



Towards Data and Computation Offloading in Mobile Cloud Computing: Taxonomy, Overview, and Future Directions

Amir Masoud Rahmani^{1,2} · Mokhtar Mohammadi³ · Adil Hussein Mohammed⁴ · Sarkhel H. Taher Karim^{5,6} · Mohammed Kamal Majeed⁷ · Mohammed Masdari⁸ · Mehdi Hosseinzadeh^{9,10}

Accepted: 29 January 2021 / Published online: 17 February 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

The rapid developments in the mobile application context illuminate the demand for more resources and processing power at Smart Mobile Devices (SMDs). Mobile Cloud Computing (MCC) enables the SMDs to offload their workloads on the remote cloud servers and benefit from the MCC's extensive resources to deal with this issue. To this end, numerous offloading schemes are provided in the literature to enhance the SMD's efficiency by offloading their workloads on the nearby cloudlets or remote cloud computing resources. This article puts forward a comprehensive survey and taxonomy of the offloading approaches designed and proposed for MCCs. It first classifies them based on the algorithms which have been used for making the offloading decisions. Then, in each category, it illuminates how the offloading decisions are made to improve application performance and mobile devices' energy efficiency, regarding offloading factors such as deadlines, costs, etc. The evaluation metrics, simulator, offloading type, and architecture of the studied schemes are compared and illuminated in each category. Furthermore, regarding the various properties of the studied offloading methods, the offloading domain's leading issues and challenges are discussed. Lastly, the concluding points are provided, and directions for the subsequent studies in the offloading context are specified.

Keywords Cloud computing · Offloading · Mobile device · Energy · Task · Stochastic · Markov

1 Introduction

Cloud computing is an interesting technology aimed to deliver uninterrupted and cost-efficient services for organizations by providing unlimited virtual resources for their customers [1, 2]. Virtualization is one of the keys the technologies have applied by the cloud service providers (CSPs) [3] to share the resource of each physical machine (PM) among multiple VMs. It facilitates the execution of the users' tasks on the VM's

✉ Mehdi Hosseinzadeh
hosseinzadeh.m@iums.ac.ir

Extended author information available on the last page of the article

isolated environment, and [4] enables the CSPs to increase their scalability and return on investment by proper resource management in their data centers (DCs) [5]. Mobile Cloud Computing (MCC) tries to provide cloud-based services for the customers equipped with smart mobile devices (SMDs) [6, 7]. Besides, regarding the small processing capability of the SMDs and their limited battery power, offloading and scheduling techniques are proposed to benefit the SMDs from the various virtual resources provided by the remote cloud servers [8]. Generally, offloading can be performed at the code, data, and application levels [9] to relieve the SMDs from the power-consuming CPU intensive applications whose demand is overgrowing because of the new emerging applications [10–12]. Also, data-intensive tasks that should access the high volume of data on the cloud repositories can be offloaded to be run on the CSP side, near the accessing data [13]. To reduce data delivery costs, it is critical to decreasing cellular network usage while satisfying QoS requirements [14].

Consequently, good offloading policies can benefit both CSPs and their customers by executing their requested tasks in the resourceful cloud computing data centers [15]. For example, offloading improves the battery life of the SMDs and relieves them from handling high resource-demanding applications and causing performance improvements in the SMDs [16]. One of the main advantages of offloading solutions is the improvement of SMDs' lifetime. Also, because applications will be executed on the cloud servers, in CPU-intensive job types, users can benefit from more cloud resources [17]. However, not all tasks should be offloaded, and not all offloading methods are energy efficient and even possible, considering constraints such as deadlines and channel conditions. Thus, effective decision-making must be made on the SMD to determine which parts of the programs should be offloaded [18].

Moreover, to be cost-effective, other factors such as the best destination site [19], monetary cost, energy, and delay incurred to send and receive the offloading request and response messages should be considered in the offloading decisions. Therefore, regarding numerous factors in the offloading process, it is considered an NP-hard problem [20]. Numerous states of the art offloading solution are provided in the literature [21–29], aiming to enhance SMDs capabilities towards handing a large amount of the data and CPU-intensive applications. Moreover, based on the access ability of the Wi-Fi links and various generations of the cellular networks, and even other SMDs, and various conflicting objectives different offloading policies can be conducted. For example, considering the destination to which offloading can be performed, the SMDs may decide to offload to a nearby cloudlet, remote cloud DC, or other SMDs. Thus, various states of the art schemes are provided in the literature, which tries to solve various kinds of offloading problems using different algorithms and techniques, discussed in other review articles such as in [30, 31].

