Bibliometric Analysis of Computation Offloading in Fog Computing

Jyoti Yadav[1] and Suman[2]

Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonipat, Haryana, India

Abstract
After accepting IoT(Internet of Things) as a research trend, computation offloading is one of the promising solutions for resource-restricted and non-scalable mobile devices. Video processing, augmented reality, object tracking, and natural language processing are resource-intensive applications getting popular day by day. These applications cannot run properly on low-resource end devices. Computation offloading in fog Computing is a key technology that allows the execution of such applications with limited resources. On the other side, computation offloading task requests lowers the core networks' communication pressure and minimizes long-distance data transmission latency. Businesses and academics are currently doing lots of research on computation offloading strategies and methodologies in the fog computing environment. Researchers have published many review papers. However, no bibliometric analysis has been done to assess the popularity of computation offloading in the fog computing environment. This paper provides a bibliometric analysis of computation offloading in fog computing, emphasizing how optimization techniques are used to solve offloading problems. The review results also presented the most popular optimization techniques for solving offloading problems in fog computing. The paper concludes with the interpretation of metaheuristic techniques and open issues of offloading problems.
.***Keywords:***
Metaheuristic, Swarm Intelligence, Computation Offloading, Fog Computing, Bibliometric Analysis

1. Introduction

IoT technology allows the augmented use of IoT devices in diverse areas of public, private, and industrial spaces like manufacturing, energy, mining, transportation, aerospace, process control, and many more[1]. Such an extension of the IoT demands a change in the traditional way of executing services over a centralized cloud. The IoT environment's essential characteristics are heterogeneity, highly geographically distributed devices, scalability, low latency, and minimum delay[2]. Most edge layer devices in IoT are equipped with sensors, actuators, or data collectors, which have limited computation capacity and energy that requires data processing locally or offloading to a remote computing device, i.e., cloud data center or distributed fog devices[3].

Fog computing is developing technology for providing processing, memory, control, and networking capabilities to IoT devices. End devices can offload their data or resource-intensive application to nearby fog nodes available in the fog layer as an alternative to the distant cloud[4]. In three-layer architecture, the fog layer is positioned between the resource-rich cloud data centers and the sensors, allowing the collection and loading of data to occur near the end devices[5]. The most accepted and practical fog devices are routers, gateways, switches, micro data centers, etc. The fog nodes execute processing tasks, reduce network traffic, and increase speed. But fog nodes have limited computation power and storage capacity compared to cloud data centers. One of the primary aims of fog computing is computation offloading, as this helps mobile devices circumvent limitations such as processing power, battery life, and storage space. The numerous factors that must be considered, such as what to offload, where to offload, how to offload, and why to offload, make this an np-hard optimization problem[6], [7].
What to offload: The decision engine should decide which tasks are offloaded to the top levels. Offloading solutions may be divided into two categories based on this fact: (I) Full offloading, in which the whole task is offloaded to a remote server such as fog nodes or cloud VM layers; (II) partial offloading, in which the task is partitioned into subtask; the only resource-intensive task is offloaded to other layers[8], [9].
When to offload: Due to uncertainties of the environment on both sides, i.e., the user side or network side, the decision engine should decide when the computation is offloaded to higher layers to meet the desired QoS. Optimal time scheduling of offloading calculations has been used to overcome this issue[10].

Manuscript received October 5, 2025
Manuscript revised October 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.10.9

Where to offload: The decision engine should decide where to offload the task, whether at the local fog server, remote fog server, or cloud server on the basis of various parameters, such as available bandwidth, latency, VM utilization, etc. Some researchers have attempted to overcome this issue by using metaheuristic techniques[11], [12].

Because of the multidimensionality of the problem, there are no best algorithms that may generate optimal solutions within the polynomial time for such problems. Metaheuristic optimization techniques handle these problems by giving near-optimal solutions within rational time. Swarm intelligence has obtained enormous recognition in the past few years due to its efficiency and efficacy in solving big and intricate problems[13].

The main objective of the paper is to present a bibliometric analysis of the existing literature on computation offloading in fog computing, especially emphasizing optimization techniques contextual to offloading in fog computing. To be precise, the subsequent points emphasize the major contributions of this review paper.

1. Upon assessing state of the art on computation offloading in fog computing, we realized the lack of a bibliometric analysis in computation offloading in fog computing. Thus we summarize the existing literature on the a) most popular journals, b)the most productive authors, c) most published countries, and d) the hotspots in fog computing.
2. Provides an overview of the metaheuristic techniques used for computation offloading so far.
3. Evaluation of the proposed algorithms on the basis of different performance metrics.
4. We interpreted the research findings and discussed some open issues in the computation offloading scope.

The remaining sections of the paper are organized as follows: In Section 2, we present a bibliometric study of literature relating to computation offloading in fog computing. In Section 3, metaheuristic techniques for computation offloading problem solutions in fog computing are presented along with performance metrics. Section 4 represents the interpretation, section 5 presents open issues of offloading, and Section 6 presents the summary and conclusions of this work.

2. Bibliometric Analysis of Computation Offloading in Fog Computing

To demonstrate the widespread adoption of computation offloading in fog computing, this section provides a bibliometric analysis of related literature. The rise in popularity of bibliometric analysis in business research over the past few years [14] can be attributed to (1) the advancement, accessibility, and approachability of bibliometric software such as Gephi, Leximancer, and VOSviewer, as well as scientific databases such as science direct, scopus, ACM digital library and web of science, etc. Bibliometric analysis helps in managing a large amount of scholarly literature. The research questions addressed by the study are as follows:

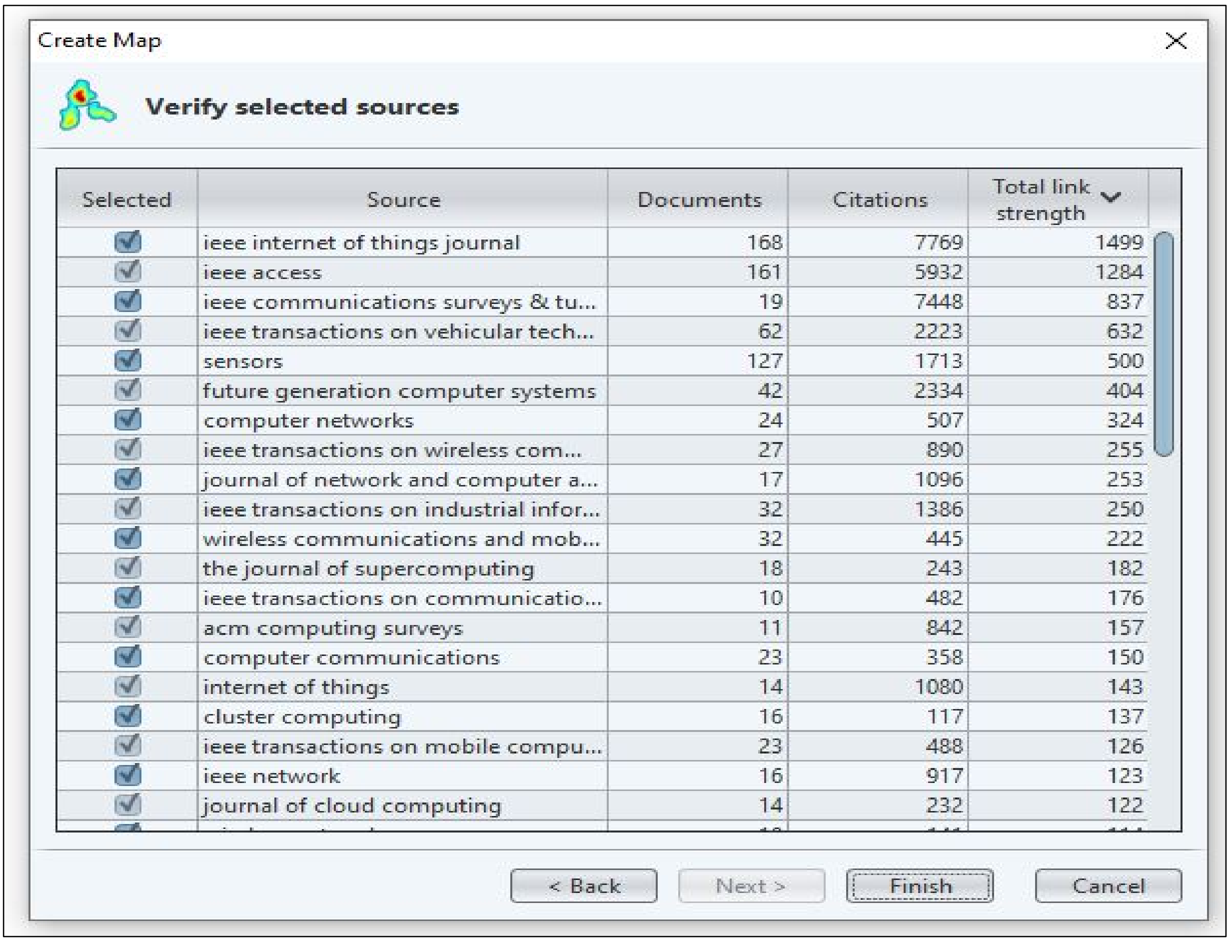
Table 1 Research Questions

Research Questions (RQ)	
RQ1	Which journals are the most popular in computation offloading in the fog computing environment?
RQ2	Which authors are the leading solver for the computation offloading problem in fog computing?
RQ3	What is the research status of computation offloading in fog computing in countries/ regions around the world?
RQ4	What are the fog computing hot spots?
RQ5	Which optimization techniques have been explored for computation offloading in fog computing?

The first four research questions are addressed by using VOS viewer bibliometric analysis tool.

RQ1: Analysis of core journals in the field of computation offloading in fog computing

As the primary disseminators of the research process, journals play a crucial role in the progress of the research topic. Figure 1 lists the journals with the highest number of publications in the field of task offloading in fog computing. Most articles about computation offloading in fog computing are published in the two most prominent journals in this area, IEEE Access and IEEE Internet of Things Journal, as shown in Figure 1. IEEE Internet of Things Journal has published 168 studies, placing it in the first position. However, IEEE Access has published 161 literature reviews in this field, placing it in second place.



Selected	Source	Documents	Citations	Total link strength
☑	ieee internet of things journal	168	7769	1499
☑	ieee access	161	5932	1284
☑	ieee communications surveys & tu...	19	7448	837
☑	ieee transactions on vehicular tech...	62	2223	632
☑	sensors	127	1713	500
☑	future generation computer systems	42	2334	404
☑	computer networks	24	507	324
☑	ieee transactions on wireless com...	27	890	255
☑	journal of network and computer a...	17	1096	253
☑	ieee transactions on industrial infor...	32	1386	250
☑	wireless communications and mob...	32	445	222
☑	the journal of supercomputing	18	243	182
☑	ieee transactions on communicatio...	10	482	176
☑	acm computing surveys	11	842	157
☑	computer communications	23	358	150
☑	internet of things	14	1080	143
☑	cluster computing	16	117	137
☑	ieee transactions on mobile compu...	23	488	126
☑	ieee network	16	917	123
☑	journal of cloud computing	14	232	122

Fig 1 Journals with the highest number of publications in computation offloading

Next, the journal citation, i.e., the journal with the most citations and the journal most frequently cited by the same source. The minimum number of journal citations was set to 50 in VOSviewer. Figure 2 is a picture of the network of journals that cite each other. In Figure 2, the number of times each item was cited is represented by its size relative to the other dots and words. The greater the size of the dots and words, the greater the number of times that they are cited. Figure 2 reveals that IEEE Access and IEEE Internet of Things is the most frequently cited journals, indicating that these two publications significantly impact the computation offloading field. In addition, the third-ranked sensor has been cited much less frequently.

