمواجهة هذا التحدي، تقدم حوسبة الضباب كامتداد للحوسبة السحابية. تعمل على تحسين جودة الخدمة (QoS) للتطبيقات التي تعاني من زمن انتقال من خلال إبقاء الموارد والخدمات قريبة من المستخدم النهائي. كيفية تخصيص الموارد المتاحة ، على سبيل المثال ، وحدة المعالجة المركزية وعرض النطاق الترددي والذاكرة بين المهام المطلوبة المختلفة بكفاءة وعادلة هو تحد معقد. الهدف الرئيسي من هذه الورقة هو دراسة مفاهيم الحوسبة الضبابية ، والهندسة المعمارية ، والبيئة ، والمقاييس التي تؤثر على تخصيص الموارد في حوسبة الضباب. كما

*Email: asamah@uobabylon.edu.iq

يلخص تصنيف مناهج تخصيص الموارد الحديثة بناءً على مقاييس جودة الخدمة (2017–2023). من ناحية أخرى ، تسليط الضوء على إيجابيات وسلبيات هذه الدراسات ، وكذلك اتجاه البحث المستقبلي لتطوير مناهج مختلفة.

1. Introduction

With the development of the world and the development of different devices and technologies, this all led to an increasing amount of data that was generated via different applications and required storage and processing. On the other hand, the number of connected Internet of Things (IoT) devices will reach 41 billion by 2025, according to estimates by the International Data Corporation and generators, more than 79 zettabytes [1]. To keep pace with this growth of applications, cloud computing has been introduced to deal with the requirements of these applications due to the scalability and flexibility of the services provided to end users [2]. The cloud computing model is efficient, but at the same time, it has disadvantages such as high latency because any requested data must be sent to a centralized data center, such as in applications for smart healthcare and smart grids, as well as in applications that are sensitive to high latency or high response times [3]. Thus, these applications need specific resources included in the environment to ensure better quality of service (QoS). With the development of the world and the development of different devices and technologies, this all led to an increasing amount of data that was generated via different applications and required storage and processing. On the other hand, the number of connected Internet of Things (IoT) devices will reach 41 billion by 2025, according to estimates by the International Data Corporation and generators, more than 79 zettabytes [1]. To keep pace with this growth of applications, cloud computing has been introduced to deal with the requirements of these applications due to the scalability and flexibility of the services provided to end users [2]. The cloud computing model is efficient, but at the same time, it has disadvantages such as high latency because any requested data must be sent to a centralized data center, such as in applications for smart healthcare and smart grids, as well as in applications that are sensitive to high latency or high response times [3]. Thus, these applications need specific resources included in the environment to ensure better quality of service (QoS).

Resource allocation in any environment represents an essential feature of getting economic benefits. Therefore, the resource allocation mechanism must ensure the fair and efficient distribution of resources among all devices to provide better QoS. An efficient resource allocation strategy can promote proper utilization of all resources with a suitable response time, enhance mobility, and minimize bandwidth [4]. It can be challenging to assign optimal resources from the available resources, such as CPU, memory, bandwidth, etc., to the end user fairly and efficiently [5]. Each device considers its bandwidth, RAM, and processing capacity [6]. To overcome these issues faced by cloud computing, Cisco introduced fog computing in 2012 [7]. Fog computing proposes an extension and solution to cloud computing limitations and supports many benefits such as sensitivity, better QoS, and being geographically distributed [8]. It will reduce the amount of data that transfers to the cloud because any request will be sent to fog nodes at the fog layer. These nodes are close to the end-user on the edge network to make processing, storage, and computation operations [9]. On the other hand, fog computing does not replace cloud computing but introduces it to minimize the disadvantages of cloud computing and enhance the services at the edge of the Internet [7].

Although numerous review papers study resource allocation in the fog computing environment, such as [6], [10–12], more deeply investigated research is required in this field. This can be done by reviewing modern, efficient approaches to resource allocation that can find an optimal way to distribute available resources between tasks and consumers. This study seeks to study, analyze, and classify the innovative approaches developed to address many issues based on QoS metrics. Further, it highlights several related issues in resource allocation over fog computing.

Accordingly, this investigation is structured as follows: Section 2 provides a high-level introduction to fog computing. Section 3 examines how quality of service measurements might be used to categorize the various strategies for allocating resources in fog computing. The concept of "fog computing" is laid forth in Section 4. Section 5 presents the other literature surveys in this field. The study concludes with Sections 6 and 7, which focus on the central problem of resource allocation in fog computing and present the paper's findings.

2. Overview and Background

In this section, an overview of fog computing will be displayed, along with its definition, architecture, and basic differences between cloud computing and fog computing.

2.1. Fog Computing

Fog computing is a decentralized model where requests for data do not need to be sent to the cloud. Its processing, storage, and computation operations can be done at the fog layer. Thus, it does not need to be a third party. Fog nodes appear hierarchically between the end-device layer and the cloud layer [1]. Fog cannot function independently since it is closely related to the presence of a cloud. This has caused the interactions between the fog and the cloud to get extra attention [10]. Figure 1 shows the schema of the fog computing model.

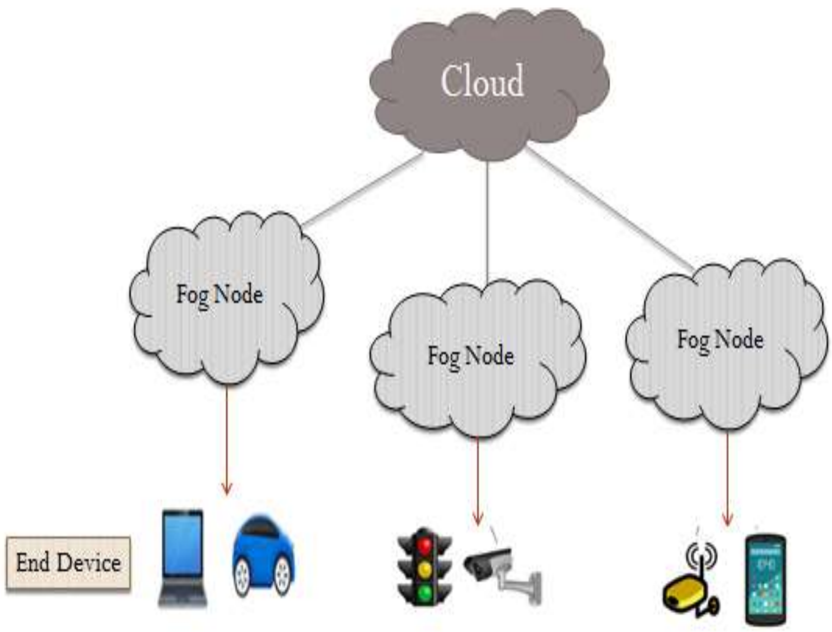

Figure 1: Fog Computing Model

2.2. Fog Architecture

In this part, the architecture of fog computing will be explained. According to Hu et al. [11], the fog computing architecture comprises three main layers, as illustrated in Figure 2.

Figure 2: Architecture of Fog Computing [12].