This survey article provides an extensive study of the offloading schemes designed and adapted for various MCC environments. We first provide the required background information about the offloading context and illustrate multiple options and critical points regarding offloading. Then, we present a taxonomy of the studied schemes based on their applied algorithm and describe the architecture and significant contributions and merits of each offloading scheme. The offloading systems are compared, and their simulation factors, simulators, and various offloading features are investigated. Finally, the concluding issues and possible future investigation directions are highlighted. The rest of this offloading survey article is organized as follows: Sect. 2 provides background knowledge about the MCC offloading. Section 3 presents a taxonomy of the offloading schemes and puts forward an overview of the studied systems. Section 4 provides a comparison of the studied offloading approaches, and finally, Sect. 5 presents the concluding remarks and future research areas.

2 Research Methodology

This section elaborates on how the required research papers applied in this article are found, selected, and finally refined. In this process, we only used articles published in the sites specified in Table 1. In this phase, we define search terms for finding existing reviews and survey papers. For this purpose, we searched the following expressions in the google scholar site:

- offloading MCC survey
- offloading MCC review
- offloading mobile cloud computing survey
- offloading mobile cloud computing review

We found a few papers satisfying these conditions and referred to them in this paper's first section. For finding the main articles containing new proposals, the following search strings are applied in Google scholar:

- offloading MCC
- offloading mobile cloud computing

The results achieved from these searches are screened according to some exclusion criteria. For example, thesis, patents, and papers of international journals that were not from Table 1's publishers were removed. We found several interesting papers for conducting the study, which will be reviewed in the next section. Figure 1 depicts the number of the proposed offloading schemes proposed each year from 2010 up to 2019. As shown in this figure, the number of these schemes is increasing, and this context can be assumed an active research area in the security literature.

3 Preliminary

This section is aimed to study the knowledge required background and the main challenges for the offloading in the MCC environment. It illustrates the common architectures applied for offloading schemes and discusses their advantages. In general, offloading solutions should handle the following challenges in MCC environments [32]:

Table 1 Applied publisher

Index	Site	URL
1	IEEE Xplore	http://www.ieee.org/web/publications/xplore/
2	ScienceDirect—Elsevier	http://www.elsevier.com
3	SpringerLink	http://www.springerlink.com
4	ACM Library	http://dl.acm.org
5	Hindawi	https://www.hindawi.com/

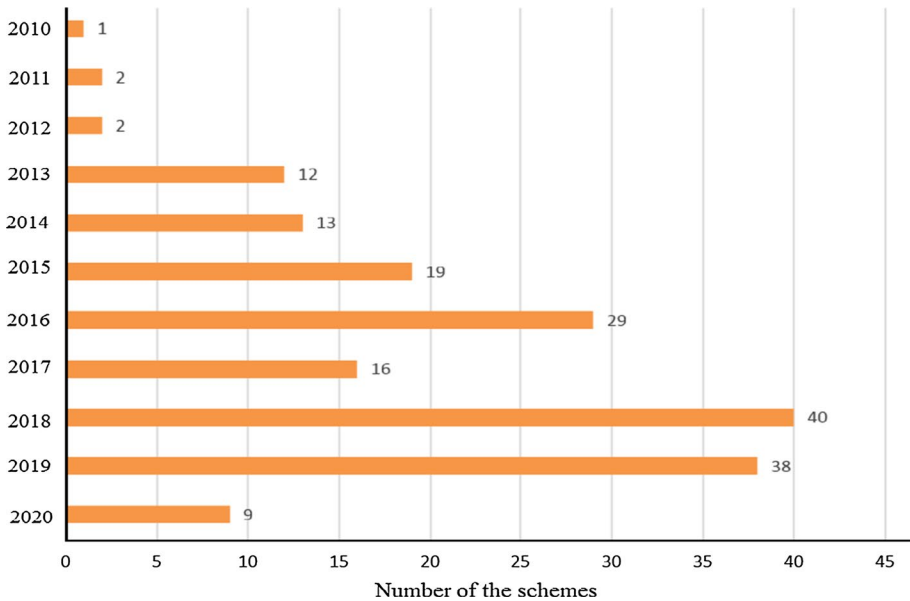


Fig. 1 Publication year of the offloading schemes for MCC

- **Inter-operability:** In the MCC environment, various SMDs may interact and connect using multiple networks to one or many servers. Offloading may be possible between different systems of the various computational capabilities.
- **Mobility**
- **Fault tolerance:** Offloading relies on wireless channels and intermittently accessible cloudlets [33] and cloud servers. Consequently, it is crucial to handle failures in the offloading process.
- **Security:** Because users' data is sent to remote cloud PMS, it should be protected during the transmission and the destination nodes that are not under the users' control.
- **Context-awareness:** An SMD should consider the user's state and infer the context information because the mechanism of offloading may vary according to the users' location and context [34]. Even for programs with dynamic decisions, the tasks that may potentially be offloaded are identified throughout the program development. Partitioning a program throughout execution is undesirable because of the very high overhead for analyzing the program.

