IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 April 2023, accepted 10 May 2023, date of publication 18 May 2023, date of current version 1 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3277826

RESEARCH ARTICLE

Task Scheduling Mechanisms for Fog Computing: A Systematic Survey

MEHDI HOSSEINZADEH[1,2], ELHAM AZHIR[3,4], JAN LANSKY[5], STANISLAVA MILDEOVA[5], OMED HASSAN AHMED[6], MAZHAR HUSSAIN MALIK[7], AND FAHEEM KHAN[8]

[1]Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
[2]School of Medicine and Pharmacy, Duy Tan University, Da Nang 550000, Vietnam
[3]Research and Development Center, Mobile Telecommunication Company of Iran, Tehran 1991954168, Iran
[4]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran 1477893855, Iran
[5]Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, 101 00 Prague, Czech Republic
[6]Department of Information Technology, University of Human Development, Sulaymaniyah 0778-6, Iraq
[7]School of Computing and Creative Technologies, College of Arts, Technology and Environment (CATE), University of the West of England, BS16 1QY Bristol, U.K.
[8]Department of Computer Engineering, Gachon University, Seongnam 13120, South Korea

Corresponding authors: Mazhar Hussain Malik (mazhar.malik@uwe.ac.uk) and Faheem Khan (faheem@gachon.ac.kr)

ABSTRACT In the Internet of Things (IoT) ecosystem, some processing is done near data production sites at higher speeds without the need for high bandwidth by combining Fog Computing (FC) and cloud computing. Fog computing offers advantages for real-time systems that require high speed internet connectivity. Due to the limited resources of fog nodes, one of the most important challenges of FC is to meet dynamic needs in real-time. Therefore, one of the issues in the fog environment is the optimal assignment of tasks to fog nodes. An efficient scheduling algorithm should reduce various qualitative parameters such as cost and energy consumption, taking into account the heterogeneity of fog nodes and the commitment to perform tasks within their deadlines. This study provides a detailed taxonomy to gain a better understanding of the research issues and distinguishes important challenges in existing work. Therefore, a systematic overview of existing task scheduling techniques for cloud-fog environment, as well as their benefits and drawbacks, is presented in this article. Four main categories are introduced to study these techniques, including machine learning-based, heuristic-based, metaheuristic-based, and deterministic mechanisms. A number of papers are studied in each category. This survey also compares different task scheduling techniques in terms of execution time, resource utilization, delay, network bandwidth, energy consumption, execution deadline, response time, cost, uncertainty, and complexity. The outcomes revealed that 38% of the scheduling algorithms use metaheuristic-based mechanisms, 30% use heuristic-based, 23% use machine learning algorithms, and the other 9% use deterministic methods. The energy consumption is the most significant parameter addressed in most articles with a share of 19%. Finally, a number of important areas for improving the task scheduling methods in the FC in the future are presented.

INDEX TERMS Fog computing, cloud computing, task scheduling, methods, quality of service.

I. INTRODUCTION

With the evolution of digital technologies, a huge amount of data is generated from numerous sources [1]. Such data can be stored and processed using cloud computing solutions. However, cloud computing cannot support the Internet of Things (IoT) mobility and security requirements [2]. Therefore, "Fog Computing (FC)" was presented by Cisco in 2012 [3] to solve these challenges. Fog computing extends the cloud services to the edge of network. It reduced the data transfer time and volume by performing the related operations using the local resources that exist near the IoT edge devices. Therefore, the use of local resources reduces costs, reduces latency, increases the level of confidentiality and security, and reduces the network traffic load. When there are no effective resources in fog computing, cloud resources are used with

The associate editor coordinating the review of this manuscript and approving it for publication was Alessandro Floris.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

higher costs. Fog computing and edge computing are often used interchangeably in literature to refer to the same concept of distributing computing resources closer to the end-users. In this paper, we will use both terms interchangeably to refer to this concept.

Task management in cloud-fog architectures is an important issue. The optimal usage of cloud-fog resources to enhance various qualitative parameters such as task execution time, cost of operations, and energy consumption is an important issue. The suitable task scheduling in the fog environment reduces the costs and processing/communication delays. One of the difficulties of researchers is the selection of an efficient task scheduling method. To achieve the mentioned objective, a comprehensive study and evaluation of task scheduling methods is essential in the fog environment.

Various researchers investigate the characteristics of different task scheduling methods in the fog environment (Table 1). For instance, Bansal et al. [4] reviewed various task scheduling algorithms in fog computing. The existing techniques in the field of fog task scheduling are classified and investigated in four main groups: static, dynamic, heuristic, and hybrid. Various related qualitative parameters such as response time, cost, and energy consumption are assessed for each technique. Furthermore, open issues and some directions for future work are presented. Alizadeh et al. [5] presented a systematic review on the task scheduling algorithms in cloud-fog. The authors reviewed different task scheduling algorithms and discussed the strength and weaknesses of the studied algorithms. Furthermore, various task scheduling tools are presented in this paper. Some open issues regarding the scheduling of tasks in cloud-fog and some directions for future works are presented in this survey. Yang and Rahmani [6] also reviewed 15 articles in the field of fog task scheduling in two main categories: heuristic and meta-heuristics. The strength and limitations of the studied algorithms are investigated in this paper. However, there is a lack of broader classification such as deterministic methods. Also, the latest published articles in the field of fog-cloud task scheduling are not included in this paper. Naha et al. [7] explored the resource management methods in fog computing. The authors presented different definitions of fog computing and discussed the distinctions of among fog and cloud. The classification presented in this paper is based on the necessities of the fog paradigm. Their research direction is focused on the research works in resource allocation, fault tolerance, simulation tools, and fog-based micro services. Numerous qualitative parameters are explored in this survey. However, there is not a clear mechanism for the article selection process. Hosseinioun et al. [8] presented a survey article about fog-based task scheduling from 2015 to 2018. The authors categorized the task scheduling techniques into two groups: dynamic and static. However, this article is written in a non-systematic manner. Therefore, the article selection is not introduced. Kaur et al. [9] reviewed the current literature on fog task scheduling algorithms in a systematic manner. The selected techniques are studied in four groups including heuristic, metaheuristic, deterministic, and hybrid techniques. This article discussed the important challenges of the current methods. The existing solutions of various challenges are also investigated. The authors analyzed the qualitative parameters and tools used for fog task scheduling. This survey helps researchers to recognize future directions to developed efficient scheduling methods. Finally, Matrouk and Alatoun [10] introduced five categories including task scheduling, resource allocation, resource scheduling, workflow scheduling, and job scheduling to review the problem of service management in fog computing environments. The authors discussed the benefits and weaknesses of each study. The presented algorithms are compared based on specific metrics and evaluation tools. However, the mechanism of article selection is unclear. Additionally, future works are not properly explained.

According to the studies in Table 1, there are some weaknesses in the existing cloud-fog task scheduling surveys as follows:

- The mechanism of article selection is not clear and the introduced surveys do not have the systematic arrangement.
- Some surveys did not evaluate the qualitative parameters for studying the techniques.
- Many surveys did not provide a comprehensive and reasonable classification of fog task scheduling techniques.
- All papers do not cover the new existing scheduling techniques, especially in 2022.

As far as we know, this survey is the first of its kind that provides a complete Systematic Literature Review (SLR) on the current task scheduling techniques in fog computing and their comparison in terms of relevant parameters. The main purpose of this study is to review the task scheduling algorithms presented in different articles, categorize them, and analyze their benefits and drawbacks. Four categories are presented to study these techniques, including machine learning, heuristic-based, metaheuristic-based, and deterministic mechanisms. Additionally, some scheduling criteria are used to assess the presented techniques. These criteria are presented as follows:

- Execution time: The time needed to complete the execution of a given task.
- Resource utilization: It refers to minimum usage of resources to execute the maximum number of tasks.
- Delay: The time needed to transfer data across the network.
- Network Bandwidth: It determines the maximum signal rate for transmitting data.
- Energy consumption: It determines the consumed energy by a resource to perform a task.
- Execution deadline: It is the acceptable time for completion of a task.
- Response time: It is the required time to respond to the user's task.
- Cost: The amount of required budget to execute a task.

TABLE 1. State of the art reviews of task scheduling mechanisms for fog computing.

Authors	Year	Journal or Conference Name	Title	Advantages	Weakness
Bansal, et al. [4]	2022	Transactions on Emerging Telecommunications Technologies	A systematic review of task scheduling approaches in fog computing	• The article selection process is clear. • The cloud-fog scheduling mechanisms are reviewed in four groups: static, dynamic, heuristic, and hybrid. • Qualitative parameters are evaluated. • Open issues are discussed. • Some directions for future works are presented.	• It ignores the exact optimization methods for fog task scheduling.
Alizadeh, et al. [5]	2020	International Journal of Communication Systems	Task scheduling approaches in fog computing: A systematic review	• The article selection process is clear. • The cloud-fog scheduling mechanisms are reviewed in four groups: static, dynamic, heuristic, and hybrid. • Qualitative parameters are evaluated. • Open issues are discussed. • Some directions for future works are presented.	• The latest published articles specially after 2020 are not included. • It ignores the exact optimization methods for fog task scheduling.
Yang and Rahmani [6]	2020	Kybernetes	Task scheduling mechanisms in fog computing: review, trends, and perspectives	• The article selection process is clear. • The fog scheduling mechanisms are reviewed in two groups: heuristic and meta-heuristics. • Qualitative parameters are evaluated. • Open issues are discussed. • Some directions for future works are presented.	• A few number of articles are investigated. • The latest published articles specially after 2018 are not included. • The presented taxonomy does not include the main important techniques of this field. • It ignores the exact optimization methods for fog task scheduling.
Naha, et al. [7]	2018	IEEE access	Fog computing: Survey of trends, architectures, requirements, and research directions	• Qualitative parameters are evaluated. • Open issues are discussed. • Some directions for future works are presented.	• Fog task scheduling is not the main focus point of the paper. • There is no specific taxonomy to review fog task scheduling mechanisms. • The article selection process is not clear. • The latest published articles specially after 2018 are not included.
Hosseinioun, et al. [11]	2020	Transactions on Emerging Telecommunications Technologies	Task scheduling approaches in fog computing: a survey	• The fog scheduling mechanisms are reviewed in two groups: dynamic and static. • Qualitative parameters are evaluated. • Open issues are discussed.	• The article is written in a non-systematic manner. • The article selection process is not clear. • The latest published articles specially after 2019 are not included.

TABLE 1. ***(Continued.)*** **State of the art reviews of task scheduling mechanisms for fog computing.**

					• It ignores the exact optimization methods for fog task scheduling.
Kaur, et al. [9]	2021	Concurrency and Computation: Practice and Experience	A systematic review on task scheduling in Fog computing: Taxonomy, tools, challenges, and future directions	• The article selection process is not clear. • The cloud-fog scheduling mechanisms are reviewed in four groups: heuristic, metaheuristic, deterministic, and hybrid. • Qualitative parameters are evaluated. • Open issues are discussed. • Some directions for future works are presented.	• The latest published articles specially after 2020 are not included. • It ignores the exact optimization methods for fog task scheduling.
Matrouk and Alatoun [10]	2021	Int. J. Networked Distributed Comput.	Scheduling Algorithms in Fog Computing: A Survey	• The cloud-fog scheduling mechanisms are reviewed in five groups: task scheduling, resource allocation, resource scheduling, workflow scheduling, and job scheduling. • Qualitative parameters are evaluated.	• The article selection process is not clear. • A few number of articles are investigated in the field of fog task scheduling. • There is no specific taxonomy to review fog task scheduling mechanisms. • Open issues are not presented.

The main objectives of this study are:

- Presenting a comprehensive systematic review of the task scheduling techniques in the fog environment.
- Providing a classification for task scheduling techniques in fog computing.
- Reviewing and comparing the selected task scheduling algorithms.
- Discussing open issues to provide new research directions in the future.

The presented study includes several types of cloud-fog task scheduling techniques with a deep understanding of the limitations and challenges related to existing scheduling methods to provide an in-depth meta-analysis in order to design effective scheduling methods. This survey enables researchers to select the appropriate scheduling technique for a specific task.