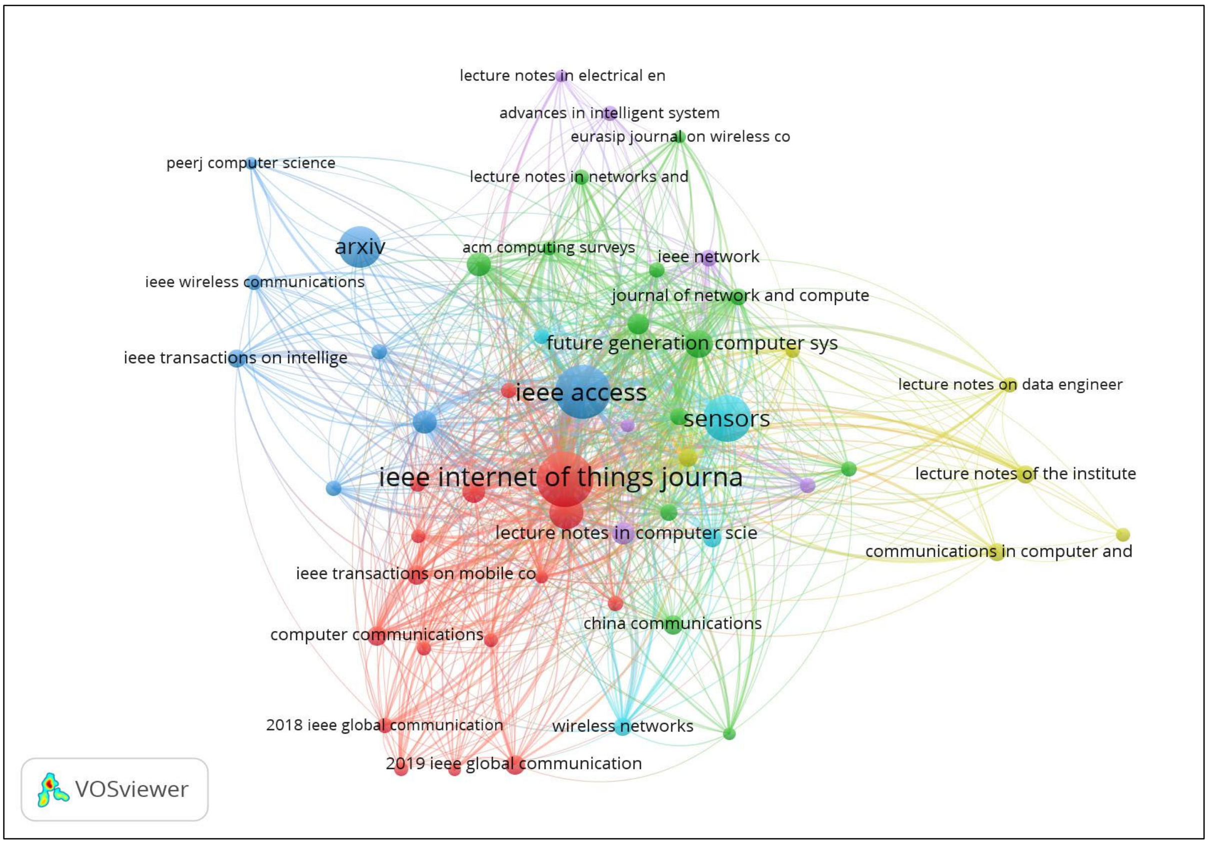


Fig 오류! 지정한 스타일은 사용되지 않습니다. Visualization of journal citation network

RQ2: Analysis of core authors in fog computing

The most productive authors are the core authors. They are the academics who have written a plethora of literature reviews for their respective fields. Analyzing and locating credible offloading experts is facilitated by examining the core authors.

Figure 3 indicates that Yang is the most prolific author. However, scientometrics has conducted extensive research on accurately quantifying academic findings' publication. Their view is that while counting works of literature is one approach, citing counts provide more insight.

The minimum number of author citations in VOSviewer was set to 14. Then, 20 of the 6198 authors meet this criterion.

Figure 4 depicts an overlay visualization of the author citation network. The sizes of the dots and words in Figure 4 represent the cited times. The authors with the most citations are Zhang and Yan in the first place, whereas Raj Kumar Buyya is second. This indicates that their work is widely recognized and significantly impacts the computation offloading in fog computing.

Create Map

Verify selected authors

Selected	Author	Documents	Citations	Total link strength
✓	yang, yang	36	651	236
✓	wang, kunlun	18	372	182
✓	luo, xiliang	18	351	143
✓	mukherjee, mithun	17	594	100
✓	chen, xu	15	924	82
✓	han, zhu	24	682	74
✓	zhang, qi	14	195	74
✓	yu, f. richard	23	1123	68
✓	chang, zheng	15	835	58
✓	zhang, yan	17	2634	58
✓	srirama, satish narayana	14	204	52
✓	chen, qianbin	14	644	48
✓	xu, xiaolong	25	922	48
✓	buyya, rajkumar	22	1308	46
✓	qi, lianyong	17	734	37
✓	zhong, zhangdui	16	159	31
✓	zhang, ning	16	416	30
✓	peng, mugen	14	592	26
✓	huh, eui-nam	14	168	15
✓	li, keqin	14	237	12

< Back | Next > | Finish | Cancel

Fig 2 Authors with the highest number of documents and citations

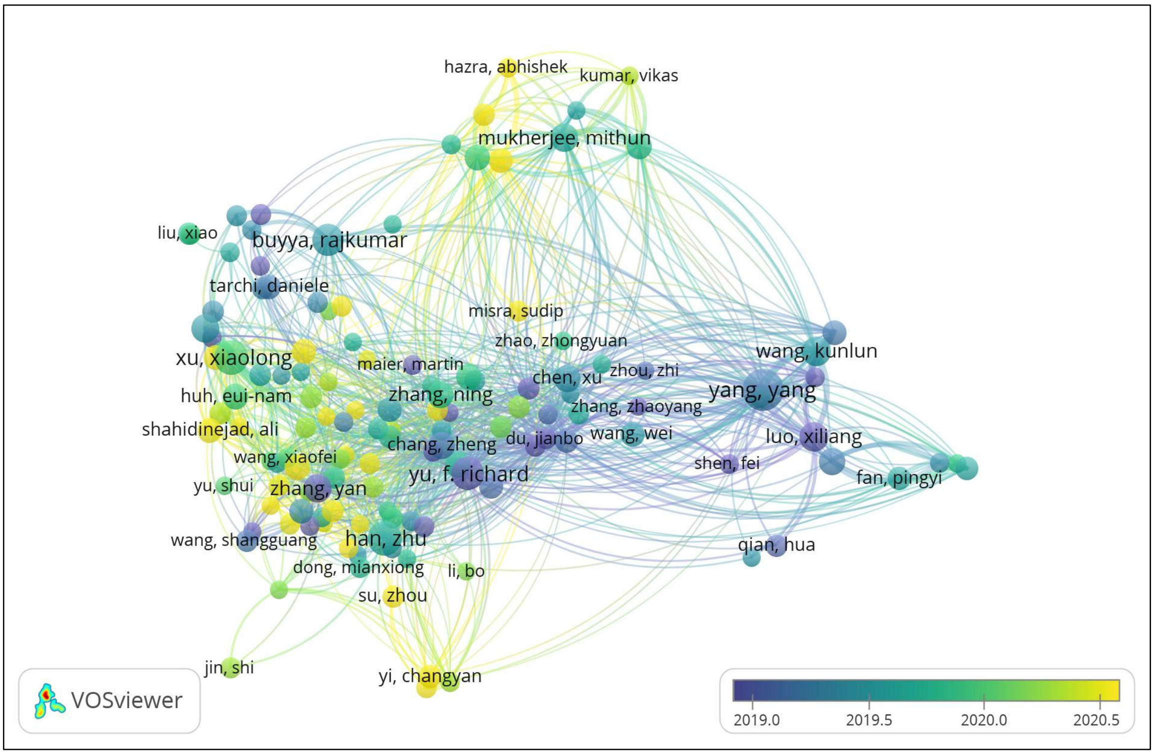
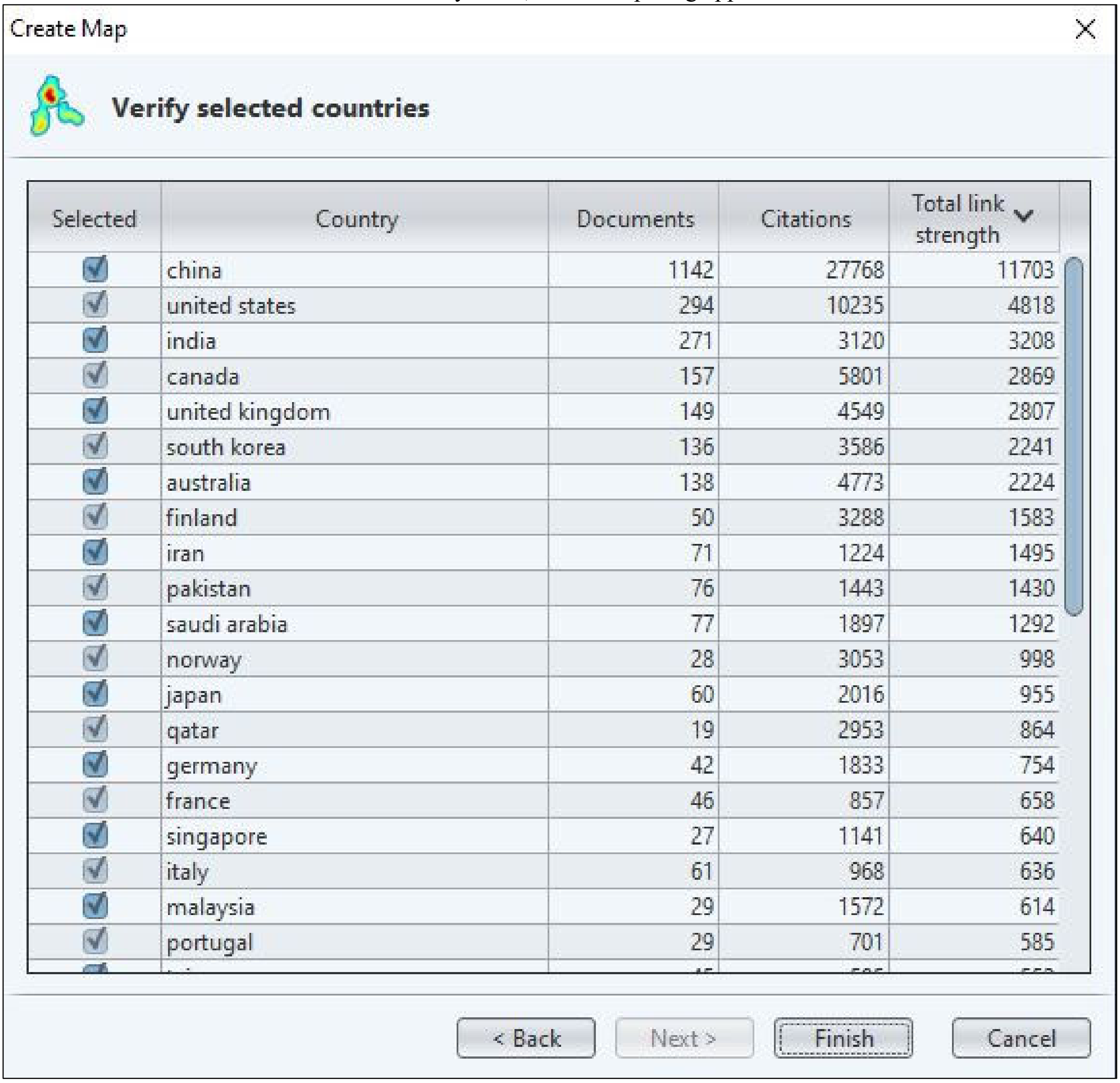