The offloading schemes may use two-tier or three-tier architecture, in the first case of which, SMDs offload directly to the remote cloud. But, in the three-tier architecture, a nearby cloudlet or MEC can reduce the response time for the SMDs and mitigate the load on remote cloud servers. However, when the cloudlet cannot handle the workloads, it forwards them to the cloud for further processing.

Figure 2 indicates various features of the offloading process in MCC. For example, regarding the offloading granularity, the offloading schemes can be categorized into the following classes [7]:

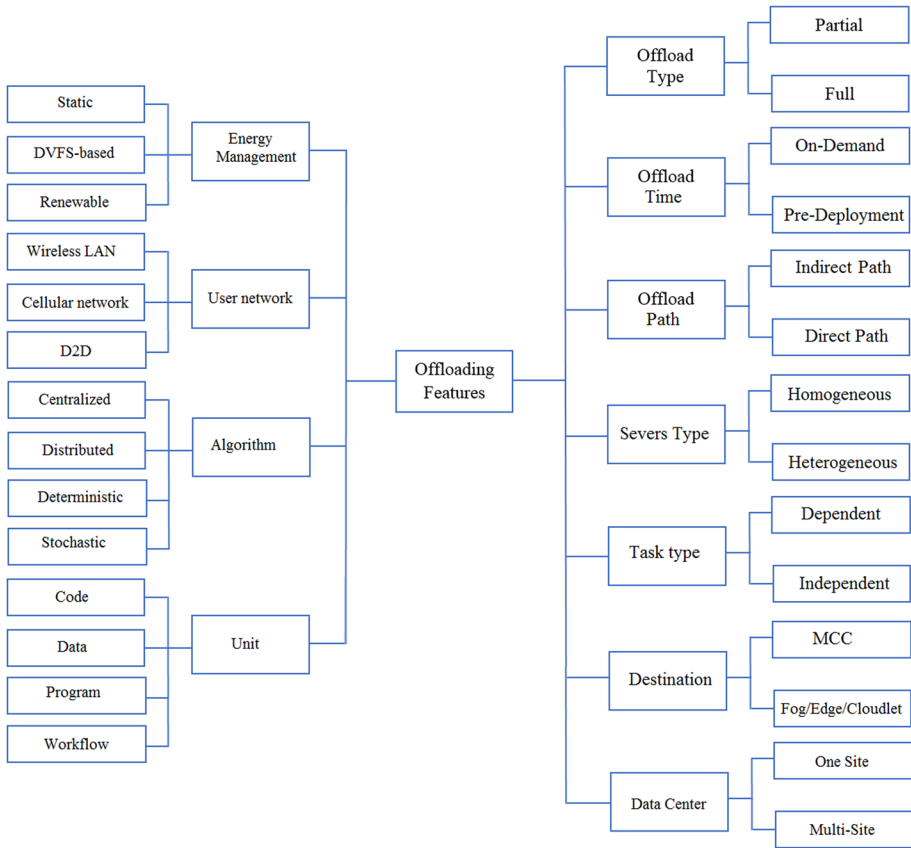


Fig. 2 Offloading features

- Full Offloading: The whole program is offloaded from an SMD to an offloading server. However, when the program’s size exceeds the components’ size, it could cause a high network overhead.
- Partial Offloading: Only a subset of a program or workflow will be offloaded. The interaction overhead between a client program on an SMD and the components offloaded on a server should be well-handled and may present a runtime overhead.

Moreover, offloading decisions can be made statically or dynamically by considering various factors [35]. Static offloading decisions are conducted during the development phase of the applications and are easier to implement, but it may not provide the promised benefits of offloading. However, dynamic offloading decisions are made during the runtime regarding various factors such as network coverage, SMD, etc. Although they require monitoring resource availability on the SMDs, they can adapt to different runtime conditions like fluctuating network bandwidths. These methods can use prediction algorithms but incur higher overheads. Moreover, offloading components can be transmitted in advance to an offloading server

(Pre-offloading), or they can be transferred dynamically at the offloading time (On-demand) [36]. The pre-offloading method obtains higher performance during the program execution, but it requires tuning the offloading server. The on-demand offloading method transmits the offloading components at the runtime and does not require a remote server's pre-configuration. But, the overhead of transmitting the required components is higher. Also, it requires searching for a proper server.