The following classification will be discussed in this article. The background is reviewed in Section II. In Section III, the article selection method is provided. The intended taxonomy for the selected fog task scheduling papers and the selected papers are investigated in Section IV. The reviewed studies will be compared and discussed in Sections V. Finally, some open issues and the conclusion are presented in Sections VI and VII.

II. BACKGROUND

In this section, some preliminaries for fog task scheduling are explained.

A. FOG COMPUTING ARCHITECTURE

Fog computing is a decentralized structure close to end-user to process huge quantities of data. The three-layer architecture of the fog computing environment is shown in Figure 1. It is appropriate for IoT applications in which numerous distributed devices require collaboration, storage, and processing [12]. As shown in Figure 1, various devices such as smartphones are placed in the IoT device layer. The IoT devices layer gathers data from sensor devices and communicates with the fog layer. The fog layer contains routers, gateways, workstations, switches, and access points. This layer is close to the receiving layer and offers computing, networking, and storage services [13]. Finally, the cloud layer consists cloud servers [7], [14], [15].

B. TASK SCHEDULING IN CLOUD-FOG

By integrating cloud and fog, numerous methods are developed for efficient management of these platforms. These methods decrease the traffic load in the cloud servers by distributing the tasks. The network edge devices send their requests to the fog and cloud. The submitted requests are reviewed on the fog and sent to the fog or cloud depending on the task characteristics.

For example, in the context of Internet of Vehicles (IoV), task scheduling becomes even more critical due to the large number of connected vehicles generating vast amounts of data [16], [17]. Efficient scheduling methods can help reduce latency and ensure timely processing of tasks, leading to improved vehicle safety and traffic management.

Task scheduling in fog computing is a complex problem due to the dynamic nature of the fog environment, which is characterized by heterogeneous resources, varying workload, and mobility. Therefore, efficient techniques are needed for task scheduling and resource management in fog-cloud [18]. Several approaches have been proposed to address this problem, including heuristic-based, optimization-based, and machine learning-based methods.

Heuristic-based methods involve using rules or heuristics to assign tasks to fog nodes based on their characteristics, such as proximity, availability, and workload [6]. These methods are simple and efficient but may not always provide optimal solutions. Many researchers improved heuristic methods to achieve a better scheduling performance [19]. In some of these methods [11], [12], [20], [21], [22], [23] the IoT nodes are clustered at the network edge to process the requested task in parallel and decentralized. In these methods, the tasks are clustered according to their features such as the operating system type, task type, task priority, task execution, task arrival time, task resource, task data, and task data heterogeneity.

Optimization-based methods involve formulating the task scheduling problem as an optimization problem and finding the optimal solution using mathematical techniques such as linear programming or integer programming. These methods can provide optimal solutions but may be computationally expensive.

Machine learning-based methods involve using machine learning algorithms to learn the task allocation patterns from historical data and predict the optimal allocation for new tasks. These methods can adapt to the dynamic environment and provide efficient solutions.

In conclusion, task scheduling is a crucial aspect of fog computing, and it requires a comprehensive understanding of the fog environment and the characteristics of the tasks and resources. Various approaches have been proposed to address this problem, and further research is needed to develop efficient and scalable solutions. The related scheduling techniques in the fog environment are reviewed in Section IV with more details.

III. RESEARCH SELECTION METHOD

The SLR process involves the investigation of previously performed studies [24]. This part provides an SLR-based review for studying task scheduling techniques in fog computing.

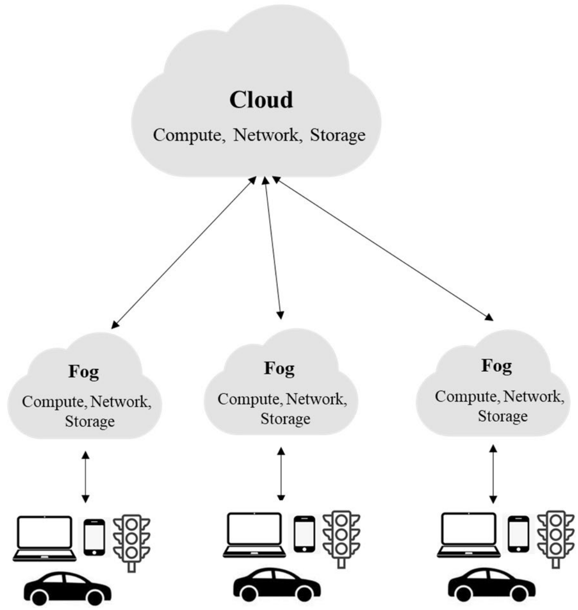


FIGURE 1. The three-layer fog computing architecture. [13]

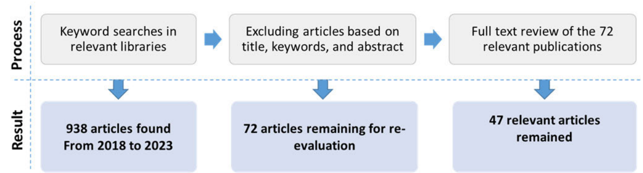


FIGURE 2. The selection criteria and evaluation of research studies.

Following the guidelines given in [25], six relevant databases are used for selecting the studies including ACM (https://dl.acm.org/), IEEE (https://ieeexplore.ieee.org/), ScienceDirect (https://www.sciencedirect.com/), Springer (https://link.springer.com/), Wiley (https://onlinelibrary.wiley.com/), and MDPI (https://www.mdpi.com/). For this purpose, a combination of keywords is constructed using the OR and AND Boolean operators to perform a search string as follows:

- (Fog OR Fog Computing) AND (Scheduling OR Task scheduling)

In the first step, the keywords search is performed on the mentioned databases, and due to a large number of search results, we filtered the studies and considered the ones published from the beginning of 2018. Then, the studies were filtered by their title and keywords. In the third step, the abstract section of the studies is examined. Lastly, from among all of the selected studies and based on a thorough review of their content, only those that were completely relevant to our topic are selected for further consideration. This process is illustrated in Figure 2.

The distribution of the selected articles by publication year and their publishers is shown in Figure 3. Elsevier has the largest number of articles with 13 articles and Wiley and MDPI have the lowest number of articles with 2 articles.

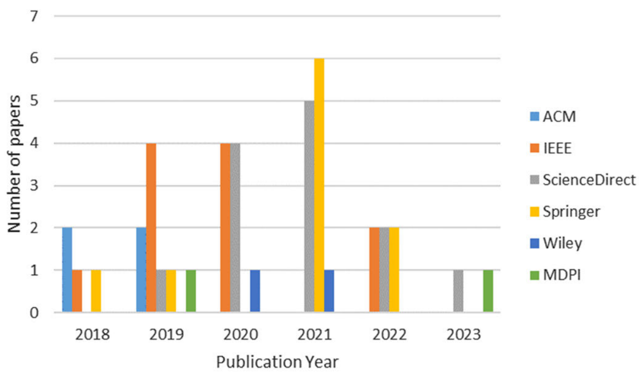


FIGURE 3. Distribution of research papers.

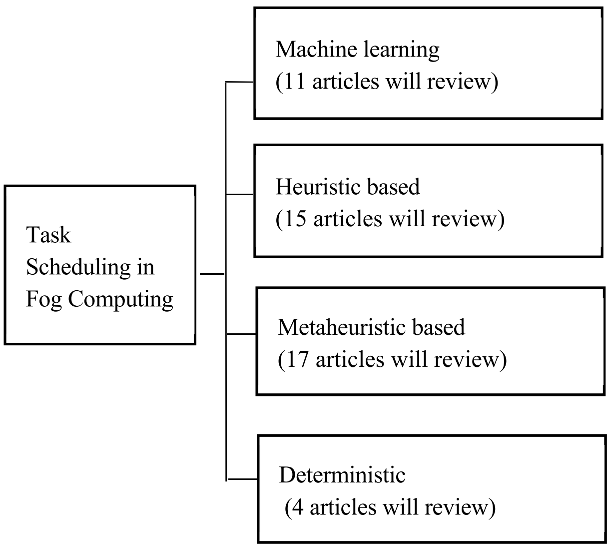


FIGURE 4. The classification of task scheduling techniques in fog computing.

Based on research purposes, this SLR survey tries to answer the following questions:

Q1. Which scheduling mechanism has attracted more attention?

Q2. Which performance features are addressed in the selected scheduling techniques?

Q3. What problems exist for future works?

IV. TASK SCHEDULING IN CLOUD-FOG MECHANISMS

As shown in Table 2, a total of 47 papers will be reviewed in this section, and their advantages and disadvantages will be discussed. Figure 4 shows the categorization of the task scheduling techniques in the fog environment, and classifies the articles studied in this paper within those categories. The proposed classification has four distinct categories including machine learning, heuristic-based, metaheuristic-based, and deterministic methods.

The papers are reviewed and compared based on qualitative parameters in the following.

A. AN OVERVIEW OF MACHINE LEARNING MECHANISMS

Traditional mechanisms cannot be used effectively in huge data sets [70], [71]. Machine learning mechanisms can be utilized in fog task scheduling to improve the efficiency and accuracy of the scheduling process. By analyzing historical data and identifying patterns, machine learning algorithms can predict future traffic loads and optimize task allocation accordingly. This can lead to reduced latency and improved system performance, as well as better resource utilization. Additionally, machine learning can be used to identify anomalies or potential failures in the system and take proactive measures to prevent them. Overall, incorporating machine learning mechanisms into fog task scheduling can enhance the effectiveness of the process and improve the overall performance of fog computing systems.

The selected machine learning-based techniques are presented in four categories in this part. Furthermore, the advantages and disadvantage of these methods are described and discussed.

1) ARTIFICIAL NEURAL NETWORK (ANN) BASED METHODS FOR TASK SCHEDULING

Arri, et al. [26] developed an optimized Task Group Aggregation (TGA) overflow handling system for fog computing environments using neural computing. The authors developed an ANN based algorithm to identify the overloaded servers and transfer the model's data to Virtual Machines (VMs) in fog computing environments. Various factors such as CPU, memory, and bandwidth are considered to balance VMs. In addition, the Artificial Bee Colony (ABC) algorithm is used to separate services and users based on their specific quality. The introduced ANN-based overflow handling algorithm enhances response time and success rate parameters compared to current approaches. However, energy consumption is not considered in this paper. Furthermore, considering the computational complexity of ANN algorithms, this work can be extended by optimizing the computational complexity of the models. Future work could also focus on incorporating energy efficiency considerations into the algorithm.

2) DEEP REINFORCEMENT LEARNING (DRL) BASED METHODS FOR TASK SCHEDULING

DRL-based methods offer a promising approach to improving task scheduling in fog computing environments, with the potential to reduce energy consumption and improve overall system performance. These methods learn from past experiences and make decisions based on real-time data, resulting in reduced latency. However, the potential disadvantage could be the complexity and computational cost of implementing the DRL algorithms, which may require significant resources.

For instance, Li et al. [27] used deep learning models for fog task scheduling in mobile crowd sensing applications. The authors enhance the efficiency of image data processing by deploying IoT devices as one of the multilayer structures

TABLE 2. Notable task scheduling mechanisms for fog computing.