Fig 3 Visualization of author's citation network

RQ3: Analysis of countries and regions in fog computing

Significant efforts have been made to encourage the growth of computation offloading in fog computing. Figure 5 depicts the nations with the most publications. Figure 5 demonstrates that China has the highest productivity, having produced 1142 literature studies with 27768 citations. It is followed by the United States, which contributed 294 literature studies with 10235 citations. Then followed by India, which contributed 271 studies with 3120 citations. Figure 6 represents density visualization of the most popular countries that have published articles in computation offloading in literature studies, providing a solid foundation for this work. If the country's citation count is higher, the text will be lighter, and the words will be larger. This puts these countries at the forefront of fog computing research and provides a brighter future for the growth of fog computing applications.



Selected	Country	Documents	Citations	Total link strength
☑	china	1142	27768	11703
☑	united states	294	10235	4818
☑	india	271	3120	3208
☑	canada	157	5801	2869
☑	united kingdom	149	4549	2807
☑	south korea	136	3586	2241
☑	australia	138	4773	2224
☑	finland	50	3288	1583
☑	iran	71	1224	1495
☑	pakistan	76	1443	1430
☑	saudi arabia	77	1897	1292
☑	norway	28	3053	998
☑	japan	60	2016	955
☑	qatar	19	2953	864
☑	germany	42	1833	754
☑	france	46	857	658
☑	singapore	27	1141	640
☑	italy	61	968	636
☑	malaysia	29	1572	614
☑	portugal	29	701	585

Fig 4 Countries with the highest number of documents and citations.

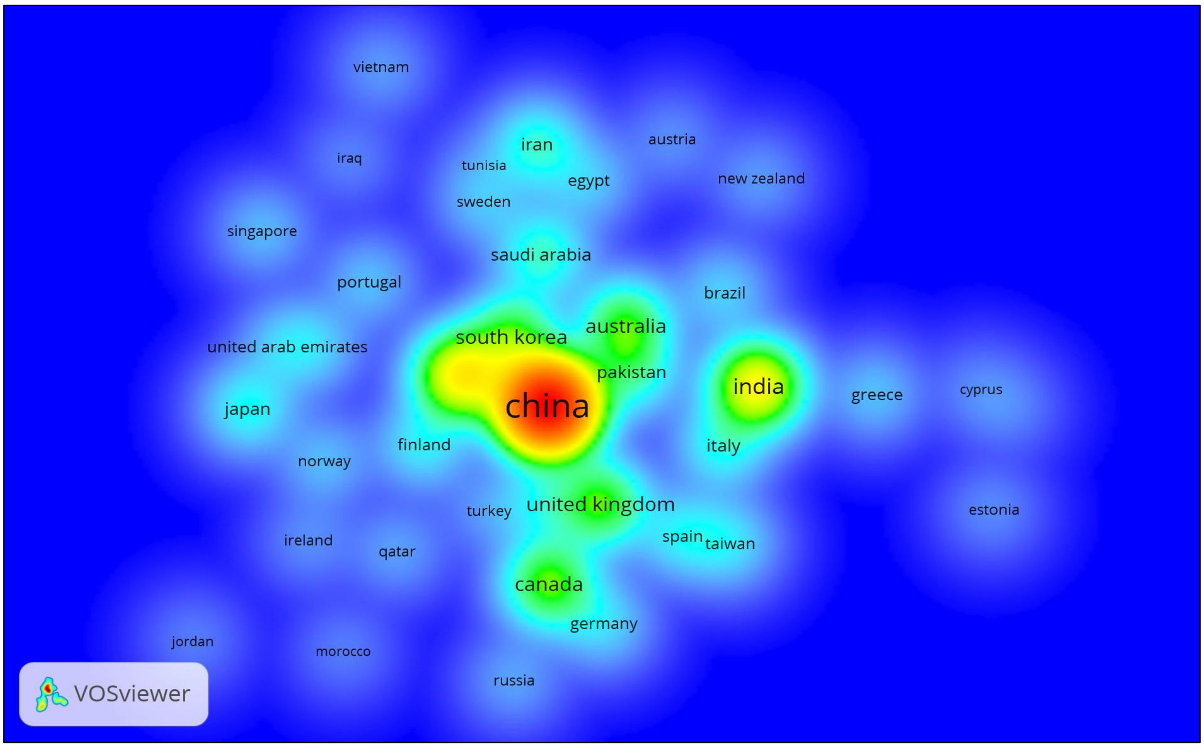


Fig 5 Density visualization of contributions of different countries for computation offloading

RQ4: Hotspot analysis in fog computing

Keywords are essential to the literature because they greatly condense its content. Research hotspots can be effectively revealed through the co-occurrence network analysis of keywords. There were four groups of related high-frequency keywords generated. A representation of the co-occurrence network of keywords is shown in Figure 7.