- Layer 1 (End Device): This is the basic layer that consists of devices such as smartphones, sensors, smart vehicles, smart watches, etc. These end devices are often called terminal nodes (TN), and they are supposed to be equipped with the Global Positioning System. These devices can operate in a heterogeneous environment with different technologies.
- Layer 2 (fog computing layer): It is the intermediate layer that connects with the cloud layer from the upper and from the bottom of the end device layer. The layer consists of many fog nodes and servers, such as routers, switches, and access points. These nodes have capabilities to share, store, and compute. In fog computing, the end device establishes a connection to the fog computing layer to obtain various services from the nodes, which may be physically located in one place or move about with the carrier. In addition, fog computing links to the cloud via the IP core network to gain access to even more robust processing and data storage resources.
- Layer 3 (cloud layer): Data centers reside on the cloud layer. This layer consists of many powerful storage devices and servers. It provides high performance and capabilities for computation, analysis, and permanent storage of massive data.

In this architecture, any device (smart things) will be connected to either fog node through wireless technologies such as WiFi, Bluetooth, 3G, 4G, etc. Each node's role in providing services depends on its position in the architecture. This architecture becomes more appropriate for IoT applications because it is very close to the end device [13].

2.3. Fog Computing vs Cloud Computing

A main difference between the cloud computing model and the fog computing model is presented in Table 1.

Table 1: Compression between Fog and Cloud Computing

Features	Cloud computing	Fog computing
Geographical distribution	Centralized	Decentralized
Latency	High	low
Energy consumption (special the energy consumption of data center coolant system)	High	low
Security	Less secure	More secure
Reliability	High	low
Deployment cost	High	low
Management	Easy	Hard
Distance to end device	Far (multiple hop)	Near (single hop)
Mobility	limited	supported
Number of nodes	Thousands	Millions
Access	Predominantly wireless	Wire and wireless
Location awareness	Partially supported	Fully supported
Computation and storage capabilities	Strong	Weak
Scalability	Less scalable	More scalable
Power source	Buttery/direct power	Direct power

2.4. Resource Allocation in Fog Computing

Each application connected to fog nodes needs many resources, such as memory, CPU, and networking [14]. To control these resources efficiently and fairly, it must use optimal resource management approaches that consider the requirements of QoS [6]. Resource Allocation (RA) aims to optimally and efficiently assign resources to tasks or requests that come from end devices [15]. The main goal of RA approaches is to provide an optimal allocation for applications, services, and other activities to minimize or maximize objectives related to the RA concepts [16]. On the other hand, many reasons make resource management one of the challenges in fog areas due to the fog environment being unpredictable, highly variable, heterogeneous, resource-constrained, having a large number of requests that need to be completed, and having an unpredictable arrival rate [17] [15].

3. Fog Environment and Performance Metrics

3.1. Fog Environment

In fog computing, many popular environments have been employed to simulate resource allocation approaches. The main environments, including iFogSim, MobFogSim, EmuFog, FogNetSim++, and YAFS, will be described in this section.

- **iFogSim**

It is one of the most common simulators to model and analyze the fog environment and is also used to estimate the impact of resource management approaches such as latency, energy consumption, and cost. The sense-process-actuate model is the major application model for iFogSim. The basic functions in CloudSim are used to implement functionalities in iFogSim. To handle events between fog environment components, the core is responsible for this handling in iFogSim [18]. iFogSim is employed to evaluate many works in IoT systems but lacks precise modeling of computation scheduling in processing elements [19].

- **MobFogSim**

MobFogSim is an add-on to iFogSim that facilitates cloud-to-cloud migration and mobility in fog computing. The objectives of this simulator are the evaluation of application, behavior, and performance. The infrastructure included sensors, actuators, devices, and a data center. MobFogSim implements three distinct migration algorithms dependent on user location-aware matrices and location [20].

- **EmuFog**

EmuFog designs fog computing scenarios. It is an extensible emulation framework. It enables researchers to design a fog computing infrastructure from scratch and design network topologies according to the use case. Although EmuFog provides a default solution for each of its sub-modules, each of them is easily extendable [21].

- **FogNetSim++**

This simulator is introduced as an extension to CloudNetSim++, which uses the available properties in OMNet++. The main goal of the development of this simulator is that the existing framework be designed to support many sensors. All the available modules in oMNet++ can be easily integrated into FogSim++. The infrastructure of this simulator is the mobile device, fog nodes, broker nodes, sensors, base stations, and geographic data centers. It consists of two modules: the end device and the broker [22].

- **YAFS**

Another Fog Simulator refers to YAFS, a simulation library employed to simulate fog, cloud, and edge scenarios. This simulator enables analyses related to resource allocation, billing management, network design, placement, scheduling, and routing. This simulator is set to reduce the number of classes to seven, which makes the learning curve quite low compared to other simulators [23].

Table 2 presents the main points based on programming language, topology structure, and topology definition for each environment.

Table 2: Main Points for each Environment

Simulator	Programing language	Topology structure	Topology definition	Open Source
iFogSim	Java	Tree	API	Yes
MobFogSim	Java	Graph	API, formats-Graph	No
EmuFog	Java	Graph	API, formats-Graph	Yes
FogNetSim++	C++	Graph	API	Yes
YAFS	Python	Graph	API, JSON Graph-format	Yes

3.2. Performance Evaluation Metrics

The performance evaluation is critical to checking the completed results of any study or research. Therefore, choosing the right metrics is essential to differentiate in all performance evaluations. In this study, approaches to resource allocation inside the fog computing environment were selected and analyzed. Hence, the most popular metrics for evaluating these approaches were identified according to relevant works. This section presents the main popular metrics employed to evaluate the proposed resource allocation approaches in fog computing.

- **Response time**

Response time (execution time) is called completion time, which represents the time required to perform tasks. It requires particular cloudlets or activities to fulfill the mission. Response time is more effective in analyzing performance tests and graphic-intensive workloads [24] [25].

- **Resource utilization**

Resource management is one of the main issues in a fog computing environment. It relates to the time needed for tasks to be executed by resources. Utilizing resources efficiently aims to optimize the income and profit of the resource provider while maintaining customer satisfaction. Resource utilization is calculated using the following formula: Resource utilization is the actual time spent by the resource to execute workloads divided by the total uptime of the resource. [26] [27].

- **Network usage**

Network usage refers to the amount of data sent back and forth over the network due to applications, servers, devices, and network users. In many networks, network usage becomes high according to many factors, but much of it comes down to frequent activity [28], [29], [25].

- **Energy consumption**

Energy is the energy consumed by resources to complete the execution workload. This metric is more effective in evaluating performance testing workloads. With an increase in the number of applications, the amount of energy consumed also increases [30] [2].

- **Latency (Delay)**

Latency is a synonym for delay. It is a critical metric for applications sensitive to delay in a fog computing environment. Latency is the expression of how much time it takes for tasks to be sent from one point to another [21], [25], [28], [29], and [30].

- **Total cost**

The total cost of a system can be determined by weighing energy consumption and job processing latency. The number of end-users per fog node increases the average end-user cost [31] [32].

- **Load balancing level**

In a fog environment, load balancing is important because it avoids the situation of overloaded or underloaded fog nodes. Better load balancing leads to better QoS metrics such as energy consumption, resource utilization, throughput, and response time. For better performance, load balancing should be high to maximize resource utilization [27], since the aim of load balancing is to maximize throughput, avoid overload, minimize response time, and optimize resource use.

4. Literature Survey

Several surveys related to fog computing have been published over the past few years. The next section outlines some of the surveys and the main points touched on in each survey.

Naha et al. [33] in their survey help the industry and research community synthesize and identify the requirements for fog computing. At first, it defines the concept of fog computing with architecture in detail. Then a classification of fog computing is introduced based on the requirements of the fog computing paradigm. Finally, it discusses existing research and gaps in resource allocation and scheduling, fault tolerance, simulation tools, and fog-based microservices. Also, it presents some open issues, which will determine the future research direction for the fog computing paradigm.