Regarding the offloading route, offloading schemes can be considered as direct and indirect schemes. Components can be offloaded directly to an offloading server or indirectly using an offloading repository. Direct transmission is simpler because it does not require a third party, but it suffers from high overhead and low performance. Indirect Transmission requires a third-party server to maintain a repository of the pre-stored components to be offloaded, which is faster, but the offloading component should be determined beforehand. Regarding delay constraints in the offloading components, the offloading operation can be categorized as delay-sensitive offloading and delay insensitive (delay tolerant) offloading. Generally, in delay-tolerant applications, response time is not critical, but improving power consumption is more important. Nevertheless, in delay-sensitive applications, low response time is an essential factor, and other offloading factors should be considered while the specified deadlines are met.

Moreover, offloading schemes may benefit from the 3G, 4G, or 5G cellular networks. Also, channel availability can always be connected or intermittently connected. As shown in Fig. 2, offloading schemes can be used to offload data, applications, and code. Furthermore, power management in offloading methods can be conducted as static, dynamic, or using renewable energy. In dynamic power management offloading solutions, it is aimed to reduce the energy consumption in the SMD side and destination side by using techniques such as DVFS. In this case, the SMD or destination node will be run with less energy while meeting QoS requirements and deadlines. Also, offloading schemes may consider applying renewable energy in the destination site, for example, on the cloud DCs. Besides, SMDs may select to offload their requests on the cloudlets, Fog computing, or MCC. Moreover, the offloading decision can be made locally or globally.

4 Proposed Offloading Schemes

Various offloading scaling schemes, such as [1, 5, 15, 28, 37–98], are provided in the offloading literature, and in this section, we only focus on the most critical approaches. This subsection is aimed to discuss the architecture of the offloading schemes and illuminate the factors considered in offloading the required items. Almost all these approaches aim to optimize their offloading policy to achieve objectives such as energy efficiency, meeting deadlines, and so on.

Figure 3 indicates the classification of the offloading frameworks, which will be investigated in this section.

- What are the significant contributions of each offloading scheme?
- What are the main algorithms and techniques applied by each offloading scheme?
- which evaluation factors and simulation environments are
- The offloading architecture, wireless standards supported by each scheme, and their limitations and merits.
- What are the remaining open issues and challenges in the offloading context in the MCC?

4.1 Fuzzy Logic-based Schemes

In [99], Wu et al. provide a multi-factor scheme that makes a multisite offloading decision and conducts offloading risk evaluation. They use fuzzy inference to aggregate the offloading benefits according to the MCC user's preference. Moreover, they apply an ACO-based algorithm to compute the assignment from application components to nearby sites. As an advantage, they consider factors such as reliability and privacy in the offloading process.

4.2 Game Theory-based Schemes

This subsection will provide a survey on the game theory-based offloading schemes such as [100–105], designed for the MCC environment. For instance, in [106], Cardellini et al. present a three-tier architecture that contains a local tier of SMDs, a cloudlet located at the WAPs but has limited resources, and a remote tier located in the CSP. It limits the use of distant servers that have higher latency. They consider a scenario without a central authority, while several SMDs share the limited resources of a nearby cloudlet and can send their computations to any tier. They provide a model to capture the user's interaction and find the effects of computation offloading on the user's performance. They consider the problem as a generalized Nash equilibrium and present a distributed algorithm for the computation of an equilibrium. But they only consider the case that offloading is used to improve the SMD performance and does not consider the loads which may exceed the SMD capacity.

The scheme presented in [107] puts forward MCIoT, a nested game model for efficient computation offloading. Each SMD finds the portion of remote offloading computation according to the Rubinstein game. This game-based technique is used in distributed IoT environments and supports deadline-aware application executions.

In [108], Chen et al. present a solution that applies game theory for offloading in MCC and applies the decentralized offloading decision-making among the SMDs. They exhibit that the game has a Nash equilibrium for homogenous and heterogeneous wireless access. The authors claimed that their method is efficient and scalable. Also, in [96], Qin et al. presented a game theory-based approach to attain the optimal performance among some users. Furthermore, in [91], the authors formulated the collaborative computing offloading problem for VANETs, which apply MEC and MCC. They try to optimize the computation of offloading decisions and computation resource allocation. They solve this problem by using a computation offloading strategy game, and an algorithm is developed, which can decrease the system complexity without affecting the performance. They indicated that their approach could achieve better results in terms of system utility and computation time.

The offloading method proposed in [109] investigated the computation offloading and radio resource allocation methods using a game theory-based approach. They considered the communication and computation cost to formulate a buyer–seller game to maximize the user's utility and sink. For finding the optimal offloading and resource allocation, the closed-form solutions are derived for the quasi-static and dynamic scenarios under uniform price and non-uniform price, respectively. An effective distributed price bargaining algorithm, named PJORA, with guaranteed convergence is proposed for the non-uniform-pricing case. They conduct experiments to determine the user's lower and upper bound and the base station's revenue.