Category	Authors	Year	Source Title	Title
Machine learning	Arri, et al. [26]	2021	Mathematics	Optimized Task Group Aggregation-Based Overflow Handling on Fog Computing Environment Using Neural Computing
	Li, et al. [27]	2019	ACM Transactions on Internet Technology (TOIT)	Deep reinforcement scheduling for mobile crowdsensing in fog computing
	Gazori, et al. [28]	2020	Future Generation Computer Systems	Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach
	Shadroo, et al. [29]	2021	Computer Networks	The two-phase scheduling based on deep learning in the Internet of Things
	Swarup, et al. [30]	2021	Procedia Computer Science	Energy Efficient Task Scheduling in Fog Environment using Deep Reinforcement Learning Approach
	Ghanavati, et al. [31]	2020	IEEE Transactions on Emerging Topics in Computing	Automata-based dynamic fault tolerant task scheduling approach in fog computing
	Nair and Bhanu [32]	2022	Journal of Scheduling	A reinforcement learning algorithm for rescheduling preempted tasks in fog nodes
	Jamil, et al. [33]	2023	Ad Hoc Networks	IRATS: A DRL-based intelligent priority and deadline-aware online resource allocation and task scheduling algorithm in a vehicular fog network
	Li, et al. [34]	2022	IEEE Internet of Things Journal	Deep-Graph-Based Reinforcement Learning for Joint Cruise Control and Task Offloading for Aerial Edge Internet of Things (EdgeIoT)
	Liu, et al. [35]	2023	Sensors	A Federated Learning Multi-Task Scheduling Mechanism Based on Trusted Computing Sandbox
	Shi, et al. [36]	2022	IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)	Federated Deep Reinforcement Learning-Based Task Allocation in Vehicular Fog Computing
Heuristic	Arisdakessian, et al. [37]	2020	IEEE/ACM Transactions on Networking	FoGMatch: an intelligent multi-criteria IoT-fog scheduling approach using game theory
	Azizi, et al. [38]	2022	Journal of Network and Computer Applications	Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach
	Hosseini, et al. [39]	2022	Computer Networks	Optimized task scheduling for cost-latency trade-off in mobile fog computing using fuzzy analytical hierarchy process
	Madhura, et al. [40]	2021	Computing	An improved list-based task scheduling algorithm for fog computing environment
	Choudhari, et al. [41]	2018	Proceedings of the ACMSE 2018 Conference	Prioritized task scheduling in fog computing
	Najafizadeh, et al. [42]	2021	Peer-to-Peer Networking and Applications	Privacy-preserving for the internet of things in multi-objective task scheduling in cloud-fog computing using goal programming approach
	Charântola, et al. [43]	2019	Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion	Component-based scheduling for fog computing

TABLE 2. ***(Continued.)*** **Notable task scheduling mechanisms for fog computing.**

Verma, et al. [44]	2021	Informatics in Medicine Unlocked	Rank based mobility-aware scheduling in Fog computing
Liu, et al. [45]	2018	IEEE Internet of Things Journal	DATS: Dispersive stable task scheduling in heterogeneous fog networks
Guo, et al. [46]	2019	IEEE Access	Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems
Xu, et al. [47]	2021	Procedia Computer Science	Adaptive scheduling strategy of fog computing tasks with different priority for intelligent production lines
Li, et al. [48]	2019	Sensors	Online workload allocation via fog-fog-cloud cooperation to reduce IoT task service delay
Li, et al. [49]	2019	IEEE Access	Resource allocation and task offloading for heterogeneous real-time tasks with uncertain duration time in a fog queueing system
Abdelmoneem, et al. [50]	2020	Computer Networks	Mobility-aware task scheduling in cloud-Fog IoT-based healthcare architectures
Aladwani [51]	2019	Procedia Computer Science	Scheduling IoT healthcare tasks in fog computing based on their importance
Aburukba, et al. [65]	2020	Transactions on Emerging Telecommunications Technologies	An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing
Rafique, et al. [18]	2019	IEEE Access	A novel bio-inspired hybrid algorithm (NBIHA) for efficient resource management in fog computing
Binh, et al. [53]	2018	Proceedings of the ninth international symposium on information and communication technology	An evolutionary algorithm for solving task scheduling problem in cloud-fog computing environment
Hosseinioun, et al. [11]	2020	Journal of Parallel and Distributed Computing	A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm
Abdel-Basset, et al. [54]	2020	IEEE Transactions on Industrial Informatics	Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications
Aburukba, et al. [55]	2021	Journal of Network and Computer Applications	A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices
Abdel-Basset, et al. [56]	2020	IEEE Internet of Things Journal	Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications
Tanha, et al. [57]	2021	Neural Computing and Applications	A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments
Boveiri, et al. [58]	2019	Journal of Ambient Intelligence and Humanized Computing	An efficient Swarm-Intelligence approach for task scheduling in cloud-based internet of things applications
Wang, et al. [59]	2018	Peer-to-Peer Networking and Applications	Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints
Najafizadeh, et al. [60]	2022	Cluster Computing	Multi-objective Task Scheduling in cloud-fog computing using goal programming approach
Movahedi and Defude [61]	2021	Journal of Cloud Computing	An efficient population-based multi-objective task scheduling approach in fog computing systems

TABLE 2. *(Continued.)* Notable task scheduling mechanisms for fog computing.

	Yadav, et al. [62]	2021	The Journal of Supercomputing	A bi-objective task scheduling approach in fog computing using hybrid fireworks algorithm
	Abdel-Basset, et al. [63]	2021	International Journal of Intelligent Systems	IEGA: an improved elitism-based genetic algorithm for task scheduling problem in fog computing
	Jia, et al. [64]	2021	Peer-to-Peer Networking and Applications	Energy and delay-ware massive task scheduling in fog-cloud computing system
	Aburukba, et al. [65]	2020	Future Generation Computer Systems	Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing
Deterministic	Kyung [66]	2021	Sensors	Prioritized Task Distribution Considering Opportunistic Fog Computing Nodes
	Tsai, et al. [67]	2021	Applied Sciences	An optimal task assignment strategy in cloud-fog computing environment
	Caminero and Muñoz-Mansilla [68]	2021	Sensors	Quality of service provision in fog computing: Network-aware scheduling of containers
	Razaque, et al. [69]	2022	Future Generation Computer Systems	Energy-efficient and secure mobile fog-based cloud for the Internet of Things

of a deep learning Convolutional Neural Network (CNN) model. The simulation outcomes revealed that the proposed solution minimizes the cost of bandwidth from the fog nodes to the cloud layer. Therefore, cloud layer computations are minimized. The presented approach focuses on optimizing energy consumption and reducing delay by dynamically adjusting the sampling rate of sensor data. It also enables proactive measures to prevent potential failures by predicting future resource availability and adjusting task scheduling accordingly. This can help meet execution deadlines and improve response times. However, when the scale of the system grows, the proposed model is unable to converge and the results are unstable.

Gazori et al. [28] also used a DRL based model for fog task scheduling in mobile crowd sensing applications. The authors proposed an efficient task scheduling algorithm based on double deep Q-learning model. The purpose of the proposed algorithm is to reduce service latency and measurement costs within the resources and time limit. Based on the given results in the paper, the introduced model outperforms simple techniques in terms of delay and tasks completion time. Furthermore, the single-point of failure problem and problems with fog load balancing are handled using the proposed algorithm. However, the algorithm converges slowly with uncertainty for high dimensional state-action space. However, the article does not provide specific information on how the DRL approach performs in terms of network bandwidth, and energy consumption criteria.

Furthermore, Shadroo et al. [29] proposed a two-phase scalable scheduling algorithm based on deep learning models to improve energy consumption in the context of IoT. In the first phase, the clustering method is used to identify the execution place of a task. Then, the task is scheduled according to the place of execution. In this work, three methods based on the Self-Organizing Map (SOM) are proposed for the clustering part. In the two proposed methods, the tasks received from the IoT layer are clustered based on their attributes using SOM and hierarchical SOM (H-SOM). In the third proposed method, Autoencoder is used to reduce the dimensions of attributes. According to the simulation results, the amount of missing tasks is reduced compared to SOM and H-SOM models. The clustering of IoT devices can reduce energy consumption. The proposed method is also scalable and can handle large numbers of IoT devices and fog nodes. However, their approach may require significant computational resources and time to train the model, which could impact the overall execution time of the scheduling process, and may not always meet the execution deadline requirements of some IoT applications, which could impact their performance and reliability. Depending on the complexity of the scheduling problem and the number of IoT devices and fog nodes involved, the proposed method also requires significant resource utilization, which could impact the performance of other applications running on the fog nodes. The scheduling process may also require significant network bandwidth to exchange information between IoT devices and fog nodes.

Swarup et al. [30] also proposed an algorithm called Clipped Double Deep Q-learning for IoT task scheduling in fog-based environments using deep reinforcement learning to address service latency and energy efficiency. Experience replay and the target networks techniques are used to develop the proposed model. The authors aim to improve energy efficiency, reduce costs, and minimize delays in task scheduling. The algorithm clusters IoT devices based on their resource requirements and schedules tasks for each cluster using deep reinforcement learning. A parallel queue is also used to ensure that there is no lag in optimal resource utilization. However, potential issues with execution time and network bandwidth may arise.

Ghanavati et al. [31] introduced a learning automaton algorithm for dynamic fault-tolerant task scheduling to optimize response time while reliable execution of tasks. The approach uses a state transition model to dynamically adjust the task scheduling based on the current system state and the availability of resources. The proposed approach also incorporates fault tolerance mechanisms to handle failures in the network or devices involved in the task execution. The advantages of the proposed approach include improved reliability and fault tolerance. The dynamic adjustment of task scheduling based on system state can also improve performance and reduce delays. However, the approach may require significant computational resources to maintain the state transition model and make dynamic adjustments to task scheduling. Additionally, the complexity of the model may make it difficult to implement and maintain in practice.

Nair and Bhanu [32] proposed an efficient algorithm for rescheduling preempted tasks in fog nodes. The introduced algorithm, Brain-Inspired Rescheduling Decision-making (BIRD), can complete the tasks in the expected time. The authors applied the actor-critic reinforcement learning model to mimic the decision-making model of the human brain. The introduced algorithm tries to achieve a rescheduling list that meets the deadline requirements and ensures optimal performance of the fog nodes through load balancing. The proposed BIRD algorithm is compared with other scheduling policies including First Come First Served (FCFS), greedy task allocation, task allocation based on least laxity, Shortest Job First (SJF), and Earliest Deadline First (EDF). Based on the given results, BIRD ensures the deadline requirement of the preempted task by load balancing and rescheduling the preempted tasks to fog nodes. The experimental evaluation shows that the execution time of the BIRD algorithm is expected to be faster than other scheduling policies. The BIRD algorithm ensures optimal performance of fog nodes through load balancing while rescheduling preempted tasks to fog nodes, thereby reducing delay caused by task preemption and optimizing resource utilization. However, the paper focuses solely on rescheduling preempted tasks in fog computing, which may limit its applicability to other areas of computer science or technology.

Overall, DRL methods provide a promising approach for fog task scheduling that can improve performance in terms of delay and energy consumption. However, implementing the DRL algorithms may pose a potential drawback due to their complexity and computational cost, which could demand considerable resources.

3) GRAPH-BASED DEEP LEARNING METHODS FOR TASK SCHEDULING

Graph Neural Networks are a type of neural network designed specifically to work with graph-structured data. This technique has gained significant attention in recent years due to its ability to address challenges in various domains, such as social network analysis and privacy-preserving machine learning. Recently, graph neural networks (GNNs) have emerged as a promising solution for optimizing task scheduling in fog computing. GNNs have emerged as a promising approach to task scheduling by leveraging the power of deep learning and graph theory.

Graph-Based Deep Learning Methods use graph-based models to represent the relationships between tasks and resources in the fog computing environment. These models can capture complex dependencies between tasks and resources, allowing for more accurate scheduling decisions. However, these methods can be computationally expensive and require large amounts of data to train.

For instance, Jamil et al. [33] proposed a new algorithm called IRATS for online resource allocation and task scheduling in a vehicular fog network. The algorithm is based on deep reinforcement learning and takes into account both priority and deadline constraints of tasks. The algorithm works by first predicting the future resource requirements of tasks using a Long Short-Term Memory (LSTM) network. Then, it uses a DRL agent to allocate resources to tasks based on their priority and deadline constraints, while also considering the current state of the network. The DRL agent learns from its past experiences to make better decisions in the future. The advantages of IRATS include its ability to handle dynamic changes in the network, its ability to prioritize tasks based on their importance, and its ability to meet task deadlines. However, one potential disadvantage of IRATS is that it requires significant computational resources to train the DRL agent. Additionally, the accuracy of the LSTM network's predictions may be affected by changes in the network environment.

Furthermore, Li et al. [34] developed a deep-graph-based reinforcement learning approach for joint cruise control and task offloading in aerial EdgeIoT systems. The proposed approach uses a graph neural network to model the complex relationships among the system components and optimize the joint control and offloading decisions. The advantages of the proposed approach include improved system performance, reduced energy consumption, and enhanced reliability. However, the approach requires significant computational resources and may suffer from scalability issues in large-scale systems. Overall, the paper presents a promising approach for optimizing EdgeIoT systems but further research is needed to address its limitations.