Nath et al. [34] discuss in this survey the evaluation of distributed computing, from utility computing to fog computing. challenges, architecture, features, technology, security, and privacy in a fog environment. Also, it summarizes the various existing works on fog computing and critically analyzes their pros and cons. Finally, it reviews the future scopes and open research areas in fog computing as an enabler for the next generation computing paradigm.

Hu et al. [3] in this survey define fog computing with its hierarchical structure, challenges, characteristics, and applications. Correspondingly, it reviews the comparison between cloud computing and fog computing. Some of the fog computing applications presented in this survey include gaming, healthcare, brain-machine interfaces, and augmented reality. It also highlighted key technologies such as naming, resource management communication, storage technologies, security, and privacy. Finally, introduce some challenges and open issues that are worth further in-depth study and research in fog computing development.

Mukherjee et al. [35] started this survey with an overview of fog computing, its fundamentals, and its architecture. It also summarized the resource allocation approach to address some of the problems, such as latency, energy consumption, and bandwidth. This survey introduced an extensive overview of state-of-the-art network applications and major research aspects for designing these networks. In addition, this survey reviews the main open challenges and research trends in the fog computing environment.

Fahimullah et al. [36] review the resource allocation approaches based on machine learning (ML) that have been provided in the FC environment. The authors in this paper divided the resource allocation approaches into six categories: resource provision, application placement, scheduling, resource allocation, task offloading, and load balancing. It presents the main points for each category: main approaches, objective matrices, tools, datasets, and comparison.

Tran-Dang *et al.* [37], in order to allocate computing resources for task computation and execution in the fog computing environment, conducted a literature review on the subject. It illustrates the algorithm model and function that assist in the determination of the best decisions in many real-world applications (such as games, robotics, and finance) based on reinforcement learning (RL). Then determine and examine these methods in relation to the three main issues of work scheduling, task offloading, and resource sharing. Finally, the main issues with RL-based algorithms, the fog computing environment, and the computing jobs in the many practical applications were also studied and analyzed in the paper. For additional research, the associated open issues are also mentioned.

Yi *et al.* [38] examine definitions of fog computing with comparable concepts, provide examples of applications that will advance fog computing, and explore numerous issues that may come up when designing and implementing fog computing systems. In addition, issues relating to QoS, interface, resource management, security, and privacy are emphasized, along with new opportunities and difficulties in fog computing for related methodologies. With the underlying IoT, edge devices, radio access techniques, SDN, NFV, virtual machines, and mobile clouds all developing quickly, fog computing will also advance. Although we believe fog computing to be a promising field, "fog computing" now requires cooperation from the underlying methodologies.

Matrouk and Alatoun’s [7] study reviews and analyzes the most significant current scheduling methods in fog computing. The best scheduling algorithms have been chosen after reading and analyzing the majority of recent articles on scheduling algorithms. Task scheduling, resource scheduling, resource allocation, job scheduling, and process scheduling are the five key areas into which this survey divides scheduling issues. According to the results of the comparison, task scheduling accounts for 57% of all utilization of scheduling algorithms in the literature. And 36% of the study publications have used the iFogSim program to put the suggested strategy into practice. By 25%, the makespan is the scheduling algorithm that is most frequently used.

The proposed study provides an introduction to the concepts of fog computing. It focused on a previous group of research within a specific time period, some of which highlighted some of the challenges that exist within the field. However, the differences it concluded can be summarized in the following points: The core concepts explored in this study differentiate it from related works by examining the following aspects:

- Modern resource allocation approaches in fog computing in the last five years (2017 –2023).
- Categorization of these approaches based on four metrics that affect the quality of service (QoS). It provides a unique perspective on their effectiveness in meeting performance criteria.

- A comprehensive summary of all the approaches that were selected is set out in the table to make it easier for the reader to know the behavior of each approach.
- Illustration of the popular evaluation metrics employed for assessing the efficacy of resource allocation approaches.
- Investigation and presentation of prevalent simulation techniques primarily utilized within the fog computing environment enrich our understanding of their applicability and significance.
- Discussion of the primary challenges and issues encountered in the context of resource allocation within fog computing, addressing distinct aspects that set it apart from other related works.

5. Classification Resource Allocation Approaches in Fog Computing

In this section, review the resource allocation approaches based on QoS metrics. This study will review only four metrics: delay and energy consumption, mobility, heterogonous fog nodes, and scalability.

5.1. Delay and Energy Consumption

Fog computing has emerged to provide better QoS and guarantee to minimize delay, especially for applications that are sensitive to delay. The network transmission delay and energy consumption can be improved by using efficient resource allocation methods that consider these two factors. In the following, some of the resource allocation approaches are used to minimize delay and energy consumption.

In [39], this study provides resource allocation and management techniques to increase the fog environment's reliability in IoT-based systems. Latency and energy efficiency are taken into consideration while allocating resources. In a fog, users can choose to prioritize cost-effectiveness over speed. The performance of the simulation was compared with an existing state-of-the-art strategy using the simulation tool iFogSim2. In the proposed technique called Reliable Resource Allocation and Management (R2AM), information from fog nodes is initially kept in queue 2, whereas data from IoT devices is initially put in a queue for later use. The IoT data is then distributed to the fog nodes in accordance with the sorted list after the fog nodes are rated in decreasing order based on their processing time. Once the IoT data has been correctly processed, the results are returned. When compared to the existing technique, the proposed strategy lowered latency by 10.3% and energy consumption by 21.85%, according to the data.

In [40], this paper discusses the Weighted Greedy Knapsack (WGK) method to supply specific services and resources in the smart parade scenario. To formulate the resource allocation problem in fog computing, this study employed a WGK approach. The weighted sum method was used in the multi-objective approach to create the objective function. This method allows for speedy processing of the suggested algorithm. Following module installation in fog devices, the desired modules are allocated the best physical resources of the fog device in accordance with weighted greedy knapsack (WGK)-based allocation. The suggested method is examined using pre-existing algorithms based on several setups, including zones, cameras, mobile devices, and fog devices. According to simulation results, WGK beats concurrent, first-come, first-served (FCFS), and delay-priority algorithms for the smart parade application in

In [27], the authors introduced an efficient prediction algorithm for resource allocation in smart healthcare systems called Effective Prediction and Resource Allocation Methodology (EPRAM). The main goal of the proposed algorithm is to achieve better QoS and reduce latency. This algorithm consists of three modules: the first is the data processing module, the second is the resource allocation module, and the last is the efficient prediction module. Unlike other resource allocation algorithms, this whole system uses a deep reinforcement learning algorithm in a new step. It also uses the PNN method to detect the heart attack probability faster than another multilayer predictor. EPRAM has proven its effectiveness in accurately and quickly predicting the patient's state. Also, it minimizes average resource utilization.

In [30], this research utilizes a meta-heuristic Particle Swarm Optimization (PSO) technique to reduce latency and power consumption. This model considers the network delay and serving rate of fog nodes. It used the iFogSim simulator to set up studies and establish an EEG game-based case study network. This program's output demonstrates that the suggested POS method performs better than existing algorithms, including First Come, First Serve and the Greedy Knapsack-based scheduling algorithm. The conclusion of the simulation optimizes latency and power in terms of energy consumption and overall execution costs.