Cluster 1: It is the study of definite challenging fog computing problems, as indicated by the red node region. Some of the problems with fog computing are managing resources, allocating resources, offloading computations, using too much energy, and latency.

Cluster 2: It is the study of the overall fog computing architecture, as depicted by the green node region. Privacy and safety will be improved even further as a direct result of the overall architecture being subjected to optimization and innovation.

Cluster 3: Blue node region represents the study of fog computing applications. Combining mobile computing with big data technology and artificial intelligence algorithms makes fog computing more applicable to smart grids, smart homes, modern health care systems, smart cities, and other applications.

Cluster 4 is the investigation of edge computing with 5G, as indicated by the yellow node area. The development of 5G promises a higher availability of bandwidth.

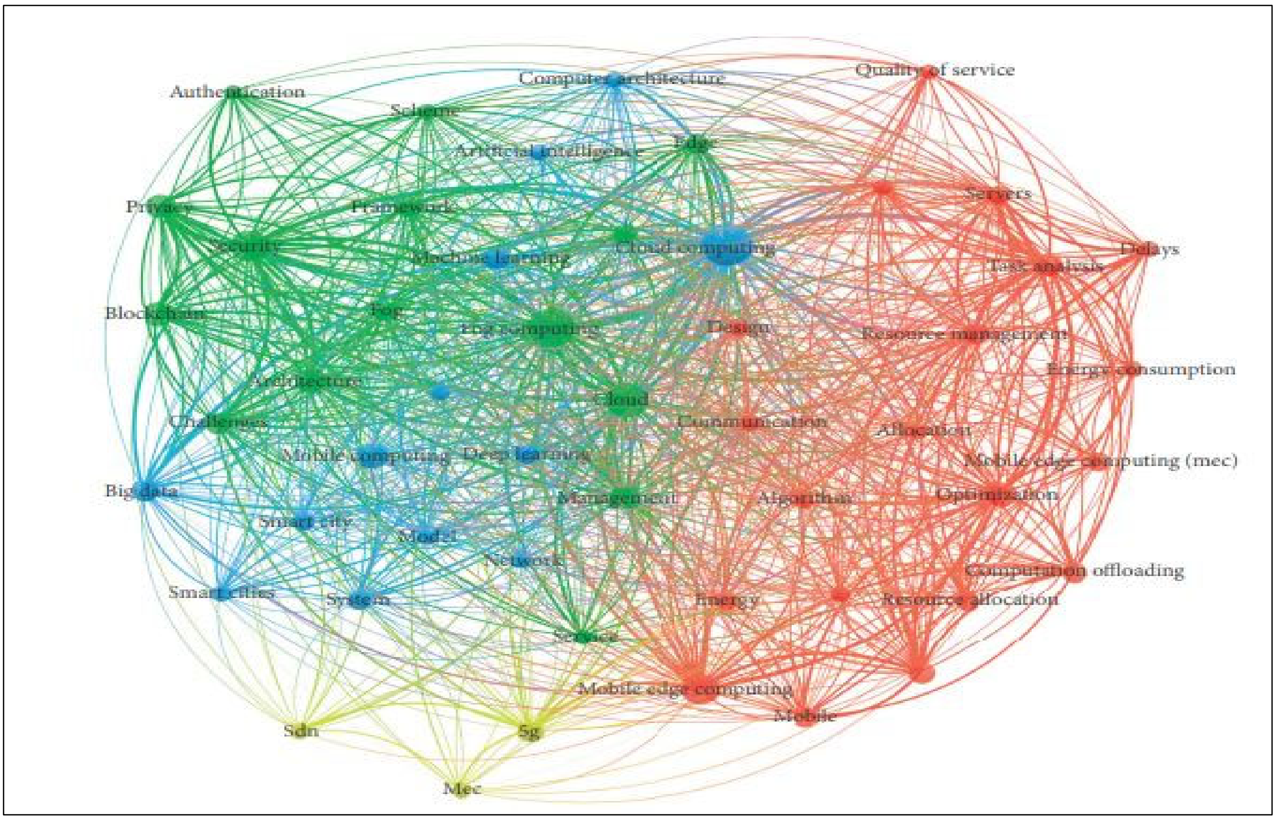


Fig 7 Hotspot analysis of fog computing

The fog computing research hotspots are understood through the statistics of frequently co-occurring keywords. In summary, it consists of four components: specific challenging problems, overall architecture, applied research, and collaborative research with 5G. Consequently, it is plausible that future research will focus on fog computing issues pertaining to resource management, resource allocation, computation offloading, load balancing, energy consumption, and delay. With the gradual maturation of fog computing technology and 5G, fog computing will be promoted and popularized further in smart grids, smart healthcare smart cities, smart transportation, and other areas.

RQ5 is about optimization techniques used for finding solutions to computation offloading problems in fog computing.

3. Optimization Techniques

To answer RQ5, we will discuss the existing research work on computation offloading in fog computing using optimization techniques. Optimization techniques find the best candidate solution from all possible sets of solutions. There are various types of optimization techniques depending upon the problems. These can be categorized as linear and nonlinear optimization techniques. Linear optimization techniques maximize or minimize the objective function subject to linear constraints, while nonlinear programming minimizes or maximizes nonlinear objective functions[16]. A nonlinear optimization problem can be further divided into deterministic and stochastic. Deterministic optimization techniques always follow specific rules to reach the destination without considering randomness. This technique accomplishes the identical solution if the originating point is alike. Moreover, an optimal solution is local instead of global.