4) FEDERATED LEARNING BASED METHODS FOR TASK SCHEDULING

Federated learning is a distributed machine learning approach that allows multiple parties to collaboratively train a model without sharing their data. This technique leverages federated learning techniques to train machine learning models on decentralized data sources (e.g., edge devices) without compromising privacy or security. The main advantage of this approach is its ability to leverage distributed computing resources while preserving data privacy and security. However, it may also require careful design of communication protocols and aggregation methods to ensure convergence and accuracy.

Liu et al. [35] presented a federated learning-based multi-task scheduling mechanism for edge computing that utilizes trusted computing sandbox technology. The proposed mechanism aims to improve the efficiency and security of task scheduling in edge computing environments. The mechanism consists of three main components: a task scheduling module, a federated learning module, and a trusted computing sandbox module. The task scheduling module is responsible for allocating tasks to edge nodes based on their capabilities and workload. The federated learning module enables the collaborative training of machine learning models on distributed data without compromising data privacy. The trusted computing sandbox module ensures the security of the system by creating a secure execution environment for each task. The proposed mechanism has several advantages, including improved efficiency, reduced communication overhead, and enhanced security. However, it also has some limitations, such as the need for specialized hardware and software support for trusted computing. Overall, the proposed mechanism shows promise for improving the performance and security of task scheduling in edge computing environments.

Shi et al. [36] presented a federated deep reinforcement learning-based task allocation mechanism for vehicular fog computing. The mechanism employs a multi-agent reinforcement learning approach to allocate tasks to vehicles in a fog computing environment. The proposed mechanism aims to enhance the efficiency and reliability of task allocation in vehicular fog computing environments. The mechanism has several advantages, including improved resource utilization, reduced communication overhead, and enhanced reliability. However, it also has some limitations, such as the need for large amounts of data for training and the potential for privacy concerns. Overall, the proposed mechanism shows promise for improving the performance and reliability of task allocation in vehicular fog computing environments.

A summary of the reviewed techniques along with their major advantages and disadvantages is presented in Table 3.

B. AN OVERVIEW OF HEURISTIC BASED MECHANISMS

Heuristic-based mechanisms for fog task scheduling are methods that use rules of thumb or best practices to allocate tasks to devices in a fog computing environment. These mechanisms do not rely on machine learning techniques, but rather on simple decision-making rules that are based on experience or intuition. Heuristic-based mechanisms can be simple and easy to implement, but they may not always result in optimal task allocation [72]. They also do not adapt to changing conditions or learn from past experiences like machine learning-based approaches do. The following section reviews several heuristic-based fog scheduling methods that utilize various factors to achieve improved scheduling outcomes.

For instance, Arisdakessian et al. [37] presented FoGMatch, an intelligent multi-criteria IoT-Fog scheduling approach that uses game theory to allocate tasks to fog nodes. FoGMatch considers multiple factors such as energy consumption, delay, and resource availability in its decision-making process. The authors compare FoGMatch with other scheduling approaches and show that it outperforms them in terms of task completion time and energy consumption. The advantage of FoGMatch is its ability to adapt to changing conditions and learn from past experiences. It also considers multiple criteria in its decision-making process, which can lead to better task allocation. However, the disadvantage is that it relies on complex algorithms and may require more computational resources than heuristic-based mechanisms.

Furthermore,.Azizi et al. [38] formulated the task scheduling problem with the objective of reducing the overall energy consumption of fog nodes while meeting the various qualitative requirements of IoT tasks. The proposed model tries to minimize deadline violation time. The authors mapped IoT tasks to fog nodes using two presented semi-greedy based algorithms including Priority-aware Semi-Greedy (PSG) and PSG with Multi-start method (PSG-M). Based on the presented results, the proposed algorithm increases the percentage of tasks that meet their deadline, reduces the total time of deadline violation, and optimizes the energy consumption of fog resources and system makespan compared to existing algorithms. According to the time complexity analysis performed in the paper, this algorithm is a suitable solution for real-time scheduling of IoT tasks in fog computing systems.

In addition, Hosseini et al. [39] presented a dynamic scheduling algorithm based on the Priority Queue, Fuzzy and Analytical Hierarchy Process (PQFAHP). The proposed PQFAHP algorithm combines multiple priorities and can perform multi-criteria prioritization. The authors implemented dynamic scheduling based on various criteria including completion time, energy consumption, RAM, and deadline. The outcomes revealed that the proposed multi-criteria PQFAHP outperforms the benchmark algorithms in terms of waiting time, delay, service level, response time, number of scheduled tasks, and energy consumption.

Xu et al. [47] proposed an algorithm based on adaptive dynamic programming to find the best path for processing data with different priorities at fog nodes. The purpose of the proposed algorithm is to reduce time delay and energy consumption to perform priority-based tasks. The simulation results showed that the proposed algorithm increases efficiency and reliability. In addition, it also reduced power consumption.

Madhura et al. [40] introduced a novel list scheduling algorithm. The proposed algorithm is implemented in three steps: level sorting, task prioritization, and processor selection. First, in order to determine the level of task,

TABLE 3. Summary of the task scheduling in machine learning mechanisms.

Paper	Technique	Advantage	Weakness
Arri, et al. [26]	ANN-based overflow management algorithm	• Low response time	• High execution time • High energy consumption
Li, et al. [27]	DRL-based models for fog task scheduling in mobile crowd sensing applications	• Low energy consumption • Low delay • Low response time • Low execution deadlines	• High uncertainty • High cost • High resource utilization • High execution time
Gazori, et al. [28]	Double deep Q-learning model	• Low delay • Low response time • Low execution deadlines	• High cost • High execution time • High resource utilization • High uncertainty
Shadroo, et al. [29]	Two-phase scheduling algorithm using SOM	• Low energy consumption • High scalability	• High execution time • High resource utilization • High delay • High execution deadline • High response time • High cost • High network bandwidth
Swarup, et al. [30]	Deep reinforcement learning	• Low energy consumption • Low cost • Low delay • Low resource utilization	• High execution time • High network bandwidth
Ghanavati, et al. [31]	Learning automata	• Low response time • Low resource utilization • Low delay • Low uncertainty	• High resource utilization • High execution time
Nair and Bhanu [32]	Reinforcement learning method to schedule preempted tasks in fog computing	• Low response time • Low delay • Low execution time • Low resource utilization	• High cost
Jamil, et al. [33]	Graph-based deep learning model	• Low execution deadline	• High resource utilization • High uncertainty
Li, et al. [34]	Deep-graph-based reinforcement learning	• Low energy consumption • Low uncertainty	• High resource utilization • Low scalability
Liu, et al. [35]	Federated Learning	• High security • Low network bandwidth	• High cost
Shi, et al. [36]	Federated Learning	• Low delay • Low response time • Low resource utilization • Low network bandwidth • High security	• High cost

the dependence of tasks is determined. In the second step, tasks with more immediate successor tasks are assigned higher priority based on the accumulated execution cost, data transfer cost, and rank of predecessor task attributes. Finally, a noncrossover method is applied to processor selection. The proposed algorithm reduced the execution cost of tasks using a noncrossover method. Based on the given results, the proposed algorithm performs better than HEFT and PEFT algorithms. In addition, the results showed that the proposed algorithm has less time complexity. However, it increased the communication cost and decreased the performance.

Choudhari et al. [41] designed a priority-based task scheduling algorithm with the aim of enhancing fog network performance and reducing costs. The proposed algorithm allocates services to users based on priority levels. In the proposed method, if the centralized module detects that there are not enough resources in the fog nodes to perform a specific task, the task is sent to the cloud after calculating the essential monetary costs. The applications are prioritized based on user expectations and service deadlines to optimize execution time and cost of tasks.

Najafizadeh et al. [42] also proposed a privacy-driven architecture for task scheduling in IoT. The authors proposed a multi-objective algorithm to reduce service time and cost. Therefore, the best solutions are determined using the Goal Programming Approach (GPA). Experimental results revealed that the proposed algorithm improves the performance and convergence speed compared to other multi-objective algorithms while considering the privacy requirements of IoT devices.

Charântola et al. [43] presented a scheduling algorithm considering the application delay requirement in a fog-cloud environment. In the proposed approach, each application is considered as a set of modules that communicate with each other to complete a task. Therefore, modules that are far from each other cause a delay in the completion time. The results of the article show that the presented algorithm is an efficient algorithm for scheduling real-time applications in multi-tiered cloudlet infrastructures.

Verma et al. [44] introduced a Rank-based Mobility-aware Scheduling (RMS) technique that uses contextual information to rank resources. MobFogSim simulation tool is used to implement the proposed RMS. The results show that RMS performs better than Distance-based Mobility-aware Scheduling (DMS) in terms of migration time, delay, downtime, tuple lost, and execution time.

Liu et al. [45] presented a decentralized algorithm called Diffractive Stable Task Scheduling (DATS) to minimize activity delay in heterogeneous fog nodes. The proposed DATS has two main elements: (1) PE-based progressive computing resources competition, and (2) the quality of experience-based synchronized task scheduling. The experimental results showed that the proposed algorithm reduces the service delay in dissimilar fog nodes by obtaining a suitable trade-off between computation resources and relationship capabilities. Moreover, the proposed DATS algorithm provides low complexity and ensures system stability. However, energy consumption is not considered in this work.

Guo et al. [46] proposed an energy-efficient and delay-guaranteed workload allocation scheme for IoT-edge-cloud computing systems. The proposed scheme considers the characteristics of IoT devices, edge servers, and cloud servers to allocate tasks to the most suitable computing resources. The scheme uses a two-stage optimization approach that first allocates tasks to edge servers based on their processing capabilities and then assigns remaining tasks to cloud servers. The proposed scheme also considers the energy consumption of computing resources and aims to minimize it while ensuring delay guarantees for tasks. The advantages of the proposed scheme include its ability to handle dynamic workload changes, its consideration of energy consumption, and its ability to provide delay guarantees for tasks. The scheme also reduces the communication overhead between IoT devices and cloud servers by allocating tasks to nearby edge servers. The disadvantages of the proposed scheme include its reliance on accurate task execution time estimation and its complexity due to the two-stage optimization approach. Additionally, the proposed scheme may not be suitable for scenarios where edge servers have limited processing capabilities or when there are a large number of IoT devices with varying task requirements.

Various dynamic programming approaches have been proposed in the literature to optimize fog scheduling. Xu et al. [47] proposed an algorithm based on adaptive dynamic programming to find the best path for processing data with different priorities at fog nodes. The purpose of the proposed algorithm is to reduce time delay and energy consumption to perform priority-based tasks. The simulation results showed that the proposed algorithm increases efficiency and reliability. In addition, it also reduced power consumption. In addition, Li, et al. [48] proposed a novel approach for workload allocation in the IoT environment. The proposed approach involves the cooperation between fog nodes and cloud servers to allocate workloads efficiently and reduce task service delay. The presented workload allocation algorithm considers the workload characteristics, resource availability, and communication latency between fog nodes and cloud servers. The algorithm uses a dynamic programming technique to optimize the workload allocation and minimize the task service delay. The advantages of the proposed approach include reduced task service delay, improved resource utilization, and increased scalability. The fog-fog-cloud cooperation enables efficient workload allocation and reduces the burden on individual fog nodes, leading to improved performance. However, the proposed approach requires a high level of coordination between fog nodes and cloud servers, which may increase communication overhead. Additionally, the dynamic programming technique used in the algorithm may be computationally expensive for large-scale IoT environments. Finally, Li, et al. [49] presented a resource allocation and task offloading scheme for heterogeneous real-time tasks with uncertain duration time in a fog queueing system. The scheme aims to minimize the total cost of task execution while meeting the deadline constraints of each task. The proposed scheme uses a dynamic programming approach to determine the optimal resource allocation and task offloading decisions. The scheme also takes into account the uncertainty in task duration time by using a probabilistic model. The advantages of the proposed scheme include its ability to handle heterogeneous tasks with uncertain duration time, its ability to meet deadline constraints, and its optimization of resource allocation and task offloading decisions. However, it requires significant computational resources to determine the optimal decisions, which may not be feasible in some scenarios.