In [110], Li et al. investigate the offloading rates of connecting to the same security cloud server under adaptable bandwidths and consider their solution as a non-cooperative

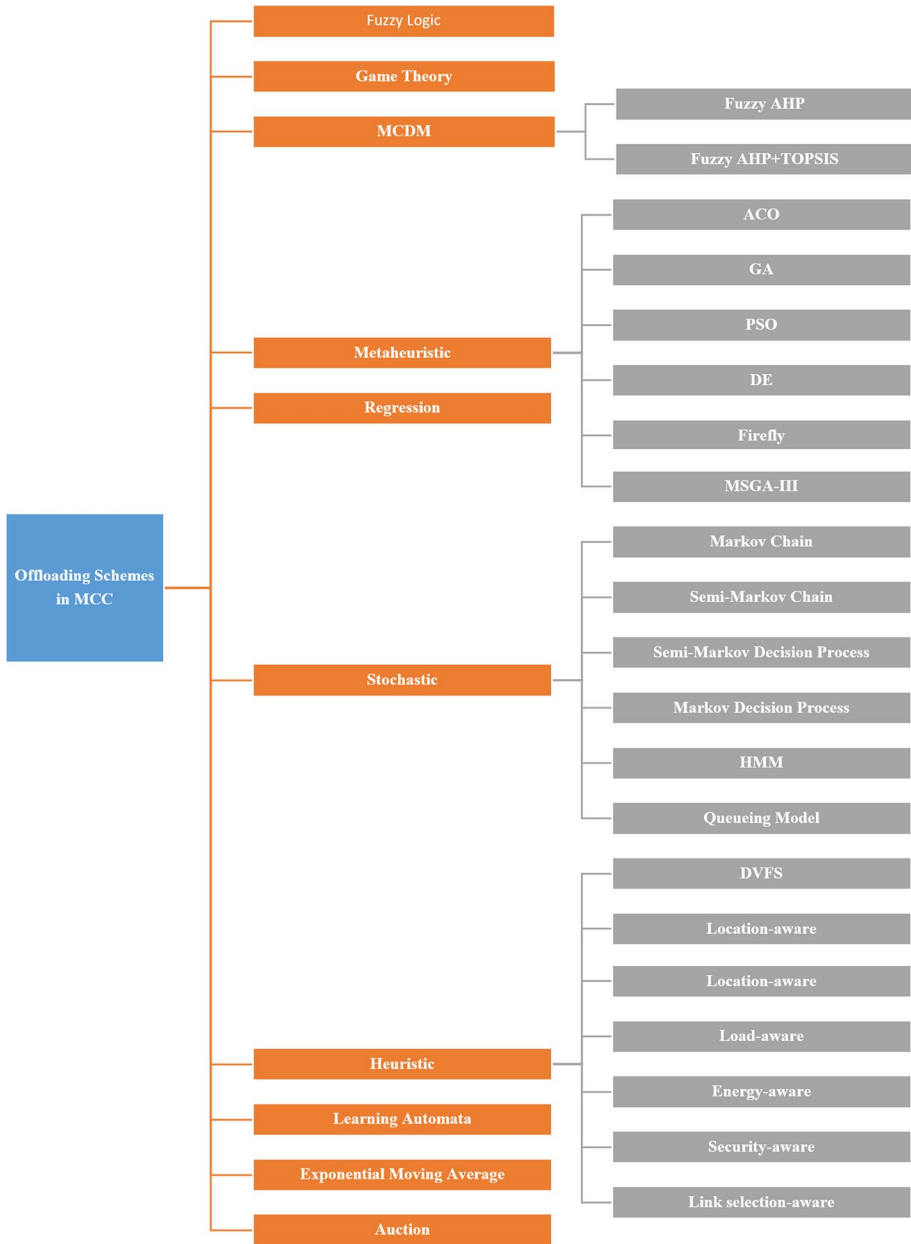


Fig. 3 Classification of the offloading and scheduling schemes in the MCC

offloading game. The Nash equilibrium of the MCC offloading game and the existing condition is presented. An offloading algorithm based on Q-learning is proposed for smartphones to determine their offloading rates for malware detection with unknown parameters

like transmission costs. Advantages include obtaining the optimum rate and enhancing the user's utility under adaptable network bandwidths.