Table 1: Optimization Techniques

Optimization Method	Pros	Cons
Linear Programming	✓ Adaptive ✓ Suitable for combinatorial problems ✓ Simple ✓ Easy to understood	✓ Only solve single optimization problems ✓ Work only on a linear variable ✓ Static in nature
Nonlinear Programming	✓ Can solve the nonlinear problem ✓ Robust ✓ Faster convergence	✓ Require high resources ✓ Can work only with nonlinear variables ✓ Complex as compared to linear programming
Deterministic	✓ Guarantee Global optimal solution for some problems	✓ Combinatorial Explosion issue
Stochastic	✓ Could handle large-scale problems	✓ It does not guarantee the optimal solution
Heuristic	✓ Flexible ✓ Easy to understand	✓ No guarantee for optimality of result obtained
Metaheuristic	✓ locate a good candidate solution in a rational time ✓ It involves some form of randomness to avoid local optima	✓ It does not guarantee the best solution
Evolutionary	✓ Solve Global Optimization Problem ✓ Scalable ✓ Robust	✓ Premature convergence ✓ Local optima issue
Swarm Intelligence	✓ Population-based ✓ Flexible ✓ Adaptive ✓ Robust ✓ Faster convergence	✓ Complex to understand ✓ Very sensitive ✓ Difficult to forecast the behavior of individual

On the other hand, stochastic optimization algorithms consider randomness and different translation rules and produce different solutions if the originating point is the same[17]. Stochastic algorithms are adaptive and produce both local and global solutions.

Stochastic algorithms can be further subdivided into two categories heuristic and metaheuristic techniques. The metaheuristic techniques are further divided into two parts evolutionary algorithm and swarm intelligence. Evolutionary algorithms are stimulated by biological phenomena to discover a near-optimal solution[18]. The most popular evolutionary algorithm is the Genetic Algorithm. On the other hand, swarm intelligence was stimulated by the activities of animals like birds, lions, wolves, ants, monkeys, insects, etc. [19]. The suitability of an algorithm for a problem depends upon the effectiveness, speed, and efficiency of the obtained solution. A suitable optimization technique is chosen for a particular problem based on the nature of the problem, accessibility of the computing resources, algorithm parameter tuning and setting, attributes of the algorithm, the expected solutions, and the ability of the decision-makers. As per the No Free Lunch Theorem, No algorithm suits all optimization problems[20]. Table no. 2 represents the pros and cons of the optimization technique

3.1 Offloading Optimization Problem

The issue of offloading in fog computing has become a striking topic for study and research in current years. The computation offloading optimization problem is to find the optimal offloading decision to offload the task and maximize the benefit. Optimization methods are widely applied in solving computation offloading problems. This issue has an optimization goal of whether the application can be offloaded from an end device to a cloud or to the fog node based on the demand of the end-users. In some research papers, the end device is responsible for deciding the offloading destination, while in other papers, gateways are responsible for offloading decisions [21]. Offloading is invariable in fog computing because end devices and fog devices are constrained in resources. For instance, the delay-sensitive task can be offloaded to fog nodes, or resource-intensive tasks can be offloaded to the cloud. For instance, author's in [22] designed a deep Q-learning-based autonomic framework that considerably improves end devices' performance by computation offloading, reducing service computing's latency. Author's in [4] designed a queuing model to find the solution to multi-objective computation offloading problems in fog computing. Authors in [23] designed a low-cost sub-optimal algorithm, where the decisions of offloading are taken by semidefinite and randomization. [24] designed a smart energy-based architecture that is applied to a cluster-based fog computing environment where offloading decisions take place.

Various researchers have used metaheuristic techniques to find the solution to the offloading problem. The metaheuristic-based algorithm shows near-optimal solutions within rational time. In general, these algorithms start with a set of random solutions then a predefined fitness function is evaluated for each solution to check its suitability of the solution[25]. An individual that meets the set threshold can participate in the next iteration. These steps are repeated until stopping criteria are met. To address offloading problem, many researchers have used meta-heuristic techniques such as[26], [27], [28], [29], [30].

3.2 Performance metrics for offloading problem

The goal of offloading systems is to provide a high quality of experience and high quality of service. To achieve this goal, offloading system's design must clearly understand how QoE and QoS are assessed. The Quality of Experience (QoE) for mobile device users is determined by four factors: completion time, energy consumption, security, and monetary cost.

Completion time: For resource-intensive tasks, the completion time is the overall time required by the system to complete the execution of the whole task[33]. It can be divided into two categories hard and soft real-time. Hard real-time is the amount of time it takes to complete a job in applications with strict real-time constraints. Such applications need a deterministic assurance that a job will be completed in a certain amount of time. Soft-real-time constraints indicate the length of time and the percentage of tasks that must be performed in a certain period of time. Soft-real-time limitations need statistical performance assurances.

Energy Consumption: The overall energy consumed on the mobile device to run an application is referred to as energy consumption[6]. This includes the energy cost of computing, data transmission energy, and energy consumed by other mobile components such as memory, storage, and sensors. The overall energy expended, on the other hand, is more than the sum of the energy expended by individual components. As a result, practical methods for evaluating energy use are required. Energy usage of the mobile device is taken into account.

Security: Security relates to maintaining the confidentiality of private data and the integrity of calculated data in offloading systems. Data is transferred to other computing resources using offloading. This data might include personal information about the user.

Table 2:List of metaheuristic techniques used for computation offloading in fog computing

Reference	Improvement strategy	Offloading	Performance Metrics	Nature of Task	Tool	Framework
[31]	Artificial Bee Colony	Where to offload	Service Time, Energy Consumption	Independent	Matlab	Edge Computing
[32]	Ant Colony Optimization	Where to offload	Response Time, Communication Cost	Independent	Matlab	Fog Computing
[33]	Firefly	Where to offload	Energy Consumption, Completion Time	Dependent	Matlab	Fog Computing
[34]	Artificial Fish Swarm Algorithm	Where to offload	Energy consumption, Computation cost	Independent	Matlab	Edge Computing
[35]	Fruit Fly Optimization	Where to offload	Energy Consumption	Dependent	Cloudsim2.0	Edge Computing
[25]	Non-sorting genetic algorithm and bees algorithm	Where to offload	Power consumption, delay	Independent	-	Fog Computing
[36]	Genetic Algorithm	Where to offload	Task completion time,	Independent	Matlab	Edge Computing
[3]	Particle Swarm Optimization	Where to offload	Cost, Resource Utilization	Independent	Matlab	Fog Computing
[37]	Whale optimization Algorithm	Where to offload	Time, Energy consumption	Independent	Matlab	Edge Computing
[38]	Smart ant colony optimization	Where to offload	Execution time	Independent	Matlab	Fog Computing

Table 3: Performance metrics used for computation offloading

References	Energy Consumption	Response Time	Task Completion time	Delay	Latency	Cost	Resource Utilization	Service Time
[31]	Y							Y
[32]		Y				Y		
[33]	Y		Y					
[34]	Y					Y		
[35]	Y							
[25]	Y			Y	Y			
[36]			Y					
[3]						Y	Y	
[37]	Y		Y					

Monitory Cost: There are two main components to the monetary cost:

Server Usage: The first component of the monetary cost is the cost of renting the server. The number of virtual servers and the kind of cloud VM utilized by the offloading system determines this.

Network Utilization: The cost of communication between the end device and the remote server is referred to as network utilization. The mobile service provider, the cloud service provider, or both may charge for this. It's the total of all incoming and outgoing traffic from the mobile device to the cloud[39].