The scheduling problem in healthcare IoT systems is complex and dynamic, with multiple criteria and constraints that need to be considered simultaneously. Therefore, several task scheduling algorithms have been proposed for healthcare IoT systems. Abdelmoneem et al. [50] introduced an effective mobility-aware heuristic-based task scheduling algorithm for cloud-Fog IoT-based healthcare architectures. The proposed method dynamically balances the distribution of tasks according to the patients' movements and the spatial/temporal residual of their sensed data. It minimized the total scheduling time during the ranking and reallocation processes, using the critical level and maximum task response time. The obtained results are compared with other common solutions. The results showed that the presented method reduces the percentage of makespan and energy consumption. However, the algorithm may require more computational resources, which can be a disadvantage in resource-constrained environments. In addition, .Aladwani [51] proposed a novel algorithm called Task Classification and Virtual Machine Categorization (TCVC) for medical data scheduling. The presented method uses the MAX-MIN algorithm to reduce the waiting time in the queue. The proposed TCVC enhanced the execution of IoT social insurance planning in the conditions of fog registration, depending on the importance of task. The results presented in the paper confirm the improvement in total execution time, total completion time, and total waiting time. However, energy consumption is not considered in the proposed algorithm.

C. AN OVERVIEW OF METAHEURISTIC BASED MECHANISMS

In meta-heuristic algorithms, a random solution space is used for task scheduling. With a few changes in meta-heuristic algorithms, they can be used to solve various optimization problems [73]. The meta-heuristic algorithms are problem independent [74]. Fog task scheduling using meta-heuristic algorithms has shown promise in improving resource utilization and reducing latency in distributed fog computing environments. In this part, some studies are reviewed on meta-heuristic task scheduling in the fog environment.

Xu et al. [75] proposed a task scheduling approach based on Laxity-Based Priority and Ant Colony System (LBP-ACS) in cloud-fog environment. The LBP algorithm is applied to obtain priority of tasks. Furthermore, Constrained Optimization Algorithm based on the ACS (COA-ACS) is used for task scheduling. Simulation results showed that the proposed algorithm performs better than Greedy for Energy (GfE), Heterogeneous End Time (HEFT) and hybrid ant colony optimization with differential evolution algorithms in terms of reducing energy consumption and failure rate in scheduling dependent tasks with mixed deadlines. However, the authors considered only associated tasks and not independent tasks.

Ghobaei-Arani et al. [52] introduced a Moth-Flame Optimization (TS-MFO) algorithm for scheduling tasks in the fog environment. Thea main goal of the presented method is to meet the QoS requirements of Cyber Physical System (CPS) applications. The introduced TS-MFO algorithm can found the best solutions to locate the fog nodes. Using the iFogSim simulator, TS-MFO is compared with Particle Swarm Optimization (PSO), Non-dominated Sorting Genetic Algorithm-II (NSGA-II), and Bee Life Algorithm (BLA). The results revealed that the TS-MFO algorithm reduces the total execution time of the tasks. However, the authors ignore the energy consumption and communication cost.

A Novel Bio-Inspired Hybrid Algorithm (NBIHA) is proposed by .Rafique et al. [18] for efficient task scheduling and resources management in the fog environment. The proposed NBIHA is a combination of Modified Particle Swarm Optimization (MPSO) and Modified Cat Swarm Optimization (MCSO) [76]. In this algorithm, the best match of fog devices for an input task is discovered based on the requested memory and CPU time. Compared with the three approaches First Come First Serve (FCFS) [77], Shortest Job First (SJF) and MPSO, the NBIHA algorithm provides better results in terms of execution time, energy consumption and average response time. However, this work does not address the communication cost.

Binh et al. [53] introduced a genetic scheduling algorithm called Time–Cost aware Scheduling (TCaS) in IoT-Fog-cloud based on computation time and operational cost. The proposed TCaS considers the trade-off between different criteria including time, cost, and user satisfaction. In this method, a three-layer model including client layer, fog layer, and cloud layer is used for resource allocation. The algorithm tries to assign tasks to the client and fog layers, and the rest of the requirements are met through the cloud layer. The performance of TCaS is evaluated with different task sets. Various factors including total response time, processing time and data center cost are measured. However, TCaS allocates resources before processing and does not consider runtime allocation of resources. Furthermore, deadline is not considered in the proposed approach.

Hosseinioun et al. [11] proposed an efficient hybrid algorithm based on invasive weed optimization and culture algorithms. The purpose of the proposed algorithm is to minimize energy consumption based on Dynamic Voltage and Frequency Scaling (DVFS). The presented results confirmed that the proposed DVFS algorithm meets the real-time requirement in heterogeneous systems.

Abdel-Basset et al. [54] proposed three algorithms based on heuristic marine predators to tackle the task scheduling problem in a fog-cloud environment. In addition to the standard MPA, the authors introduced two other versions: modified MPA (MMPA), and improved MMPA. In MMPA, the exploitation capability of the MPA algorithm is improved based on the latest updated positions rather than the last best one. Furthermore, the authors improved MMPA toward the best approach with mutation and ranking strategy based reinitialization. The proposed algorithms are compared with other meta-heuristic algorithms and genetic algorithms in

TABLE 4. Summary of the task scheduling in heuristic-based mechanisms.

Paper	Technique	Advantage	Weakness
Arisdakessian, et al. [37]	An intelligent scheduling algorithm based on game theory	• Low energy consumption • Low execution deadline	• High resource utilization
Azizi, et al. [38]	Semi-greedy based algorithms	• Low energy consumption • Low execution deadline • Low complexity	• -
Hosseini, et al. [39]	Fuzzy and analytical hierarchy process	• Low delay • Low response time • Low energy consumption	• High resource utilization
Madhura, et al. [40]	List scheduling algorithm	• Low resource utilization • Low response time • Low complexity	• High cost
Choudhari, et al. [41]	Priority-based task scheduling algorithm	• Low execution time • Low cost	• High delay
Najafizadeh, et al. [42]	Goal programming approach	• Low execution time • Low cost	• -
Charântola, et al. [43]	Module based communication	• Low execution time • Low response time • Low execution deadline • Low delay	• Low network bandwidth
Verma, et al. [44]	Rank-based Mobility-aware Scheduling	• Low delay • Low execution time	• -
Liu, et al. [45]	Matching theory	• Low delay • Low uncertainty • Low complexity	• High energy consumption
Guo, et al. [46]	An energy-efficient and delay-guaranteed workload allocation scheme	• Low energy consumption • Low delay • Low network bandwidth	• High execution time
Xu, et al. [47]	Dynamic programming	• Low energy consumption • Low execution time • Low cost	• High resource utilization
Li, et al. [48]		• Low delay • Low resource utilization • High scalability	• High network bandwidth • High execution time
Li, et al. [49]		• Low execution deadline	• High resource utilization
Abdelmoneem, et al. [50]	A mobility aware heuristic-based scheduling and allocation approach	• Low energy consumption • Low response time	• High resource utilization
Aladwani [51]	MAX-MIN algorithm	• Low execution time • Low execution deadline • Low delay	• High resource utilization • High energy consumption

terms of energy consumption, makespan, flow time and carbon dioxide emission rate. The obtained outcomes showed that the improved MMPA has a better performance compared to other algorithms.

Aburukba et al. [55] presented a heuristic-based scheduling approach using Genetic Algorithm (GA) to meet the maximum requests amount considering their deadlines. The architecture of the proposed method is based on the three-layer fog computing architecture. To ensure QoS delivery, deadline misses are minimized using a mixed integer programming optimization model. The simulation result confirmed that the deadline misses of the proposed method id 20% to 55% better than round robin and priority scheduling techniques. The proposed technique provides a near-optimal solution in reasonable computational time.

Abdel-Basset et al. [56] proposed a Harris Hawks Optimization algorithm based on a Local Search strategy (HHOLS) for scheduling energy-aware tasks in a fog environment to meet QoS requirements in the Industrial IoT (IIoT). In this paper, the proposed HHOLS is compared with other meta-heuristic methods based on several criteria such as energy consumption, makespan, cost, flow time, and carbon dioxide emission rate. Based on the given results, the proposed HHOLS algorithm performs better compared to other algorithms.

Tanha et al. [57] presented a new hybrid algorithm based on a combination of Genetic And Thermodynamic Simulated Annealing Algorithm (GATSA) to solve workflow scheduling in a cloud environment with regard to makespan minimization. The proposed GATSA algorithm uses thermodynamic laws to reduce the temperature. Experiments conducted under different conditions confirmed the qualitative performance of GATSA against other counterparts in terms of evaluation criteria. However, GATSA requires more time to execute than others.

Boveiri et al. [58] proposed an efficient variation of ant colony optimization algorithm named Max–Min Ant System (MMAS) for optimal task-graph scheduling in IoT applications. The proposed system uses ant colony optimization to schedule tasks in multiprocessor systems. The main purpose of this work is to identify the priority of the tasks to assign them to the appropriate systems. The proposed system is evaluated using different random task graphs with varied shape parameters. The outcomes revealed the effectiveness and superiority of the presented MMAS system against its traditional counterparts in terms of performance. However, the proposed approach does not address energy consumption.

Wang et al. [59] proposed a Cooperative Multi-Task Scheduling based on Ant Colony Optimization algorithm (CMSACO). The purpose of the proposed work is to improve task profit, task deadline, task dependency, node heterogeneity and load balance. Based on the presented results, the proposed algorithm executes the offloaded tasks in fog devices by improving the delay factor, finish time and energy consumption.

Najafizadeh et al. [60] proposed a Multi-Objective Simulated Annealing (MOSA) algorithm for secure distribution of tasks on cloud and fog nodes based on deadline limits. Also, the Goal Programming Approach (GPA) is used to identify a solution that meets multiple goals. In this method, access level and scheduling based on client request goals are considered for assigning IoT tasks between fog and cloud nodes. The proposed method is compared against Multi-Objective Particle Swarm Optimization (MOPSO), Multi-Objective Tabu Search (MOTS), and Multi-Objective Moth-Flame Optimization (MOMF). The experimental results revealed that the proposed algorithm has achieved better performance in terms of service delay time, access level control and deadline. In addition, it has achieved satisfactory results in terms of service cost.

Movahedi and Defude [61] formulated the task scheduling problem as an Integer Linear Programming (ILP) optimization model to optimize time and energy consumption in the fog environment. Furthermore, an enhanced Whale Optimization Algorithm (WOA) algorithm called Opposition-based Chaotic Whale Optimization Algorithm (OppoCWOA) is presented to solve the modeled task scheduling problem. The efficiency of the proposed OppoCWOA is proved compared to the original WOA, ABC, PSO, and GA algorithms in terms of convergence speed and accuracy in achieving time-energy balance.

Yadav et al. [62] proposed a hybrid scheduling algorithm for the fog environment. The presented algorithm is a combination of Heterogeneous Earliest Finish Time First (HEFT) and fireworks algorithm (FWA) for bidirectional optimization. The main purpose of this work is to reduce makespan and cost.

Abdel-Basset et al. [63] introduced an Improved Elitism Genetic Algorithm (IEGA) to solve the fog task scheduling problem. In the proposed IEGA, mutation and crossover rates are modified to explore more combinations that may constitute near-optimal permutation. In addition, in order to avoid getting stuck in local minima and identify a better solution, the proposed algorithm mutates a number of solutions based on a certain probability. The results showed that the proposed IEGA performs better than recent robust optimization algorithms in terms of makespan, flow time, fitness function, carbon dioxide emission rate, and energy consumption.

Jia et al. [64] formulated a low-complexity Pareto optimization-based model for task scheduling, where queuing models are provided for delay estimation and energy consumption models for heterogeneous resources. In the proposed model, a set of non-dominated solutions is obtained by local search procedures. Then, the non-dominated solutions are improved using a tree-based local search method. The introduced method is compared with four classical algorithms for similar problems. The experimental results demonstrated the efficacy and robustness of the proposed algorithm.

Aburukba et al. [65] presented an efficient GA-based algorithm for scheduling IoT requests. The outcomes showed that the introduced algorithm is superior in terms of overall delay compared to Waited Fair Queuing (WFQ), Priority-Strict Queuing (PSQ), and RR techniques. Furthermore, the proposed algorithm improves meeting the requests deadlines. Moreover, this work can be extended with various functions to maximize the utilization of resources and minimize latency.