In [41], this paper suggests a fog-based spider web algorithm (FSWA), a heuristic method that improves response time (RT) and decreases delay time (DT) during workflow throughout the fog network's numerous edge nodes. The basic goal is to find and compute on the closest f-node while minimizing latency between the network's various nodes. The smooth allocation of resources, the availability of services, and the quality of service (QoS) metrics will all improve with a reduction in latency. When it comes to problems with resource optimization in remote computing settings, latency can have a significant impact. Fog computing has significantly lower latency compared to cloud computing.

In [24], the work introduced a hybrid approach, taking into account network-level and node-level strategies to minimize delay and energy consumption using caching schemas. In the proposed approach, the nodes are classified into clusters according to the type of service. When the request comes in, it is handled by the gateway and sent to a suitable cluster of fog nodes that will select an active fog node based on the current energy state and capacity to service the request. Then the fog node will save the popular contents by using a filtration mechanism. The popular content will be saved by using Zipf distribution. It used a load-balancing algorithm to distribute the load between fog nodes. Simulation results show that the advanced caching schema reduces the latency by 85.29% when compared to without caching and by 67.4% when using the caching schema. Also, the proposed method reduces consumer energy consumption by 92.6% without caching and by 82.7% when using a caching schema.

In [42], the researchers proposed a resource representation model. This methodology enables the exposure of device-specific resources via Mobile Edge Computing Application Programming Interfaces (MECAPI) to improve resource allocation in a fog environment. In this study, resource allocation was formulated as a Lyapunov optimization problem. The information obtained from MECAPI, such as CPU, memory, storage, and networking, is utilized by the fog's supervisory entity to make appropriate judgments regarding the distribution of jobs to each network node. The outcome reveals that the suggested model, which combines resource allocation optimization with resource representation, minimizes latency and enhances system performance.

In [43], the study proposed a device- and human-driven intelligent method to minimize latency and energy consumption in a fog computing environment. This method was implemented in two case studies. The first study employed machine learning to identify user behavior and provide an adaptive, low-latency media access control layer among sensor devices. The second study focused on task offloading. This technique is developed for intelligent end-user devices to select the offloading decision in the presence of many fog nodes.

Additionally, it reduces energy and latency. The findings reveal substantial yet unrealized intelligence potential for addressing the challenge of fog computing.

In [44], the authors in this paper proposed a load balancing approach to address the trade-off in F-RANs (fog computing-based radio access networks), with the objective of expanding the centralized cloud radio access networks' (C-RANs') computation and storage capabilities to the network edge. However, the proposed approach solved the trade-off problem of transmission and computing latencies in F-RANs. The simulation results demonstrate that the suggested strategy outperforms the greedy approach to satisfy the essential objectives, including low-latency and limited job offloading to the cloud in the F-RAN for low-latency communications.

In [2], this paper introduced a novel technique known as Gaussian Process Regression for Fog-Cloud Allocation (GPRFCA) for fog and cloud infrastructure. This method anticipates future demand to prevent requests from being blocked. The result of this algorithm's performance is that it optimizes energy consumption at an acceptable level, avoids overloading that minimizes blocking requests, and keeps latency at an acceptable level. This strategy benefits from good features for both cloud-ward and cloud-forward mechanisms.

5.2. Mobility

When data is generated at the edge, consumption and generation can occur at different places and times. Cloud computing does not support or take into account mobility, and data processing can occur at geographically distant data centers [29]. In fog computing, the distribution capacity allows storage and execution to be modeled at different places. The following are some resource allocation approaches that consider mobility.

In [45], the capacity planning framework proposed in this work optimizes the deployment of both fixed and mobile FNs. Utilizing the spatiotemporal variations in demand, it reduces installation and operational costs while maintaining the required QoS. In this study, we offer a data-driven capacity planning framework that uses integer linear programming (ILP) and a heuristic algorithm to generate a cost-optimal deployment plan of CFNs and VFNs from real-world traffic statistics and application characteristics. The system calculates the quantity and variety of FNs required in various areas to meet the demand for computing resources, and it organizes the trajectory and timetables of VFNs based on actual bus schedules. The trial findings show the framework's capacity to lower expenses. The outcomes also demonstrate that, at the cost of higher installation costs, the use of mobile FNs reduces operational expenditures. Furthermore, due to the dense deployment of VFNs, more operational expenses will be reduced over time in times and locations with higher traffic density and a larger daily variation.

In [20], MobFogSim, an expansion of iFogSim, will be used to overcome the mobility issue. MobFogSim was tested by comparing simulation results to those produced from a real testbed in which containers provide fog services. In this study, further MobFogSim trials are conducted that consider the various mobility patterns of a user influenced by Luxembourg SUMO traffic. The findings of this study indicate that MobFogSim can provide a valuable foundation for supporting fog computing for mobile user apps.

In [46], a general three-tiered fog computing architecture was suggested, and the mobility of user equipment was described by the amount of time spent in each coverage area of fog nodes, which was found to follow an exponential distribution. An NP-hard mixed-integer nonlinear programming model was used to describe the issue. The two components of this issue are task offloading and resource allocation. It used a Gini coefficient-based fog computing selection algorithm to get sub-optimal offloading. To solve the computation resource allocation problem, a resource optimization algorithm based on a genetic algorithm was implemented. The simulation result indicates the introduced algorithm can achieve optimal revenue performance compared with other baseline algorithms.

In [29], the scheduling issue in fog and cloud computing's hierarchical composition was first introduced. The scheduling algorithm needs to be flexible enough to handle user requests on the go and various application types by keeping tabs on information in both the local fog and the cloud. The research looks at how an app's performance can impact a user's mobility and how that can be considered to optimize its implementation. In addition, the difficulties posed by fog users' tendency to move around and change locations were highlighted.

5.3. Heterogeneous Fog Nodes

Unlike cloud computing, the fog computing architecture and the devices are heterogeneous and distributed, so the applications need to be designed to work more easily with fog nodes [25]. In the following, some resource allocation approaches highlight heterogeneous factors in fog computing.

The authors in this study [47] reduce network consumption and latency by creating an algorithm that dynamically assigns the right sensor devices to fog nodes. From the rate of sensing frequency of the sensor coupled to the edge device, the proposed technique calculates the volume of information detected by the edge device. The suggested policy considers the heterogeneity and processing power of the devices when connecting the network nodes. The comparison's findings demonstrate that the suggested algorithm significantly lowers processing costs in the cloud, delays, and network consumption. Any application can be executed using the suggested technique. More applications of the proposed design will be implemented in my future work, and the proposed method will be modified to allow for the study of numerous parameters.

In order to reduce the cost per end-user, which is a weighted sum of energy consumption and processing time, the authors [31] proposed a new offloading approach. In order to complete the tasks, this study makes use of the fact that the fog computing nodes are not homogeneous and have varying CPU frequencies. The primary goal of this study is to identify the minimum amount of data required to complete a task that can be processed locally as well as the maximum amount of data that can be offloaded to the most desirable fog node and the faraway cloud, subject to the constraints of available resources and processing time. In this work, the offloading profile and ideal cost of the offloading are visualized across a wide variety of parameter values via simulation.

In [25], for diverse resource constraints in a fog computing setting, the authors presented a policy based on distributed microservices for deploying Internet of Things applications. Putting microservices as close to the data source as possible uses their decentralized and scalable nature, reducing latency and network consumption. In addition to the planned decentralized placement, service discovery, and load balancing, a fog node architecture was proposed to support these features. A simulation utilizing the iFogSim tool demonstrated that the strategy might reduce latency and bandwidth consumption by as much as 85%. Simulated findings also show that the time it takes to deploy a microservice is significantly less than a centralized deployment.