The optimization metrics used in computation offloading problems are listed in table 4. Some of the matrices are energy-related, cost-related, and latency related, while others are related to tasks such as response time, service time, and completion time. As we can see in the table, energy-related matrices are mostly addressed. Approx. 48%(of reviewed work) of researchers have considered bijective metrics like [37] have used time and energy consumption while 35% have considered multi-objective optimization problem, but most are converted to single-objective optimization problem either by using weighted sum method or by making one parameter as a constraint. In the cost parameter, some researchers have considered communication cost and computation cost separately, while others considered the cost of offloading the task at the fog node or cloud node. In the same way, energy consumption is calculated based on task execution at the end-user, at the fog node, or in a cloud.

4. Interpretations

The following interpretation has been made based on this study:

(a) Many researchers have combined one meta-heuristic algorithm with another one. Generally, they removed the shortcoming of one algorithm by the strength of another algorithm. Authors in [40] have combined the Genetic Algorithm(GA) with the Particle Swarm Optimization algorithm(PSO) so that inefficient offloading can be avoided or the algorithm should not be premature into a locally optimal solution. Feng et al. [41] combined Grey Wolf Optimization (GWO) and Whale optimization algorithm(WOA) to optimize computation offloading in edge computing. [32] have combined ant colony optimization algorithm(ACO) and PSO for task offloading.

(b) Population-based metaheuristic algorithms can also be combined with Local search-based algorithms. Local Search techniques find out the optimal solution from the best regions in the search space, which are selected by meta-heuristic techniques. In this context, [42] has combined the greedy technique with PSO for computation offloading in edge computing.

(c) Some researchers have used transition operators to improve the quality of the solution. In the case of the firefly algorithm, various methods have been proposed for Levi flight updation, which affects the solution[43]. In the case of the Ant Colony Optimization technique (ACO), different methods have been used for pheromone updates. This influences the search strategy of ACO[30]. The various modified version of GWO has been proposed for offloading task in cloud computing, edge computing, and fog computing[44]–[47].

(d) The performance of metaheuristic algorithms can be improved by modifying initial population generation methods. The initial population can be generated using some greedy selection or local search techniques.

Population for upcoming iterations can be generated by selecting the best solution, or we can improve the output of the next iteration by considering the previous iteration's output.

5. Open issues

This section discusses the open issues and future research directions associated with computation offloading in fog computing. In the above section, we have seen how optimization techniques or swarm intelligence techniques were used to optimize offloading decisions because they are more appropriate for the dynamic and heterogeneous environment. Metaheuristic techniques provide an optimal solution in the shortest time while meeting the demands of applications. The analysis conducted in this paper revealed that a

Manuscript received October 5, 2025
Manuscript revised October 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.10.9

substantial amount of research had been conducted on offloading, particularly in fog computing. Moreover, various techniques like queuing models, linear or nonlinear programming have historically emphasized distinct objectives. However, it is critical to provide a solution that maximizes the number of objectives. The existing research recommends focusing on various parameters, including energy, payment cost, latency, delay, communication cost, and computation cost. It is critical to note that there are still several unresolved issues in the offloading process that the research community must investigate. These issues contain resource allocation for offloaded tasks, which may occur on fog servers or cloud servers. Offloading decisions must be based on the network's current context also because the network's nature is dynamic. Hence, there should be fault-tolerant mechanisms for resending lost data while minimizing response time and energy consumption. Additionally, offloading techniques should be automated to discover network areas, new nodes, and missing components. This will improve the efficiency of the offloading process. One of the most promising research directions is combining SI-based offloading theory with fog computing to enhance existing centralized solutions and adapt them to a distributed schema while considering multiple objectives. The ability of nodes to move around in the network is also important. So, it's possible that offloading choices could be better if we considered historical movements. This information can be stored in fog nodes for better offloading decisions and used to predict mobile locations. Improvements to the offloading process are possible with the help of prediction models and optimization theory. Lastly, offloaded tasks still raise privacy and security concerns as they travel through the network. Offloading decisions can also be made with the help of AI-based models that can foresee potential attacks.

6. Conclusion

A bibliometric analysis was conducted to find out the best journal, the most productive author, the contribution of countries and hotspots of fog computing to represent the popularity of the research topic. Then a brief overview of optimization techniques is given in detail. However, optimization techniques are widely used for the solution of various problems in different engineering fields. It was observed that optimization techniques could also be used for solving offloading problems in fog computing. In the literature, authors focused on various performance matrices like energy consumption reduction, service completion time, execution cost reduction, average resource utilization, etc. Based on the bibliometric analysis and study of optimization techniques, various interpretations are made, and several open issues were identified. Since fog computing technology supports IoT devices, there is a large scope for future research in the IoT environment.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

[1] W. Paper, "Prepare to succeed with the Internet of Things," pp. 1–9, 2017.

[2] S. Trab, A. Zouinkhi, E. Bajic, M. N. Abdelkrim, and H. Chekir, "IoT-based risk monitoring system for safety management in warehouses," *International Journal of Information and Communication Technology*, vol. 13, no. 4, pp. 424–438, 2018.

[3] M. Adhikari, S. N. Srirama, and T. Amgoth, "Application Offloading Strategy for Hierarchical Fog Environment Through Swarm Optimization," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4317–4328, 2020.

[4] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multi-objective Optimization for Computation Offloading in Fog Computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2018.

[5] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, "Fog Computing: A Comprehensive Architectural Survey," *IEEE Access*, vol. 8, pp. 69105–69133, 2020.

[6] J. Fang, J. Shi, S. Lu, M. Zhang, and Z. Ye, "An efficient computation offloading strategy with mobile edge computing for iot," *Micromachines*, vol. 12, no. 2, 2021.

[7] H. H. Elazhary and S. F. Sabbeh, "The W5Framework for Computation Offloading in