D. AN OVERVIEW OF DETERMINISTIC MECHANISMS

Deterministic mechanisms always produce the same output for a given input. Deterministic mechanisms for fog task scheduling involve using a predefined set of rules and algorithms to allocate tasks to fog nodes. Exhaustive search is an example of a deterministic approach to the task scheduling problem. The exhaustive algorithms enumerate the entire search space to find the optimal plan based on the given cost model. These mechanisms do not involve any randomness or probabilistic methods. Deterministic mechanisms for fog task scheduling are simple and easy to implement. However, they may not always result in optimal resource utilization or latency reduction. The fog task scheduling techniques regarding the deterministic mechanisms are discussed in this part.

Kyung [66] presented a prioritized task distribution model considering static and opportunistic fog nodes with respect to their mobility. In this model, it is possible to process delay-sensitive tasks in static fog nodes and delay in-sensitive tasks in opportunistic fog nodes. Based on the results presented in this paper, the proposed model performs better than other traditional models in terms of service response delay and outage probability. However, there are some potential disadvantages to this approach. The ranking of fog nodes may not always accurately reflect their actual computing capabilities or workload. Additionally, the prioritization of tasks may result in lower-priority tasks being delayed or neglected, which could impact overall system performance.

Tsai et al. [67] proposed a linear transformation model to solve the nonlinear task assignment problem in cloud-fog systems. Various criteria including execution time and operating costs are considered for optimal task allocation. The proposed model can find the optimal solution based on task requirements, nodes' processing speed, and nodes' resource usage cost. Although the proposed deterministic approach can identify an optimal solution, the computational complexity of the approach grows rapidly as the problem size increases.

Caminero and Muñoz-Mansilla [68] proposed a network-aware scheduling algorithm to select the appropriate fog node to execute an application within a given deadline. The status of the network is considered in the introduced scheduling technique. The proposed algorithm is an extension to the Kubernetes default scheduler. The results revealed that the presented algorithm can execute all the submitted tasks within the deadline and reach the optimal solution. Furthermore, Yin, et al. [78] is also presented a container-based task scheduling algorithms in Fog computing. The obtained outcomes by the authors revealed that the proposed algorithms improve resource utilization by minimizing delay.

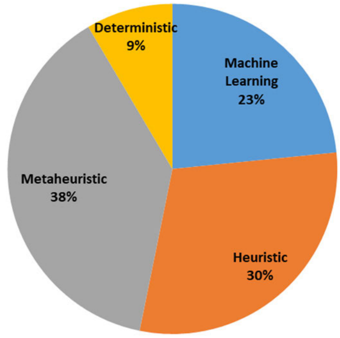


FIGURE 5. Various types of fog scheduling techniques used in the selected articles.

Razaque et al. [69] presented an efficient hybrid algorithm to decrease energy consumption for the Mobile Fog-Based Cloud (MFBC). In the proposed algorithm, the voltage scaling factor is used to reduce energy consumption. The identity of mobile cloud users is also secured using block chain technology. The proposed algorithm includes job-scheduling and machine power calculation modules to allocate tasks for mobile fog cloud users in a time-series fashion to improve the throughput and latency. The presented approach also proposed a secure key management scheme that ensures secure communication between the fog nodes and the cloud. The performance of the proposed algorithm is significantly better than the state-of-the-art algorithms in terms of security, energy efficiency, throughput and latency. However, there are some disadvantages to consider. One potential disadvantage is the complexity of the architecture, which may require significant resources to implement.

V. RESULTS AND COMPARISONS

The selected task scheduling mechanisms in the fog environment are investigated in the previous section. The most important fog scheduling mechanisms until 2023 are presented. Based on the reviewed articles, machine learning, heuristic-based, metaheuristic-based, and deterministic mechanisms are the four main categories of task scheduling mechanisms in fog-cloud. Figure 5 shows the popularity of various techniques in the fog task scheduling, which clearly shows that metaheuristic and heuristic are the most popular ones.

The reason why metaheuristic and heuristic methods are more popular in fog-based task scheduling papers compared to deterministic and machine learning methods could be due to several factors.

Firstly, metaheuristic and heuristic methods are well-suited for solving complex optimization problems that involve multiple objectives and constraints, which is common in fog-based task scheduling. These methods are based on iterative search algorithms that explore the solution space to find the best solution. This approach is particularly useful when the

TABLE 5. Summary of the task scheduling in metaheuristic-based mechanisms.

Paper	Technique	Advantage	Weakness
Xu, et al. [75]	Laxity-Based Priority and Ant Colony System	• Low energy consumption	• High execution time
Ghobaei-Arani, et al. [52]	Moth-Flame optimization algorithm	• Low execution time	• High energy consumption • High cost
Rafique, et al. [18]	Bio-Inspired hybrid algorithm	• Low execution time • Low energy consumption • Low response time	• High cost
Binh, et al. [53]	Genetic-based scheduling algorithm	• Low response time • Low execution time • Low cost	• High execution deadline
Hosseinioun, et al. [11]	Invasive weed optimization and culture algorithms	• Low energy consumption • Low execution time	• -
Abdel-Basset, et al. [54]	Marine predators-based algorithms	• Low energy consumption • Low execution deadline	• -
Aburukba, et al. [55]	Heuristic-based scheduling approach using the genetic algorithm	• Low execution deadline • Low complexity	• -
Abdel-Basset, et al. [56]	Harris Hawks optimization algorithm based on a local search strategy	• Low energy consumption • Low cost	• High execution time
Tanha, et al. [57]	Hybrid genetic and thermodynamic simulated annealing algorithm	• Low energy consumption • Low execution deadline	• High execution time • High complexity
Boveiri, et al. [58]	Max–Min Ant System	• Low response time • Low cost	• High energy consumption
Wang, et al. [59]	Cooperative Multi-Task Scheduling based on Ant Colony Optimization algorithm	• Low execution deadline • Low energy consumption • Low execution time	• -
Najafizadeh, et al. [60]	Multi-objective simulated annealing	• Low delay • Low execution deadline • Low cost	• High execution time
Movahedi and Defude [61]	Opposition-based Chaotic Whale Optimization Algorithm	• Low energy consumption	• High cost

TABLE 5. *(Continued.)* **Summary of the task scheduling in metaheuristic-based mechanisms.**

		• Low execution time • Low uncertainty	
Yadav, et al. [62]	Hybrid Heterogeneous Earliest Finish Time First (HEFT) and fireworks algorithm (FWA)	• Low cost • Low execution deadline	• High execution time
Abdel-Basset, et al. [63]	Enhanced elitism genetic algorithm	• Low execution deadline • Low energy consumption	• -
Jia, et al. [64]	Pareto-optimization-based Massive Task Scheduling Framework	• Low delay • Low energy consumption • Low execution time • Low cost • Low complexity	• -
Aburukba, et al. [65]	A heuristic algorithm based on a GA algorithm	• Low execution deadline • Low delay • Low resource utilization	• High execution time

TABLE 6. **Summary of the task scheduling in deterministic mechanisms.**

Paper	Technique	Advantage	Weakness
Kyung [66]	Prioritized task distribution model	• Low response time • Low uncertainty	• High complexity
Tsai, et al. [67]	Linear transformation model	• Low delay	• High execution time • High resource utilization
Caminero and Muñoz-Mansilla [68]	Bio-Inspired hybrid algorithm	• Low execution deadline • Low delay	• High complexity
Razaque, et al. [69]	Prioritized task distribution model	• Low energy consumption • Low execution time • Low delay • High security	• High resource utilization

problem is NP-hard or when the solution space is large and complex.

Secondly, metaheuristic and heuristic methods are adaptable and flexible, which makes them suitable for dynamic environments such as fog computing. These methods can quickly adapt to changes in the environment and adjust the scheduling decisions accordingly. In contrast, deterministic methods are typically based on predefined rules and may not be able to handle unexpected events or changes in the environment.

Thirdly, in the cloud-fog environment, the preference is to find suboptimal solution, but in short period of time. The metaheuristic-based methods employed nature inspired algorithms to find a near optimal solution within reasonable time. The main advantage of metaheuristic-based techniques is their simplicity of development. Henceforth, metaheuristics

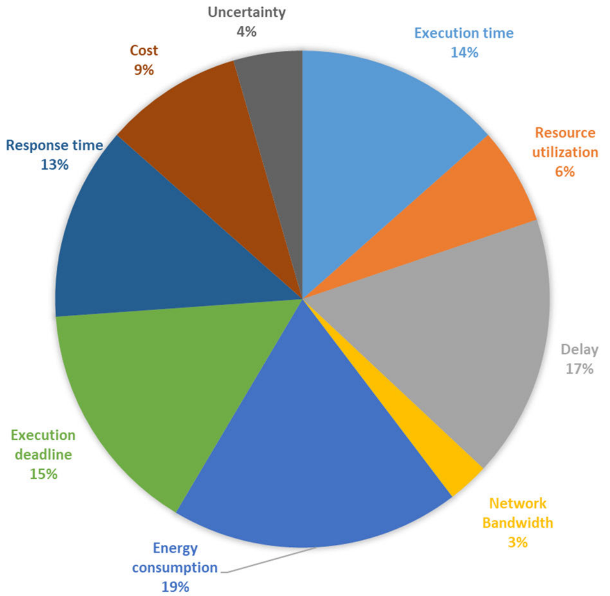


FIGURE 6. Various parameters improved in the selected articles.

can often find good solutions with less computational effort to solve large size and complex problems.

Fourthly, machine learning methods require large amounts of training data and may not be suitable for real-time applications with strict timing requirements. In addition, machine learning models may not be able to generalize well to new and unseen situations, which is a common challenge in fog computing environments.

Overall, the popularity of metaheuristic and heuristic methods in fog-based task scheduling papers can be attributed to their ability to handle complex optimization problems, adaptability to dynamic environments, and suitability for real-time applications. However, it is important to note that each method has its strengths and weaknesses, and the choice of method depends on the specific requirements of the application.

From the compared result (Figure 6), the energy consumption with 19% is the most used in scheduling algorithms. Moreover, delay, execution deadline, and response time are very important parameters in the scheduling algorithms of fog computing.

As per our review, it is evident that a significant number of scholars have focused on reducing energy consumption in fog computing. This is primarily due to the fact that fog computing devices are often resource-constrained and operate on limited battery power. Therefore, optimizing energy consumption is critical to ensure the efficient and effective functioning of these devices. In our comprehensive review of existing literature, we found that several studies have proposed various mechanisms to reduce energy consumption in fog-based task scheduling. Overall, it is clear that energy consumption is a critical parameter that needs to be considered when designing task scheduling mechanisms for fog computing. By incorporating energy-efficient mechanisms, we can ensure that fog computing devices operate optimally while conserving energy and extending their battery life.

The reason why delay, execution deadline, and response time are most attended in fog-based task scheduling papers is because these factors are directly related to the performance of the system and the satisfaction of the end-users. Execution deadline refers to the maximum time allowed for a task to be completed, which is essential for applications with strict timing requirements.

On the other hand, factors such as network bandwidth, uncertainty, and resource utilization are relatively less frequently studied in the literature because they are indirectly related to the performance of the system and may not have a significant impact on the end-users' satisfaction.

However, it is important to note that all these factors are interrelated, and optimizing one factor may affect the others. Therefore, it is necessary to consider all these factors when designing a fog-based task scheduling mechanism to ensure optimal performance and energy efficiency. Based on the reviewed articles and the criteria included in them, it is possible to consider the exchange between different criteria. For instance, to increase the availability of resources at an acceptable cost, the trade-off between delay and cost can be considered. As another example, the trade-off between resource usage and energy consumption can be used to increase network throughput and bandwidth and thus increase network robustness.

VI. OPEN ISSUES AND CHALLENGES

This section provides some challenges for fog-based task scheduling from various perspectives: (1) security, (2) energy management, (3) resource management, (4) fault tolerance, (5) heterogeneity, and (6) task scheduling mechanisms.