As proposed in [48], an incentive mechanism based on contract theory was used to incentivize nodes at the network's periphery. The negotiation between fog nodes and the task publisher must be optimized for the problem to be solved. When a fog node and a task publisher work together, they reach Nash equilibrium, the best possible contract. The architecture can provide user hardware with enough computing units for computation offloading. The ideal node, as shown by the simulation results, can guarantee the individual rationality and incentive compatibility of the fog nodes while maximizing the utility for the task publisher.

5.4. Scalability

Scalability requirements in cloud computing are hard to meet in applications such as living systems [28]. Hence, it needs to propose smart approaches that do not rely on the cloud model for processing or execution tasks, following some resource allocation approaches that consider scalability factors.

In [21], this study proposed an extensible and scalable simulation called EmuFog in a fog computing environment. This simulation embeds fog nodes in the network topology. It executes Docker-based apps on nodes connected to an emulated network, allowing the researcher to create the architecture of a network according to the use case. Both synthetic and real-world network topologies were used to test this emulation's scalability and effectiveness. The results showed that it scales well and is effective regardless of the network's size.

In [49], the authors considered that the computational loads are transferred to an edge device, and a single edge device is insufficient. In a fog computing network, the analysis, sensing, and transmissions between edge nodes help to enhance scalability in this environment. This study analyzes the positive cases of malaria vector-borne disease-affected information from 2001 to 2014 in Maharashtra state, India. The architecture of Fog2Fog enhances the scalability of a health GIS system.

In [28], this paper presents an expanded cloud IoT architecture for optimizing network bandwidth and empowering edge devices to do intelligent processing independently of the cloud. It conforms to the Spin-Leaf network architecture. This research demonstrated that low-latency, high-bandwidth apps may send and receive data between the cloud and edge devices without degrading QoS. The Spin-Leaf Fog Computing Network (SL-FCN) was used to lessen network congestion and delay times. Because it coexists with the cloud computing data center, this design is scalable. The findings showed that the FCN provided dependable QoS while maintaining fault tolerance for traffic demands.

Table 3 consists of the following main entries: QoS metrics, research problems, research objectives, and contributions, as well as the platform and metrics used in the evaluation, as well as the pros and cons, are also illustrated.

Table 3: Summary of Fog Computing-based Resource Allocation Approaches

Ref	Index of Publications	QoS Metric	Research Issue	Objective/ Contribution	Platform and Metris	Pros and Cons
[40] Shaikh. et al. (2023)	Goggle Scholar , Scopus, WoS	delay and energy consumption	Effective resource allocation and management systems must be designed to	Resource allocation and management strategy take which into consideration	Sim(iFogSim2) Metc: Delay and energy	✓Able to reduce latency by 10.3% and energy consumption by 21.85%. ✓Manage resource allocation in IoT transportation.

			accommodate changing user needs.	latency and energy efficiency when assigning resources.	consumption.	• There is a potential for execution failure due to limited communication device range.
[41] Shruthi .et al. (2022)	Goggle Scholar, Scopus	delay and energy consumption	The resources are required for each educational application that includes several modules to run.	Proposed WGK algorithm for the smart parade scenario to provide certain services/ resources.	Sim.: (iFogSim) Met: Energy consumption and total execution cost.	✓Better energy consumption and consumption cost . ✓Better performance than FCFS and Delay-priority algorithms.
[27], Talaat et al. (2022)	Goggle Scholar, WoS	delay and energy consumption	Effective resource management in fog environment via real-time resource allocating.	A proposed EPRAM in order to minimize the average resource utilization and increase accurately	Sim.: (iFogSim) Metc.: Load balancing level, Turn-around time, Average resource utilization, Waiting time.	✓A suitable algorithm in the case of a real-time system leads to load balancing. ✓Effective in monitoring and predicting the state of a patient accurately and quickly in the healthcare system. • More distributed requirement. • Need to be tested at different levels.
[30] Jabour and Al-Libawy (2021)	Goggle Scholar, Scopus	delay and energy consumption	Addressing the distributed tasks in different applications of IoT via fog nodes can affect QoS and reaction time.	A suggested PSO algorithm to be the main part of the proposed approach in order to manage resources (power and latency).	Sim.: (iFogSim) Metc.: Latency and Average energy.	✓Locate an optimum allocation to reduce the energy consumption for the devices. ✓The proposed approach particularly enhances the response times of IoT (VRGame) applications.
[42] Dar et al. (2020)	Goggle Scholar, Scopus	delay and energy consumption	Trace and locate the nearest f-node for computation and to reduce the latency across the various nodes in a network.	Propose a Fog-based Spider Web Algorithm (FSWA), which reduces the delays time (DT) and enhances the response time (RT).		✓improve the latency and RRT interactions among various nodes in fog computing. ✓searches the available proximal computing resources. ✓
[24] Shahid et al. (2020)	Goggle Scholar, Scopus, WoS	delay and energy consumption	Efficient content distribution	A proposed caching mechanism based on popularity in content delivery fog networks	Simulator (N/A). Metc.: Energy, Time taken	✓Popular contents are found using random distribution. ✓Performs load balancing mechanism to evenly distribute the load in Fog network to eliminate energy holes problem.

						✓Improved energy efficiency and reduces Fog network delay. • Random popularity.
[43] Amine et al. (2019)	Goggle Scholar , Scopus	delay and energy consump tion	Diversity of physical resources available in each device and distributing the treatment efficiently and dynamically.	A proposed a resource representation method that qualifies to expose the resources of each device via MEC APIs for optimizing RA	Sim.: (N/A). Metc.: Request arrival rate.	✓A problem of resource allocation had been formulated as a Lyapunov optimization. ✓The impact of the proposed method on latency. ✓Improve the performance of the system.
[31] Duy et al. (2019)	Goggle Scholar , Scopus, WoS	delay and energy consump tion	The requirements of latency and energy efficiency for time-critical IoT applications.	A proposed a device-driven and human-driven intelligence approach as essential enablers to decrease latency and energy consumption in fog computing	Sim.: (OpenMot e-CC2538). Metc.: Delay, Packet delivery ratio radio, Duty cycle, Power consumpti on, and Goodput.	✓Investigated the benefit of intelligence as a factor for fog computing. ✓Two case studies were employed. ✓Select its offloading decision in the presence of multiple fog nodes nearby, ✓Minimize energy and latency objectives
[44] Mukher jee. et al. (2018)	Goggle Scholar , Scopus, WoS	delay and energy consump tion	The transmission latency between F-APs, F-AP-to-end-user, and fronthaul latency strongly depends on interference power.	Load balancing scheme to address the tradeoff between transmission and computing latencies in F-RANs	Sim: (N/A) Metc:laten cy and % of content fetched from the cloud.	✓Low latency. ✓Minimize task offloading to the cloud. Need to optimize the near-optimal latency to low latency and high-reliability.
[2] Rodrig e S. et al. (2018)	Goggle Scholar , Scopus	delay and energy consump tion	Decision-making process on where to allocate resources to run the tasks of an application.	A proposed a novel GPRFCA mechanism for RA in infrastructure composed of combined clouds and fogs.	Sim.: (iFogSim). Metc.: energy consumpti on blocking ratio latency	✓Maintain the energy consumption at appropriate levels ✓Avoid overloading the fog decreasing requests blocking. ✓Kept latency at appropriate levels, especially delay-sensitive ones. • Evaluation of energy consumption for only to end user-device. • Not consideration of multiple levels of fog nodes.
[45] Mao.et al.	Goggle Scholar ,	Mobility	Capacity planning, which decides	Propose a data-driven capacity	Sim.: (N/A)	✓reduce costs. ✓The deployment of mobile FNs saves