the Internet of Things," *IEEE Access*, vol. 6, no. c, pp. 23883–23895, 2018.
[8] F. Yu, H. Chen, and J. Xu, "DMPO: Dynamic mobility-aware partial offloading in mobile edge computing," *Future Generation Computer Systems*, vol. 89, pp. 722–735, 2018.
[9] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4804–4814, 2019.
[10] M. Zeng, Y. Li, K. Zhang, M. Waqas, and D. Jin, "Incentive Mechanism Design for Computation Offloading in Heterogeneous Fog Computing: A Contract-Based Approach," *IEEE International Conference on Communications*, vol. 2018-May, pp. 1–6, 2018.
[11] M. M. Hussain and M. M. S. Beg, "CODE-V: Multi-hop computation offloading in Vehicular Fog Computing," *Future Generation Computer Systems*, vol. 116, pp. 86–102, 2021.
[12] Y. Zhang, M. Kishk, and S. Alouini, "Computation Offloading and Service Caching in Heterogeneous MEC Wireless Networks," *IEEE Transactions on Mobile Computing*, 2021.
[13] S. O. Ogundoyin and I. A. Kamil, "Optimization techniques and applications in fog computing: An exhaustive survey," *Swarm and Evolutionary Computation*, vol. 66, no. January, p. 100937, 2021.
[14] N. Donthu, S. Kumar, D. Mukherjee, N. Pandey, and W. M. Lim, "How to conduct a bibliometric analysis: An overview and guidelines," *Journal of Business Research*, vol. 133, no. April, pp. 285–296, 2021.
[15] R. Colomo-palacios, "A systematic literature review of fog computing," vol. 24, no. 1, pp. 28–30, 2016.
[16] R. W. Cottle and M. N. Thapa, *International Series in Operations Research & Management Science Linear and Nonlinear Optimization*. 2017.
[17] M. M. Alqarni, A. Cherif, and E. Alkayal, "A survey of computational offloading in cloud/edge-based architectures: Strategies, optimization models and challenges," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 3, pp. 952–973, 2021.
[18] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann, "Evolutionary Algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 3, pp. 178–195, 2014.
[19] H. Ahmed and J. Glasgow, "Swarm Intelligence : Concepts , Models and Applications Technical Report 2012-585," *Technical Report*, vol. 585, no. February, pp. 1–50, 2012.
[20] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," vol. 1, no. 1, pp. 67–82, 1997.
[21] C. Sonmez, A. Ozgovde, and C. Ersoy, "Fuzzy workload orchestration for edge computing," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 769–782, 2019.
[22] M. G. R. Alam, M. M. Hassan, M. Zi. Uddin, A. Almogren, and G. Fortino, "Autonomic computation offloading in mobile edge for IoT applications," *Future Generation Computer Systems*, vol. 90, pp. 149–157, 2019.
[23] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog / Cloud Computing Systems with Min-Max Fairness Guarantee," vol. 6778, no. c, 2017.
[24] A. Bozorgchenani, S. Disabato, D. Tarchi, and M. Roveri, "An energy harvesting solution for computation offloading in Fog Computing," *Computer Communications*, vol. 160, no. March, pp. 577–587, 2020.
[25] M. Keshavarznejad, "Delay-aware optimization of energy consumption for task offloading in fog environments using metaheuristic algorithms," *Cluster Computing*, vol. 0123456789, 2021.
[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
[27] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
[28] M. O. Okwu and L. K. Tartibu, "Particle Swarm Optimisation," *Studies in Computational Intelligence*, vol. 927, pp. 5–13,

2021.
[29] A. H. Gandomi, X. S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.
[30] W. Bao, H. Ji, X. Zhu, J. Wang, W. Xiao, and J. Wu, "ACO-based solution for computation offloading in mobile cloud computing," *Big Data and Information Analytics*, vol. 1, no. 1, pp. 1–13, 2015.
[31] M. Babar, M. S. Khan, A. Din, F. Ali, U. Habib, and K. S. Kwak, "Intelligent Computation Offloading for IoT Applications in Scalable Edge Computing Using Artificial Bee Colony Optimization," *Complexity*, vol. 2021, pp. 1–12, 2021.
[32] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-Based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020.
[33] R. F. Abdel-kader, N. E. El-sayad, and R. Y. Rizk, "Efficient energy and completion time for dependent task computation offloading," pp. 1–22, 2021.
[34] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6398–6409, 2018.
[35] K. Lin, S. Pankaj, and D. Wang, "Task offloading and resource allocation for edge-of-things computing on smart healthcare systems R," *Computers and Electrical Engineering*, vol. 72, pp. 348–360, 2018.
[36] Z. Li and Q. Zhu, "Genetic algorithm-based optimization of offloading and resource allocation in mobile-edge computing," *Information (Switzerland)*, vol. 11, no. 2, pp. 1–13, 2020.
[37] M. Huang, Q. Zhai, Y. Chen, S. Feng, and F. Shu, "Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing," *Sensors*, vol. 21, no. 8, pp. 1–24, 2021.
[38] A. Kishor and C. Chakarbarty, "Task Offloading in Fog Computing for Using Smart Ant Colony Optimization," *Wireless Personal Communications*, no. 0123456789, 2021.
[39] S. S. N. Perala, I. Galanis, and I. Anagnostopoulos, "Fog Computing and Efficient Resource Management in the era of Internet-of-Video Things (IoVT)," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2018-May, pp. 1–5, 2018.
[40] R. Ezhilarasie, M. S. Reddy, and A. Umamakeswari, "A New Hybrid Adaptive GA-PSO Computation Offloading Algorithm for IoT and CPS Context Appplication," 2019.
[41] S. Feng, Y. Chen, Q. Zhai, M. Huang, and F. Shu, "Optimizing computation offloading strategy in mobile edge computing based on swarm intelligence algorithms," *Eurasip Journal on Advances in Signal Processing*, vol. 2021, no. 1, 2021.
[42] S. A. Zakaryia, S. A. Ahmed, and M. K. Hussein, "Evolutionary offloading in an edge environment," *Egyptian Informatics Journal*, no. xxxx, 2020.
[43] M. Adhikari and H. Gianey, "Energy efficient offloading strategy in fog-cloud environment for IoT applications," *Internet of Things*, vol. 6, p. 100053, 2019.
[44] G. Natesan and A. Chokkalingam, "An improved grey wolf optimization algorithm based task scheduling in cloud computing environment," *International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 73–81, 2020.
[45] A. Alzaqebah, R. Al-Sayyed, and R. Masadeh, "Task scheduling based on modified grey Wolf optimizer in cloud computing environment," *2019 2nd International Conference on New Trends in Computing Sciences, ICTCS 2019 - Proceedings*, pp. 1–6, 2019.
[46] K. Sreenu and S. Malempati, "FGMTS: Fractional grey Wolf optimizer for multi-objective task scheduling strategy in cloud computing," *Journal of Intelligent and Fuzzy Systems*, vol. 35, no. 1, pp. 831–844, 2018.
[47] M. Y. Mulge, "Optimization of task scheduling algorithm using modified mean Grey-Wolf," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 4, pp. 192–200, 2019.

Jyoti Yadav is Assistant Professor in Dronacharya Govt. College, Gurugram, and research scholar in CSED at Deenbandhu Chhotu Ram University of Science & Technology, Murthal, Haryana, India. She holds M.Tech degree from T.I.T&S college, Bhiwani, and B.tech degree from S.I.T.M college, Rewari. She has qualified for various national level exams like UGC-NET(JRF), and GATE(Thrice). Her research interest includes Fog Computing, Cloud Computing, and Wireless network.

Dr. Suman is working as Professor in CSED in Deenbandhu Chhotu Ram University of Science & Technology, Murthal, Haryana, India. She has been into teaching and research for about 17 years. She did her Ph.D. from Deenbandhu Chhotu Ram University of Science & Technology, Murthal, Haryana, India. Her research areas include Heterogeneous Wireless Networks, Wireless Sensor Networks, Cloud Computing, Internet of Things(IoT) and VANETs. She did her M.Tech in Computer Science & Engineering from Kurukshetra University, Kurukshetra, India. She has published more than 60 papers in various journals of repute.