- Security

One of the critical challenges in the domain of fog computing is ensuring the security of task scheduling in a distributed environment. Researchers have identified that the presence of multiple edge devices with diverse computing power, network bandwidth, and varying security levels can be a potential threat to the security of task scheduling. To mitigate these security risks, various security mechanisms have been proposed to ensure the confidentiality, integrity, and availability of data in fog computing systems [35], [69]. However, these proposed solutions need to be tested and evaluated comprehensively to ensure their effectiveness and efficiency. Therefore, further research is required to address the security challenges of fog task scheduling and to develop more robust and trustworthy security solutions to provide secure and reliable task scheduling in fog computing.

- Energy management

One of the key challenges in fog computing is managing energy consumption, particularly in relation to the scheduling of tasks. The fog environment consists of a large number of distributed nodes. Therefore, the energy consumption in the fog nodes is higher compared to the cloud environment. In this regard, it is necessary to develop energy-efficient storage protocols and fog computing architectures in the fog environment. For example, some researchers [79] have developed load balancing in diverse fog nodes to enhance energy consumption. Fog computing can reduce the energy consumption of devices by offloading computation tasks to nearby fog nodes. The fog nodes should be selected based on their energy efficiency to reduce the overall energy consumption of the system. Energy management is essential for extending the lifespan of fog devices, reducing operational costs and improving the overall performance of the fog computing network.

However, there are still unresolved issues concerning energy management in task scheduling algorithms for fog computing. Improvements in the energy efficiency of task scheduling mechanisms in fog computing will promote more sustainable and cost-effective solutions for the growing field of IoT devices, improving performance and reducing environmental impacts.

- Resource Management

Fog nodes have a high workload, while they do not have enough computing power and storage. Fog computing involves heterogeneous resources that vary in terms of processing power, storage capacity, and communication bandwidth. Therefore, one of the requirements in the fog environment is the effective utilization and management of fog node resources. The fog nodes should be selected based on their available resources to ensure efficient resource utilization.

In addition, the mobility of end devices changes dynamically with the metrics such as bandwidth, storage, computations, and latency. Therefore, various resource management and scheduling techniques such as dynamic modules, placement techniques, migration and consolidation of edge devices are proposed for optimal resource utilization. However, these techniques still need to be developed and explored.

- Fault tolerance

Fog computing involves mobile nodes that can fail or become unavailable due to various reasons such as network congestion, hardware failure, or power outage. Therefore, fault tolerance is an essential issue that needs to be addressed in task scheduling. Fault tolerance in fog task scheduling is an essential aspect of ensuring the reliability and effectiveness of fog computing systems. The increasing trend towards deploying complex and resource-intensive applications on fog computing platforms necessitates the need for a fault-tolerant approach to scheduling tasks. Although various fault-tolerant mechanisms have been proposed, they still face numerous challenges in effectively handling faults and failures in the fog environment. Therefore, the need for further investigation and development of more efficient and reliable fault-tolerant mechanisms for fog task scheduling remains a vital open issue in the field of fog computing. Addressing this issue will not only enhance the performance and reliability of fog computing but also enable the adoption of more advanced fog computing platforms in various critical applications.

- Heterogeneity

One important issue related to fog computing is the heterogeneity in task scheduling. Fog nodes have varying capabilities and resources, therefore the scheduling of tasks must take into account the heterogeneity of these nodes. This issue is especially important in real-time applications, where tasks need to be processed as quickly and efficiently as possible. One solution to this problem is to use machine learning techniques to predict the most suitable fog nodes for specific tasks. Another approach is to divide tasks into smaller sub-tasks that can be executed by different fog nodes with appropriate resources. It is important to continue researching and developing effective strategies to manage the heterogeneity in fog task scheduling in order to maximize the benefits of fog computing in various application domains.

- Task scheduling mechanisms

Task scheduling in fog computing is a complex problem that requires efficient and scalable solutions. Therefore, task scheduling mechanisms are an open issue that needs further research and development. New intelligent algorithms and techniques need to be developed to address the challenges of task scheduling in the dynamic and heterogeneous fog environment.

VII. CONCLUSION

The state of the art task scheduling mechanisms in fog-cloud are investigated in this paper. According to the reviewed articles from 2018 to 2023, the number of articles published in the field of cloud-fog task scheduling is very high in 2021. A taxonomy is also introduced for fog task scheduling techniques including machine learning, heuristic-based, metaheuristic-based, and deterministic mechanisms. The selected 47 articles are investigated in these four categories. The benefits and weaknesses of each of these techniques are reviewed according to the quality criteria defined in the article. The investigation of fog task scheduling mechanisms showed that the energy consumption criterion has the most importance and priority among other major factors for implementing scheduling in the fog environment. In general, task scheduling mechanisms in the fog environment still need improvements in terms of reducing energy consumption, enhancing security, improving performance, reducing latency, managing fault tolerance, etc.

The main purpose of this survey is to help researchers understand task scheduling algorithms and their challenges in fog environments. This survey tries to perform a comprehensive and systematic study in cloud-fog, but it also has some limitations. This paper fails to study various fog-based task scheduling methods available in various sources. Nevertheless, the presented results help researchers to develop more effective task scheduling techniques in fog-cloud environments.

ACKNOWLEDGMENT

The result was created through solving the student project "Security analysis and developing lightweight ciphers and protocols" using objective-oriented support for specific university research from the University of Finance and Administration, Prague, Czech Republic. The authors would like to thank Michal Merta and Zdeněk Truhlář for their help with the research connected with the topic of the article.

REFERENCES

[1] S. Klein, "The world of big data and IoT," in *IoT Solutions in Microsoft's Azure IoT Suite*. Cham, Switzerland: Springer, 2017, pp. 3–13.

[2] M. Laroui, B. Nour, H. Moungla, M. A. Cherif, H. Afifi, and M. Guizani, "Edge and fog computing for IoT: A survey on current research activities & future directions," *Comput. Commun.*, vol. 180, pp. 210–231, Dec. 2021.

[3] 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.

[4] S. Bansal, H. Aggarwal, and M. Aggarwal, "A systematic review of task scheduling approaches in fog computing," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 9, p. e4523, Sep. 2022.

[5] M. R. Alizadeh, V. Khajehvand, A. M. Rahmani, and E. Akbari, "Task scheduling approaches in fog computing: A systematic review," *Int. J. Commun. Syst.*, vol. 33, no. 16, p. e4583, Aug. 2020.

[6] X. Yang and N. Rahmani, "Task scheduling mechanisms in fog computing: Review, trends, and perspectives," *Kybernetes*, vol. 50, no. 1, pp. 22–38, Mar. 2021.

[7] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[8] P. Hosseinioun, M. Kheirabadi, S. R. K. Tabbakh, and R. Ghaemi, "ATask scheduling approaches in fog computing: A survey," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, Mar. 2022.

[9] N. Kaur, A. Kumar, and R. Kumar, "A systematic review on task scheduling in fog computing: Taxonomy, tools, challenges, and future directions," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 21, p. e6432, Nov. 2021.

[10] K. Matrouk and K. Alatoun, "Scheduling algorithms in fog computing: A survey," *Int. J. Netw. Distrib. Comput.*, vol. 9, no. 1, pp. 59–74, 2021.

[11] P. Hosseinioun, M. Kheirabadi, S. R. Kamel Tabbakh, and R. Ghaemi, "A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm," *J. Parallel Distrib. Comput.*, vol. 143, pp. 88–96, Sep. 2020.

[12] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3246–3257, Aug. 2018.

[13] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*. Cham, Switzerland: Springer, 2018, pp. 103–130.

[14] Z. Hao, E. Novak, S. Yi, and Q. Li, "Challenges and software architecture for fog computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 44–53, Mar. 2017.

[15] 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, Nov. 2017.

[16] Y. Liu, A. Yang, Q. Zeng, Y. Sun, J. Gao, and Z. Lv, "Task scheduling of real-time traffic information processing based on digital twins," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 18, 2022, doi: 10.1109/TITS.2022.3196166.

[17] Q. Qi, L. Zhang, J. Wang, H. Sun, Z. Zhuang, J. Liao, and F. R. Yu, "Scalable parallel task scheduling for autonomous driving using multi-task deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13861–13874, Nov. 2020.

[18] 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.

[19] J. Wang and D. Li, "Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing," *Sensors*, vol. 19, no. 5, p. 1023, Feb. 2019.

[20] A. Asensio, X. Masip-Bruin, R. J. Durán, I. de Miguel, G. Ren, S. Daijavad, and A. Jukan, "Designing an efficient clustering strategy for combined fog-to-cloud scenarios," *Future Gener. Comput. Syst.*, vol. 109, pp. 392–406, Aug. 2020.

[21] A. Shahidinejad, M. Ghobaei-Arani, and M. Masdari, "Resource provisioning using workload clustering in cloud computing environment: A hybrid approach," *Cluster Comput.*, vol. 24, no. 1, pp. 319–342, Mar. 2021.

[22] I. Ullah and H. Y. Youn, "Task classification and scheduling based on K-means clustering for edge computing," *Wireless Pers. Commun.*, vol. 113, no. 4, pp. 2611–2624, Aug. 2020.

[23] P. Vinothiyalakshmi and R. Anitha, "Workload mining in cloud computing using extended cloud Dempster–Shafer theory (ECDST)," *Wireless Pers. Commun.*, vol. 114, no. 1, pp. 389–407, Sep. 2020.

[24] Y. Charband and N. J. Navimipour, "Online knowledge sharing mechanisms: A systematic review of the state of the art literature and recommendations for future research," *Inf. Syst. Frontiers*, vol. 18, no. 6, pp. 1131–1151, Dec. 2016.

[25] B. Kitchenham, "Procedures for performing systematic reviews," Dept. Comput. Sci., Keele Univ., Keele, U.K., Tech. Rep. 0400011T.1, vol. 33, 2004, pp. 1–26.

[26] H. S. Arri, R. Singh, S. Jha, D. Prashar, G. P. Joshi, and I. C. Doo, "Optimized task group aggregation-based overflow handling on fog computing environment using neural computing," *Mathematics*, vol. 9, no. 19, p. 2522, Oct. 2021.

[27] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–18, May 2019.

[28] P. Gazori, D. Rahbari, and M. Nickray, "Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach," *Future Gener. Comput. Syst.*, vol. 110, pp. 1098–1115, Sep. 2020.

[29] S. Shadroo, A. M. Rahmani, and A. Rezaee, "The two-phase scheduling based on deep learning in the Internet of Things," *Comput. Netw.*, vol. 185, Feb. 2021, Art. no. 107684.

[30] S. Swarup, E. M. Shakshuki, and A. Yasar, "Energy efficient task scheduling in fog environment using deep reinforcement learning approach," *Procedia Comput. Sci.*, vol. 191, pp. 65–75, 2021.

[31] S. Ghanavati, J. Abawajy, and D. Izadi, "Automata-based dynamic fault tolerant task scheduling approach in fog computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 488–499, Jan. 2022.

[32] B. Nair and S. Bhanu, "A reinforcement learning algorithm for rescheduling preempted tasks in fog nodes," *J. Scheduling*, vol. 25, pp. 1–19, Apr. 2022.

[33] B. Jamil, H. Ijaz, M. Shojafar, and K. Munir, "IRATS: A DRL-based intelligent priority and deadline-aware online resource allocation and task scheduling algorithm in a vehicular fog network," *Ad Hoc Netw.*, vol. 141, Mar. 2023, Art. no. 103090.

[34] K. Li, W. Ni, X. Yuan, A. Noor, and A. Jamalipour, "Deep-graph-based reinforcement learning for joint cruise control and task offloading for aerial edge Internet of Things (EdgeIoT)," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21676–21686, Nov. 2022.

[35] H. Liu, H. Zhou, H. Chen, Y. Yan, J. Huang, A. Xiong, S. Yang, J. Chen, and S. Guo, "A federated learning multi-task scheduling mechanism based on trusted computing sandbox," *Sensors*, vol. 23, no. 4, p. 2093, Feb. 2023.

[36] J. Shi, J. Du, J. Wang, and J. Yuan, "Federated deep reinforcement learning-based task allocation in vehicular fog computing," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2022, pp. 1–6.

[37] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and N. Kara, "FoG-Match: An intelligent multi-criteria IoT-fog scheduling approach using game theory," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1779–1789, Aug. 2020.