(2022)	Scopus		where and how much computing resources to deploy.	planning framework that optimizes the deployment of stationary and mobile FNs.	Metc: latency, traffic flow, CDF.	operational costs at expense of additional installation cost. It lacks to short-term on-demand scheduling
[20] Puliafito.et al (2020)	Goggle Scholar, Scopus, WoS	Mobility	Migration of a fog service	A proposed MobFogSim to overcome the limitation of consumer mobility.	Sim.: (MobFogSim) Metc.: migration time, downtime, data transferred, network use, average number of migrations (live/cold), average delays	✓Significant and interesting study. ✓Enable modeling of device mobility and service migration in fog computing. ✓MobFogSim is capable to use an user-defined migration method. ✓Additional investigations were carried out in MobFogSim taking account of various mobility patterns of a user, derived from Luxembourg SUMO traffic. ✓MobFogSim supplies a reasonable basis for keeping applications fog computing in case of the consumer is mobile and a migration method is required to move the state.
[46] Wang.et al (2019)	Goggle Scholar, Scopus	Mobility	Limited service coverage of fog computing nodes	A proposed mobility-aware offloading and computation allocation algorithm in fog computing networks.	Sim.: (N/A) Metc.: revenue of time and energy, computing capacity, revenue with different mean value, migration cost	✓Reduced the probability of migration so as to maximize the total revenue of UEs. ✓The proposed algorithm was considering mobility can effective. ✓Achieve quasi-optimal revenue performance compared with other baseline algorithms • How to reduce the cost of migration for the migrated tasks
[29] Bittencourt.et al (2017)	Goggle Scholar, Scopus	Mobility	Scheduling problem in the hierarchical composition of fog and cloud computing	An investigation of the scheduling problem in fog computing concentrates on how consumer mobility may impact application performance	Sim.: (N/A) Metc.: delay, application modules, total network use.	✓Address and consider scheduling and resource management strategies that require types of user applications, the range, and the mobility of smart. ✓Improved execution based on application characteristics. ✓Classified application and user mobility as two key factors to be

						associated with scheduling in supplying efficient resource management.
[47] Hassan. et al. (2022)	Goggle Scholar , Scopus, WoS	Heterogeneous Fog Nodes	The connection of suitable sensor nodes to the parent fog node plays an essential role in achieving the optimum performance of the system.	Dynamically assigns appropriate sensor devices to fog nodes to achieve a reduction in network utilization and latency.	Sim.: iFogSim toolkit. Metc: latency and network usage.	✓Reduce the processing cost, delay and network consumption. ✓Capable to execute at any type of application. • Need to deploying more application on the proposed system. • Need to more analyze and design due node failure in system.
[32] Mukherjee. et al (2020)	Goggle Scholar , Scopus	Heterogeneous Fog Nodes	A task offloading for the end-users (task processing time) in the fog-cloud environment	A proposed an offloading approach to find the optimal amount of offloaded task data under energy constraints and delay.	Sim.: (N/A) Metc.: total cost cost per fog node average number of received offloading tasks	✓Considering the heterogeneous nature (with different CPU clock speeds) of the fog computing nodes. ✓Minimized the total cost of the system. ✓Applied the SDR to the QCQP problem. • Studying and investigating deadline-aware task offloading in fog-cloud networks are required.
[25] Pallewatta. et al. (2019)	Goggle Scholar , Scopus	Heterogeneous Fog Nodes	Microservices-based IoT applications within Fog environments.	A proposed decentralized microservices placement algorithm for microservices-based IoT applications	Sim.: (iFogSim) Metc.: latency, network usage, time required for application microservices placement,	✓A proposed decentralized placement algorithm for microservices-based IoT applications in fog networks. ✓A support decentralized placement for fog node architecture along with load balancing and service discovery. ✓Handling service discovery and load balancing-related challenges of the microservices architecture. • A failure of the fog node • A lower-level fog node is not considered.
[48] Zeng. et al. (2018)	Goggle Scholar , Scopus, WoS	Heterogeneous Fog Nodes	A share idles computing resources to a fog node in a fog environment as an optimization problem.	A proposed incentive is a framework of contract theory to motivate fog nodes to share their idle computing resources.	Sim.: (N/A) Metc.: utility of tasks publisher utility of fog node	✓The developed optimal contract is the Nash equilibrium the solution acquired by the task publisher and fog nodes. ✓Eliminates the impact of information asymmetry and specifies compensation to fog nodes according to their kinds.

						✓The utility of the task publisher is maximized while ensuring the personal rationality and motivation compatibility of fog nodes. • The contract-based method should be generalized in case of fog node types have different distributions.
[21] Mayer. et al. (2018)	Goggle Scholar , Scopus	Scalabilit y	An experimental evaluation of protocol or/and an application design in a controllable and repeatable manner.	A proposed EmuFog as an extensible and scalable emulation framework for fog computing environments.	Sim.: (EmuFog) Metc.: Latency	✓A proposal was made to design the network topology according to the use case, and embed Fog Computing nodes in the topology. ✓Enables the from-scratch design of Fog Computing infrastructures and the emulation of real applications and workloads. • This simulator needs more capabilities. • Required to embed mobility models both for clients and Fog nodes.
[49] Barik.e t al. (2017)	Goggle Scholar , Scopus	Scalabilit y	Scalability in fog computing for Geo-health data systems.	A proposed Fog2Fog architecture in order to enhance the scalability of health GIS systems	Sim.: (N/A) Metc.: Probability of blocking	✓Improving scalability in fog frameworks. • It was applied on limited area
[28] Okafor. et al (2017)	Goggle Scholar , Scopus, WoS	Scalabilit y	A scalable IoT data-canter environment functions in fog computing	A proposes an extended cloud IoT model that optimizes bandwidth while allowing edge devices to process data intelligently without relying on a cloud network.	Sim.: (N/A) Metc.: latency network load usage	✓Low-latency and bandwidth-intensive applications can transfer data to the cloud without impacting QoS performance. ✓Network congestion and reducing latency issues in a highly distributed and multilayer virtualized IoT data-center environment • Need more studies to validate this model for IoT data stream processing.

7. Open Issues in Resource Allocation

Despite modern approaches, many resource allocation problems are related to fog computing. However, the number of devices is expected to continuously increase, may be connected to the Internet, and create more and more data. Thus, the creation and consumption of data require scalable resource management. In this section, we highlight many points that can help the research community address them.

- ***Mobility***: According to the distribution capacity provided by the fog model, many tasks can be executed and processed at various locations. Resource allocation becomes hard and complex in the mobility environment due to many challenges, such as time and distance constraints. Also, the mobility of smart devices requires efficient resource management and scheduling that take mobility into consideration.
- ***Scheduling tasks***: The main goal of scheduling tasks is to assign a set of tasks to fog nodes to justify the QoS requirements optimally and minimize the execution and transmission time of these tasks depending on the available set of resources.
- ***Heterogeneous fog applications:*** The devices in the fog computing environment are heterogeneous, resource-constrained, and distributed, so the IoT applications need to be designed with efficient resource allocation approaches to work more easily with fog nodes.
- ***Management of resources***: In any system, the management of resources is an important factor in avoiding congestion and overload, so efficient administration must be achieved through the efficient distribution of the available resources, which is the main issue in the fog computing resource allocation problem.
- ***Application placement***: Placement is important in managing resources in a fog computing environment because it is classified into three categories. The first category is centralized placement, which requires information from all devices in a fog environment. The second category is a decentralized placement in which the brokers have part of the information, and the third is a hierarchical placement.
- ***Energy***: In a fog computing environment, users must decide whether to offload tasks to near fog nodes based on energy consumption. Thus, how to save energy consumption and provide efficient approaches for efficient offloading needs to be addressed.
- ***Delay***: In any environment, for the best user experience, we should execute and process tasks with a reduced service delay for end-user applications, especially those sensitive to delay, and optimize key performance metrics for users.