In [111], the authors investigated the multi-user offloading problem for MCC, in which SMDs are active occasionally, and the status of the wireless channels change dynamically. They formulated the offloading decision of the SMDs using a stochastic game and indicated that it is equivalent to a weighted game that more than one Nash Equilibrium. They also propose a multi-agent stochastic learning algorithm to reach the Nash Equilibrium with a guaranteed convergence rate. They conduct simulations to validate the effectiveness of this algorithm and evaluate its performance in a dynamic environment.

In [112], Kuang et al. focused on the SMDs' energy-saving by offloading and formulated the multi-user offloading using game theory. They proposed an offloading game consisting of two algorithms denoted as BOT or beneficial offloading threshold and BOG or beneficial offloading group. BOT algorithm can obtain the maximum number of SMDs that each SMD can tolerate to offload their tasks, and BOG finds beneficial offloading SMDs group. The authors proved that their model has a Nash equilibrium and also, by conducting the required evaluations, indicated that their scheme could achieve energy-saving via offloading.

In [41], the authors attempted to deal with the offloading problem in vehicular cloud computing environments. They introduced a two-player Stackelberg game to deal with computation offloading and consider factors such as service price and task completion duration. They derived the offloading game's Stackelberg equilibriums and, using the Monte-Carlo simulations, indicated that their method could mitigate the task execution time while increasing the service providers' profit.

The offloading approach proposed in [113] introduces a four-tier architecture to determine the specific decision-maker for the task offloading and considers factors such as time and energy consumption. They formulated the offloading problem as an evolutionary game and solved it using the Maynard replicator dynamics. They simulated their scheme in MATLAB and applied 100 python tasks in their experiments. They then indicated that their scheme could reduce traffic and convergence time.

In [114], Shen et al. proposed DOM, a dynamic task offloading method for cloud-edge computing using minority game, a useful tool containing a distributed mechanism to minimize the dependency on centralized control in resource allocation. In the minority game, reinforcement learning is applied to optimize the distributed decision-making of participants. The authors analyzed the effectiveness of their scheme using a real dataset.

In [115], a distributed cloudlet system is established consisting of many cloudlets and computations which select appropriate cloudlets. They formulated this dynamic offloading problem as a transferable coalition game and to achieve a successful offloading. They considered factors such as user-mobility predictive probability, energy, cloudlet, and channel availability. The authors conducted the required experiments and demonstrated that their dynamic offloading approach achieves a better result with low complexity.

4.3 MCDM-based Offloading Schemes

This subsection will provide a study of the MCDM-based offloading schemes that apply algorithms such as AHP, fuzzy AHP, and TOPSIS in the offloading process.

4.3.1 Fuzzy AHP-based Offloading Schemes

In [116], Wu et al. present an MCDM-based approach using the AHP and the TOPSIS methods to select the destination CSP. The AHP is used to specify the weights of the criteria for cloud-path selection, and fuzzy TOPSIS is used to rank CSPs. Different criteria such as speed, bandwidth, security, price, and availability are in the selection process. However, they have not compared their approach against other offloading solutions.

4.3.2 AHP + Fuzzy TOPSIS Based Schemes

In [117], Singla et al. put forward a fuzzy AHP approach to select a CSP to offload data by determining the weights of five criteria and obtaining the CSPs' final ranking. They indicate that availability and speed are significant and show the optimum CSP where the users can migrate their computation-intensive tasks to reduce execution time and with the least overhead. But, they have not compared their scheme with other offloading solutions to highlight their solution merits. Table 2 provides the properties of the fuzzy logic, game theory, and MCDM-based offloading schemes outlined in this section.

4.4 Meta-heuristic Based Offloading Schemes

This subsection provides a review of the meta-heuristic [118] based offloading schemes applied for the MCC environment. It classifies them according to the meta-heuristic algorithms used to optimize objectives such as energy consumption, latency, and performance.

4.4.1 ACO-Based Offloading Schemes

In [65], Wang et al. introduce CMSACO, a cooperative multi-tasks scheduling based on ACO, which considers factors such as task profit, deadline, heterogeneity, and load balancing. They try to increase the profit considering constraints like resource usage, executive order, and completion time. The task-flows are classified into independent tasks, linear chain, tree structure, and mesh structure. But, this scheme only conducts static scheduling and cannot be used for online scheduling.

The scheme presented in [119] introduces HLMCM, a hybrid local MCC model with the application scheduling structure. They provide a scheduling algorithm for the MCC-based on the MAX-MIN ant system. Nevertheless, they have not considered the applications' dynamic resource requirements.

In [102], Guo et al. provided a task scheduling method called ACO to improve the performance of the SMDs in the cloud environment and Hadoop. This ACO-based algorithm conducts dynamical task scheduling and enhances throughput and QoS.