[38] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," *J. Netw. Comput. Appl.*, vol. 201, May 2022, Art. no. 103333.

[39] E. Hosseini, M. Nickray, and S. Ghanbari, "Optimized task scheduling for cost-latency trade-off in mobile fog computing using fuzzy analytical hierarchy process," *Comput. Netw.*, vol. 206, Apr. 2022, Art. no. 108752.

[40] R. Madhura, B. L. Elizabeth, and V. R. Uthariaraj, "An improved list-based task scheduling algorithm for fog computing environment," *Computing*, vol. 103, no. 7, pp. 1353–1389, Jul. 2021.

[41] T. Choudhari, M. Moh, and T.-S. Moh, "Prioritized task scheduling in fog computing," in *Proc. ACMSE Conf.*, Mar. 2018, pp. 1–8.

[42] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Privacy-preserving for the Internet of Things in multi-objective task scheduling in cloud-fog computing using goal programming approach," *Peer Peer Netw. Appl.*, vol. 14, no. 6, pp. 3865–3890, Nov. 2021.

[43] D. Charântola, A. C. Mestre, R. Zane, and L. F. Bittencourt, "Component-based scheduling for fog computing," in *Proc. 12th IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, Dec. 2019, pp. 3–8.

[44] K. Verma, A. Kumar, M. S. Ul Islam, T. Kanwar, and M. Bhushan, "Rank based mobility-aware scheduling in fog computing," *Informat. Med. Unlocked*, vol. 24, 2021, Art. no. 100619.

[45] Z. Liu, X. Yang, Y. Yang, K. Wang, and G. Mao, "DATS: Dispersive stable task scheduling in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3423–3436, Apr. 2019.

[46] M. Guo, L. Li, and Q. Guan, "Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems," *IEEE Access*, vol. 7, pp. 78685–78697, 2019.

[47] F. Xu, Z. Yin, A. Gu, Y. Li, H. Yu, and F. Zhang, "Adaptive scheduling strategy of fog computing tasks with different priority for intelligent production lines," *Proc. Comput. Sci.*, vol. 183, pp. 311–317, 2021.

[48] L. Li, M. Guo, L. Ma, H. Mao, and Q. Guan, "Online workload allocation via fog-fog-cloud cooperation to reduce IoT task service delay," *Sensors*, vol. 19, no. 18, p. 3830, Sep. 2019.

[49] L. Li, Q. Guan, L. Jin, and M. Guo, "Resource allocation and task offloading for heterogeneous real-time tasks with uncertain duration time in a fog queueing system," *IEEE Access*, vol. 7, pp. 9912–9925, 2019.

[50] R. M. Abdelmoneem, A. Benslimane, and E. Shaaban, "Mobility-aware task scheduling in cloud-fog IoT-based healthcare architectures," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107348.

[51] T. Aladwani, "Scheduling IoT healthcare tasks in fog computing based on their importance," *Proc. Comput. Sci.*, vol. 163, pp. 560–569, Jan. 2019.

[52] M. Ghobaei-Arani, A. Souri, F. Safara, and M. Norouzi, "An efficient task scheduling approach using moth-flame optimization algorithm for cyber–physical system applications in fog computing," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, Feb. 2020.

[53] H. T. T. Binh, T. T. Anh, D. B. Son, P. A. Duc, and B. M. Nguyen, "An evolutionary algorithm for solving task scheduling problem in cloud-fog computing environment," in *Proc. 9th Int. Symp. Inf. Commun. Technol.*, 2018, pp. 397–404.

[54] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, and N. Kumar, "Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5068–5076, Jul. 2021.

[55] R. O. Aburukba, T. Landolsi, and D. Omer, "A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices," *J. Netw. Comput. Appl.*, vol. 180, Apr. 2021, Art. no. 102994.

[56] M. Abdel-Basset, D. El-Shahat, M. Elhoseny, and H. Song, "Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12638–12649, Aug. 2021.

[57] M. Tanha, M. Hosseini Shirvani, and A. M. Rahmani, "A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments," *Neural Comput. Appl.*, vol. 33, no. 24, pp. 16951–16984, Dec. 2021.

[58] H. R. Boveiri, R. Khayami, M. Elhoseny, and M. Gunasekaran, "An efficient swarm-intelligence approach for task scheduling in cloud-based Internet of Things applications," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 9, pp. 3469–3479, Sep. 2019.

[59] T. Wang, X. Wei, C. Tang, and J. Fan, "Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints," *Peer Peer Netw. Appl.*, vol. 11, no. 4, pp. 793–807, Jul. 2018.

[60] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multi-objective task scheduling in cloud-fog computing using goal programming approach," *Cluster Comput.*, vol. 25, no. 1, pp. 141–165, Feb. 2022.

[61] Z. Movahedi, B. Defude, and A. M. Hosseininia, "An efficient population-based multi-objective task scheduling approach in fog computing systems," *J. Cloud Comput.*, vol. 10, no. 1, pp. 1–31, Dec. 2021.

[62] A. M. Yadav, K. N. Tripathi, and S. Sharma, "A bi-objective task scheduling approach in fog computing using hybrid fireworks algorithm," *J. Supercomput.*, vol. 78, pp. 4236–4260, Aug. 2021.

[63] M. Abdel-Basset, R. Mohamed, R. K. Chakrabortty, and M. J. Ryan, "IEGA: An improved elitism-based genetic algorithm for task scheduling problem in fog computing," *Int. J. Intell. Syst.*, vol. 36, no. 9, pp. 4592–4631, Sep. 2021.

[64] M. Jia, J. Zhu, and H. Huang, "Energy and delay-ware massive task scheduling in fog-cloud computing system," *Peer Peer Netw. Appl.*, vol. 14, no. 4, pp. 2139–2155, Jul. 2021.

[65] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet of Things requests to minimize latency in hybrid fog-cloud computing," *Future Gener. Comput. Syst.*, vol. 111, pp. 539–551, Oct. 2020.

[66] Y. Kyung, "Prioritized task distribution considering opportunistic fog computing nodes," *Sensors*, vol. 21, no. 8, p. 2635, Apr. 2021.

[67] J.-F. Tsai, C.-H. Huang, and M.-H. Lin, "An optimal task assignment strategy in cloud-fog computing environment," *Appl. Sci.*, vol. 11, no. 4, p. 1909, Feb. 2021.

[68] A. C. Caminero and R. Muñoz-Mansilla, "Quality of service provision in fog computing: Network-aware scheduling of containers," *Sensors*, vol. 21, no. 12, p. 3978, Jun. 2021.

[69] A. Razaque, Y. Jararweh, B. Alotaibi, M. Alotaibi, S. Hariri, and M. Almiani, "Energy-efficient and secure mobile fog-based cloud for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 127, pp. 1–13, Feb. 2022.

[70] A. M. Rahmani, E. Azhir, S. Ali, M. Mohammadi, O. H. Ahmed, M. Y. Ghafour, S. H. Ahmed, and M. Hosseinzadeh, "Artificial intelligence approaches and mechanisms for big data analytics: A systematic study," *PeerJ Comput. Sci.*, vol. 7, p. e488, Apr. 2021.

[71] A. M. Rahmani, "Automatic COVID-19 detection mechanisms and approaches from medical images: A systematic review," *Multimedia Tools Appl.*, vol. 81, pp. 28779–28798, Mar. 2022.

[72] J.-F. Cordeau, M. Gendreau, and G. Laporte, "A Tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, no. 2, pp. 105–119, Sep. 1997.

[73] E. Azhir, N. J. Navimipour, M. Hosseinzadeh, A. Sharifi, M. Unal, and A. Darwesh, "Join queries optimization in the distributed databases using a hybrid multi-objective algorithm," *Cluster Comput.*, vol. 25, pp. 1–16, Nov. 2021.

[74] F. Xhafa and A. Abraham, "Computational models and heuristic methods for grid scheduling problems," *Future Gener. Comput. Syst.*, vol. 26, no. 4, pp. 608–621, Apr. 2010.

[75] J. Xu, Z. Hao, R. Zhang, and X. Sun, "A method based on the combination of laxity and ant colony system for cloud-fog task scheduling," *IEEE Access*, vol. 7, pp. 116218–116226, 2019.

[76] K.-C. Lin, Y.-H. Huang, J. C. Hung, and Y.-T. Lin, "Modified cat swarm optimization algorithm for feature selection of support vector machines," in *Frontier and Innovation in Future Computing and Communications*. Cham, Switzerland: Springer, 2014, pp. 329–336.

[77] U. Schwiegelshohn and R. Yahyapour, "Analysis of first-come-first-serve parallel job scheduling," in *Proc. SODA*, vol. 98, 1998, pp. 629–638.

[78] L. Yin, J. Luo, and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4712–4721, Oct. 2018.

[79] S. Sharma and H. Saini, "A novel four-tier architecture for delay aware scheduling and load balancing in fog environment," *Sustain. Computing, Informat. Syst.*, vol. 24, Dec. 2019, Art. no. 100355.

MEHDI HOSSEINZADEH received the B.S. degree in computer hardware engineering from Islamic Azad University, Dezfol Branch, Iran, in 2003, and the M.Sc. and Ph.D. degrees in computer system architecture from the Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2005 and 2008, respectively. He is the author/coauthor of more than 150 publications in technical journals and conferences. His research interests include SDN, information technology, data mining, big data analytics, e-commerce, e-marketing, and social networks.

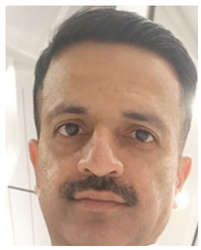
ELHAM AZHIR received the B.S. and M.Sc. degrees in software engineering from Qazvin Islamic Azad University, Qazvin, Iran, in 2010 and 2014, respectively. She is currently pursuing the Ph.D. degree in computer engineering with the Science and Research Branch, Islamic Azad University, Tehran, Iran. Her main research interests include query optimization, cloud computing, big data analytics, distributed systems, and programming.

JAN LANSKY received the M.S. and Ph.D. degrees in computer science (software systems) from Charles University, Prague, Czech Republic, in 2005 and 2009, respectively. Since March 2009, he has been a Professor with the Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, Prague, where he has also been the Head of the Department, since September 2014. His research interests include cryptocurrencies, text compression, and databases.

STANISLAVA MILDEOVA received the degree from the University of Economics, Prague, Czech Republic. Since 2017, she has been an Associate Professor and the Deputy Head of the Department of Informatics and Mathematics, Faculty of Economic Studies, University of Finance and Administration, Prague. In her work, she focuses on applied informatics and systems science with a focus on systems dynamics. She has been the Editor-in-Chief of the *Acta Informatica Pragensia* journal.

OMED HASSAN AHMED received the B.Sc. and Higher National Diploma (HND) degrees in information technology from Teesside University, U.K., and the M.Sc. degree in computer science from Newcastle University, U.K. He is currently a Ph.D. Researcher with the University of Huddersfield, U.K. He is also the Head of the Department of Information Technology, College of Science and Technology, University of Human Development, Iraq, where he has been a Faculty Member, since 2013.

MAZHAR HUSSAIN MALIK received the Ph.D. degree in computer science with specialization in computer networks. He was involved in academic and different industrial positions for various universities and companies, including the Institute of Southern Punjab, Multan Pakistan, SBE Electronics, U.K., NCR Corporation, USA, and The University of Lahore, Islamabad. Since September 2017, he has been a Senior Lecturer with the Global College of Engineering and Technology, Muscat, Oman. His research interests include wireless communications, cloud computing, sensors networks, network security and forensics, and artificial intelligence. He is an editorial board member of peer-reviewed journals and a Reviewer of IEEE Access, the *International Journal of Advanced Computer Science and Applications* (IJACSA), and the *International Journal of Computer Science and Information Security* (IJCSIS).



FAHEEM KHAN received the bachelor's degree from the University of Peshawar, Pakistan, in 2009, and the Ph.D. degree in computer science from the University of Malakand, Pakistan, in 2017. He is currently an Assistant Professor with Gachon University, South Korea. He is the author/coauthor of more than 30 publications in technical journals and conferences. His research interests include computer networking, information technology, sensor networks, wireless networks, mobile ad-hoc networks, the IoT, AI, and social networks.