6. Conclusion

In order to alleviate the strain on cloud computing resources, fog computing has been included in a wide range of cloud-enabled systems (such as IoT-enabled applications, healthcare applications, management applications, etc.), which have improved system performance. The variety, mobility, and dynamic change of the fog computing environment, however, make it a complex resource pool and major obstacles to creating an effective and efficient resource allocation approach. The main goal of this study is to succinctly review recent progress in fog computing-based resource allocation approaches. Overall, the paper summarizes the present state of the relationship between fog computing and resource allocation. It creates an understanding for the readers through an overview, classification, and the opportunity for future research. For this purpose, we started by describing several topics that have been included and explained, such as fog computing's definition, architecture, and functionalities and features. Then highlight the most prominent simulator environments and popular evaluation metrics for fog computing. Moreover, modern proposed approaches from 2017 to 2023 were presented with their classification according to QoS metrics, and comparison also showed the main pros, cons, work environment, metrics, research issues, and objectives for each approach. Finally, several research issues related to fog computing were briefly explained. Our future work will be to extend and investigate more approaches based on metrics that may affect efficiency in the fog domain and its applications.

References

[1] I. Martinez, A. S. Hafid, and A. Jarray, "Design, Resource Management, and Evaluation of Fog Computing Systems: A Survey," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2494–2516, 2021, doi: 10.1109/JIOT.2020.3022699.

[2] R. A. C. Da Silva and N. L. S. Da Fonseca, "Resource Allocation Mechanism for a Fog-Cloud Infrastructure," *IEEE Int. Conf. Commun.*, vol. 2018-May, pp. 1–6, 2018, doi: 10.1109/ICC.2018.8422237.
[3] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on Fog Computing: Architecture, Key Technologies, Applications and Open Issues," *J. Netw. Comput. Appl.*, vol. 98, no. September, pp. 27–42, 2017, doi: 10.1016/j.jnca.2017.09.002.
[4] H. Wadhwa and R. Aron, "TRAM: Technique for Resource Allocation and Management in Fog Computing Environment," *J. Supercomput.*, vol. 78, no. 1, pp. 667–690, 2022, doi: 10.1007/s11227-021-03885-3.
[5] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga, "Resource Management Techniques for Cloud/Fog and Edge Computing: An Evaluation Framework and Classification," *Sensors*, vol. 21, no. 5, pp. 1–23, 2021, doi: 10.3390/s21051832.
[6] S. Rahul and R. Aron, "Fog Computing Architecture, Application and Resource Allocation: A Review," *CEUR Workshop Proc.*, vol. 2889, pp. 31–42, 2021.
[7] K. Matrouk and K. Alatoun, "Scheduling Algorithms in Fog Computing: A Survey," *Int. J. Networked Distrib. Comput.*, vol. 9, no. 1, pp. 59–74, 2021, doi: 10.2991/IJNDC.K.210111.001.
[8] M. Mukherjee, L. Shu, and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018, doi: 10.1109/COMST.2018.2814571.
[9] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: A review," *Big Data Cogn. Comput.*, vol. 2, no. 2, pp. 1–18, 2018, doi: 10.3390/bdcc2020010.
[10] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 416–464, 2018, doi: 10.1109/COMST.2017.2771153.
[11] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on Fog Computing: Architecture, Key Technologies, Applications and Open Issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, 2017, doi: 10.1016/j.jnca.2017.09.002.
[12] H. Rafique, M. A. Shah, S. U. Islam, T. Maqsood, S. Khan, and C. Maple, "A Novel Bio-Inspired Hybrid Algorithm (NBIHA) for Efficient Resource Management in Fog Computing," *IEEE Access*, vol. 7, pp. 115760–115773, 2019, doi: 10.1109/ACCESS.2019.2924958.
[13] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, "Fog Computing for the Internet of Things: A Survey," *ACM Trans. Internet Technol.*, vol. 19, no. 2, 2019, doi: 10.1145/3301443.
[14] D. T. Nguyen, L. B. Le, and V. K. Bhargava, "A Market-Based Framework for Multi-Resource Allocation in Fog Computing," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1151–1164, 2019, doi: 10.1109/TNET.2019.2912077.
[15] B. Jamil, H. Ijaz, M. Shojafar, K. Munir, and R. Buyya, "Resource Allocation and Task Scheduling in Fog Computing and Internet of Everything Environments: A Taxonomy, Review, and Future Directions," *ACM Comput. Surv.*, vol. 10, no. 01, 2022, doi: 10.1145/3513002.
[16] I. Ben Lahmar and K. Boukadi, "Resource Allocation in Fog Computing: A Systematic Mapping Study," *2020 5th Int. Conf. Fog Mob. Edge Comput. FMEC 2020*, pp. 86–93, 2020, doi: 10.1109/FMEC49853.2020.9144705.
[17] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, "Resource Management Approaches in Fog Computing: a Comprehensive Review," *J. Grid Comput.*, vol. 18, no. 1, 2020, doi: 10.1007/s10723-019-09491-1.
[18] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in the Internet of Things, Edge and Fog Computing Environments," *Softw. - Pract. Exp.*, vol. 47, no. 9, pp. 1275–1296, 2017, doi: 10.1002/spe.2509.
[19] D. Seo *et al.*, "Dynamic iFogSim: A Framework for Full-Stack Simulation of Dynamic Resource Management in IoT Systems," *2020 Int. Conf. Omni-Layer Intell. Syst. COINS 2020*, no. February 2021, 2020, doi: 10.1109/COINS49042.2020.9191663.
[20] C. Puliafito *et al.*, "MobFogSim: Simulation of Mobility and Migration for Fog Computing," *Simul. Model. Pract. Theory*, vol. 101, no. July, 2020, doi: 10.1016/j.simpat.2019.102062.
[21] R. Mayer, L. Graser, H. Gupta, E. Saurez, and U. Ramachandran, "EmuFog: Extensible and Scalable Emulation of Large-Scale Fog Computing Infrastructures," *2017 IEEE Fog World Congr.*