4.4.2 GA-Based Offloading Schemes

Several GA-based offloading solutions, such as [58, 119, 120], are introduced for MCC. For instance, in [121], the authors provide a scheduling solution based on DNA combinations and GA processing under the precedence level. It decreases energy usage and task execution time while meeting the deadlines. It provides reliability by retrieving the

Table 2 Properties of the fuzzy logic, game theory, and MCDM-based offloading schemes

Evaluation factors	Simulators/Environments										Properties			Offloading type											
	Energy consumption	Execution time	Cost	Off-loading rate	Performance tuning	Speed execution	Util-ity	Secu-rity	Avail-ability	Profit	Delay	Band-width	Rev-enu	Mat-lab	Cloud-sim	Amazon	Real environment	Two-tier Architecture	Three-tier Architecture	Multisite	Predictive	Data off-loading	Code off-loading	Com-puting offload-ing	
[99]				✓										✓											✓
[105]			✓												✓					✓					✓
[106]		✓																							✓
[107]		✓																							✓
[108]			✓																						✓
[96]			✓						✓																✓
[91]										✓															✓
[109]			✓																						✓
[110]																									✓
[111]			✓																						✓
[112]			✓						✓																✓
[41]			✓																						✓
[115]		✓																							✓
[116]			✓																						✓
[117]			✓																						✓

processed data by the SMD and prevents task failure in MCC. Nevertheless, they have not evaluated their solution.

In [29], Goudarzi et al. present GAMCO, which applies GA for determining the optimum partitioning for offloading to the multisite clouds and mitigate makespan. GA operators are enhanced to achieve the best possible solutions. In this scheme, a chromosome represents a participant application, and each gene indicates a cloud server or local site.

In [122], Kaushi et al. considered mobile applications' partitioning. They used GA to find an optimum application partitioning and offloading solution to reduce the waiting time, communication energy, and computation energy by considering various SMD hardware components to minimize power consumption.

The scheme presented in [123] indicates a solution to handle user movement in the MCC. It uses a mobility model to provide a fault tolerance solution to optimize the computation offloading for workflows. They find a near-optimum solution for this problem using a GA-based method. As an advantage, this scheme predicts the user's mobility path using a Markov chain.

The offloading scheme presented in [85] introduced a technique to estimate the tasks' completion time in a wireless network with high accuracy. An enhanced GA using the 2D chromosome is provided to handle multi-objective task scheduling.

The scheme presented in [124] puts forward a GA-based approach for a heterogeneous multisite offloading problem that considers factors such as battery usage and execution time for computation offloading. Nevertheless, they do not evaluate it against other offloading solutions to highlight its merits.

In [125], Tout et al. presented an offloading approach that monitors resource usage and increases the performance of the SMDs. This scheme can reduce the resource consumption overhead and power usage of the applications running in each persona, accelerating their execution and improving their scalability, allowing better adoption. By adapting multi-objective optimization, this scheme can mitigate memory usages, execution time, and CPU of the application and improve the performance and viability. Although their approach can save SMD power, it does not employ DVFS on the SMD-side, further reducing power consumption.

4.4.3 PSO-Based Offloading Schemes

In [126], Manukumar et al. introduced an efficient multi-objective offloading decision-making scheme regarding the complexity of mobile applications' and SMD context. Moreover, the PSO algorithm was explored dynamically to identify the optimal cloud by considering makespan and VM cost. This method improved the performance of the computational task by reducing the battery power consumption and makespan.

In [127], Wang et al. introduced a task offloading method for MCC that considers factors such as energy, deadline, and channel constraint. They introduced NAIWPSO, adaptive inertia weight-based PSO, to create a channel constraint-based method to achieve a near-optimal offloading decision. They indicated that their task offloading method outperforms other schemes regarding the metrics such as SMDs' power usage, makespan, and the algorithm's running time by conducting the required experiments.

In [128], Alli et al. employed a neuro-fuzzy model for securing data at the smart gateway. In this scheme, the SMDs select fog nodes using PSO for offloading their loads. However, when the fog node cannot handle the load, it forwards the load to the MCC. In this offloading scheme, the authors computed fog computing nodes' availability

regarding factors such as processing capabilities and power. Besides, in this scheme, reinforcement learning is used for cloud selection. The authors evaluated their scheme with JAVA Programming and NS-3 simulator. They compared their secure offloading model and indicated that their scheme reduces latency.

4.4.4 DE-Based Offloading Schemes

Several offloading schemes have been proposed using the differential evolution (DE) algorithm. In [40], Huang et al. proposed an energy-aware computation offloading using cloudlets to mitigate high latency in offloading to remote clouds. They tried to improve the offloading method of workflows using the DE algorithm to reduce the power usage of the SMDs with time constraints.