FWC 2017, pp. 1–6, 2018, doi: 10.1109/FWC.2017.8368525.
[22] T. Qayyum, A. W. Malik, M. A. K. Khattak, O. Khalid, and S. U. Khan, "FogNetSim++: A Toolkit for Modeling and Simulation of Distributed Fog Environment," *IEEE Access*, vol. 6, pp. 63570–63583, 2018, doi: 10.1109/ACCESS.2018.2877696.
[23] I. Lera, C. Guerrero, and C. Juiz, "YAFS: A Simulator for IoT Scenarios in Fog Computing," *IEEE Access*, vol. 7, pp. 91745–91758, 2019, doi: 10.1109/ACCESS.2019.2927895.
[24] M. H. Shahid, A. R. Hameed, S. ul Islam, H. A. Khattak, I. U. Din, and J. J. P. C. Rodrigues, "Energy and Delay Efficient Fog Computing Using Caching Mechanism," *Comput. Commun.*, vol. 154, pp. 534–541, 2020, doi: 10.1016/j.comcom.2020.03.001.
[25] S. Pallewatta, V. Kostakos, and R. Buyya, "Microservices-Based IoT Application Placement Within Heterogeneous and Resource Constrained Fog Computing Environments," *UCC 2019 - Proc. 12th IEEE/ACM Int. Conf. Util. Cloud Comput.*, pp. 71–81, 2019, doi: 10.1145/3344341.3368800.
[26] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, "Performance Evaluation Metrics for Cloud, Fog and Edge Computing: A Review, Taxonomy, Benchmarks and Standards for Future Research," *Internet of Things (Netherlands)*, vol. 12, no. September, 2020, doi: 10.1016/j.iot.2020.100273.
[27] F. M. Talaat, "Effective Prediction and Resource Allocation Method (EPRAM) in fFog Computing Environment for Smart Healthcare System," *Multimed. Tools Appl.*, vol. 81, no. 6, pp. 8235–8258, 2022, doi: 10.1007/s11042-022-12223-5.
[28] K. C. Okafor, I. E. Achumba, G. A. Chukwudebe, and G. C. Ononiwu, "Leveraging Fog Computing for Scalable IoT Datacenter Using Spine-Leaf Network Topology," *J. Electr. Comput. Eng.*, vol. 2017, 2017, doi: 10.1155/2017/2363240.
[29] O. F. and P. Bittencourt, Luiz F., Diaz-Montes, Javier, Buyya, Rajkumar, Rana and Manish, "Mobility-Aware Application Scheduling in Fog Computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, 2017, doi: 10.1109/MCC.2017.27.
[30] I. M. Jabour and H. Al-Libawy, "An Optimized Approach for Efficient-Power and Low-Latency Fog Environment Based on the PSO Algorithm," *Proc. 2021 2nd Inf. Technol. to Enhanc. E-Learning other Appl. Conf. IT-ELA 2021*, pp. 52–57, 2021, doi: 10.1109/IT-ELA52201.2021.9773443.
[31] M. Mukherjee *et al.*, "Computation Offloading Strategy in Heterogeneous Fog Computing with Energy and Delay Constraints," *IEEE Int. Conf. Commun.*, vol. 2020-June, no. June, 2020, doi: 10.1109/ICC40277.2020.9148852.
[32] 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, 2020, doi: 10.1002/dac.4340.
[33] R. K. Naha *et al.*, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018, doi: 10.1109/ACCESS.2018.2866491.
[34] S. B. Nath, H. Gupta, S. Chakraborty, and S. K. Ghosh, "A Survey of Fog Computing and Communication: Current Researches and Future Directions," eprint arXiv:1804.04365, no. i, pp. 1–47, 2018, [Online]. Available: http://arxiv.org/abs/1804.04365
[35] M. Mukherjee, L. Shu, and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018, doi: 10.1109/COMST.2018.2814571.
[36] M. Fahimullah, S. Ahvar, and M. Trocan, "A Review of Resource Management in Fog Computing: Machine Learning Perspective," eprint arXiv , pp. 1–15, 2022, [Online]. Available: http://arxiv.org/abs/2209.03066
[37] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq, and D.-S. Kim, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," *J. Commun. Networks*, vol. 24, no. 1, pp. 83–98, 2022, doi: 10.23919/jcn.2021.000041.
[38] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," *Mobidata 2015 - Proc. 2015 Work. Mob. Big Data, co-located with MobiHoc 2015*, vol. 2015-June, no. June 2015, pp. 37–42, 2015, doi: 10.1145/2757384.2757397.
[39] H. U. Atiq, Z. Ahmad, S. K. uz Zaman, M. A. Khan, A. A. Shaikh, and A. Al-Rasheed, "Reliable Resource Allocation and Management for IoT Transportation Using Fog Computing," *Electronics*, vol. 12, no. 6, p. 1452, Mar. 2023, doi: 10.3390/electronics12061452. [Online]. Available:

http://dx.doi.org/10.3390/electronics12061452

[40] G. Shruthi, M. R. Mundada, and S. Supreeth, "The Resource Allocation Using Weighted Greedy Knapsack Based Algorithm in an Educational Fog Computing Environment," *Int. J. Emerg. Technol. Learn.*, vol. 17, no. 18, pp. 261–274, 2022, doi: 10.3991/ijet.v17i18.32363.

[41] A. R. Dar, D. Ravindran, and S. Islam, "Fog-based Spider Web Algorithm to Overcome Latency in Cloud Computing," *Iraqi Journal of Science*, vol. 61, no. 7, pp. 1781–1790, Jul. 2020. doi: 10.24996/ijs.2020.61.7.27.

[42] A. Abouaomar, S. Cherkaoui, A. Kobbane, and O. A. Dambri, "A Resources Representation for Resource Allocation in fog Computing Networks," *2019 IEEE Glob. Commun. Conf. GLOBECOM 2019 - Proc.*, 2019, doi: 10.1109/GLOBECOM38437.2019.9014146.

[43] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. S. Quek, and H. Shin, "Enabling Intelligence in Fog Computing to Achieve Energy and Latency Reduction," *Digit. Commun. Networks*, vol. 5, no. 1, pp. 3–9, 2019, doi: 10.1016/j.dcan.2018.10.008.

[44] M. Mukherjee, Y. Liu, J. Lloret, L. Guo, R. Matam, and M. Aazam, "Transmission and Latency-Aware Load Balancing for Fog Radio Access Networks," *2018 IEEE Glob. Commun. Conf. GLOBECOM 2018 - Proc.*, pp. 1–6, 2018, doi: 10.1109/GLOCOM.2018.8647580.

[45] W. Mao, O. U. Akgul, A. Mehrabi, and B. Cho, "Data-Driven Capacity Planning for Vehicular Fog Computing," in *I*EEE Internet of Things Journal, vol. 9, no. 15, pp. 13179-13194, 1 Aug.1, 2022, doi: 10.1109/JIOT.2022.3143872.

[46] D. Wang, Z. Liu, X. Wang, and Y. Lan, "Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks," *IEEE Access*, vol. 7, pp. 43356–43368, 2019, doi: 10.1109/ACCESS.2019.2908263.

[47] Syed Rizwan Hassan, Ishtiaq Ahmad, Ateeq Ur Rehman, Seada Hussen, and Habib Hamam, "Design of Resource-Aware Load Allocation for Heterogeneous Fog Computing Environments," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 3543640, 11 pages, 2022. https://doi.org/10.1155/2022/3543640

[48] M. Zeng, Y. Li, K. Zhang, M. Waqas, and D. Jin, "Incentive Mechanism Design for Computation Offloading in Heterogeneous Fog Computing: A Contract-Based Approach," *IEEE Int. Conf. Commun.*, vol. 2018-May, pp. 1–6, 2018, doi: 10.1109/ICC.2018.8422684.

[49] R. Barik, H. Dubey, S. Sasane, C. Misra, N. Constant, and K. Mankodiya, "Fog2Fog: Augmenting Scalability in Fog Computing for Health GIS Systems," *Proc. - 2017 IEEE 2nd Int. Conf. Connect. Heal. Appl. Syst. Eng. Technol. CHASE 2017*, pp. 241–242, 2017, doi: 10.1109/CHASE.2017.83.