In [129], Huang et al. proposed NSDE, a non-dominated sorting DE algorithm for handling offloading in MCCs. This scheme models applications as a constrained multi-objective optimization problem and applies the NSDE algorithm to solve it while minimizing the multimedia workflows' power usage. By conducting several experiments, they verified the validity of their method.

Table 3 presents the properties of the offloading schemes which have applied metaheuristic algorithms.

4.4.5 Firefly-Based Offloading Schemes

The offloading scheme presented in [130], establishes an approach for the fog-cloud environment and handling delay-intensive and resource-intensive applications with an optimal task offloading strategy. They proposed an offloading method based on the Firefly algorithm and tried to minimize factors such as power usage and computational time. They show that their approach performs other metrics energy, CO₂ emission, computational time, and temperature.

4.4.6 NSGA-III Based Offloading Schemes

The offloading scheme proposed in [131], Xu et al. introduced COM, a computation offloading method for cloud-edge computing that considers factors such as energy usage and execution time for SMDs. This scheme applies the NSGA-III algorithm to deal with the multi-objective task offloading.

In [132], Xu et al. presented MOC for computation offloading on the cloud-edge computing environments. This scheme presents a vehicle-to-vehicle routing algorithm. In this scheme, when the edge computing environments are overloaded, the offloading will be conducted on the clouds. Besides, NSGA-III is applied for multi-objective optimization aiming for load balancing in edge computing and decreasing the power consumption and tasks makespan. They used the multiple criteria decision making and additive weighting to evaluate their achieved solutions by the NSGA-III. At last, by conducting the required experiments, the authors validated the efficiency of their offloading approach.

In [133], the authors provided MCO, a multi-objective computation offloading method. They considered an energy usage model for the SMDs in the cloudlet-based MCC. An improved NSGA-II algorithm for handling computation offloading and achieving energy

Table 3 Properties of the meta-heuristic based offloading schemes

Scheme	Evaluation factors										Simulators/environments				Properties				Offloading type									
	Energy consumption	Resource utilization	Execution time	Cost	Profit	Completion time	Transmission time	Data transfer time	Response time	CPU usage	Delay	Makespan	Matlab	CloneCloud	Amazon	Java	Real environment	Cloudsim	Two-tier architecture	Three-tier architecture	multitasking	Fault-tolerant	Predictive	Data offloading	Code offloading	Computing offloading		
[65]	✓			✓							✓		✓						✓							✓		
[119]	✓							✓								✓				✓							✓	
[102]			✓												✓					✓								
[121]	✓		✓			✓																✓						
[122]	✓		✓										✓															
[123]	✓		✓									✓										✓						
[88]	✓			✓																								
[134]				✓																								
[124]	✓		✓										✓															
[125]	✓		✓							✓																		
[135]	✓		✓																									
[126]	✓		✓																									
[127]	✓		✓																									
[129]	✓		✓																									
[40]	✓		✓																									
[129]	✓		✓																									
[130]	✓		✓																									
[131]	✓		✓																									
[132]		✓																										
[133]	✓																											

saving for SMDs, considering the workflow deadlines. At last, several experiments are conducted to indicate the effectiveness of their offloading approach.

4.5 Regression-Based Offloading Schemes

This subsection provides a review of the regression-based offloading systems designed for the MCC environment. These solutions are aimed to apply regression-based prediction techniques to enhance the performance and effectiveness of the offloading solution. For example, in [136], Dhanya et al. try to decide to offload according to the current context and the SMD's optimization choice. They make offloading decisions based on regression analysis, which takes less time and energy.

In [137], Khoda et al. present ExTrade, a code offloading system that uses a 5G cellular network and reduces the applications' response time and battery power. They make offloading decisions using the Lagrange multiplier and introduce a statistical regression-based model to estimate the task execution time considering the adaptable environment and application usage. However, their solution only supports one server and does not discuss the availability of multiple offloading servers.

4.6 Stochastic Schemes

This subsection discusses the offloading schemes designed for the MCC environment, which have applied stochastic models to improve the offloading systems' various properties.

4.6.1 Markov Chain-Based Offloading Schemes

Numerous Markov chain-based schemes such as [18] are employed for offloading in MCC, which this subsection discusses. For example, in [59], Meng et al. introduce a secure and cost-aware offloading method that applies a hybrid continuous-time Markov chain and a queuing model. They indicate the variance of random delays reduces the random padding effectiveness, and the measurements of an attacker should grow linearly with the standard deviation of the random delays.

The scheme presented in [138] offers an offloading game for an SMD that tries to select its offloading rate, an attacker that selects its malicious behavior mode, and a security agent aimed to protect the offloading process. They derive the Nash and Stackelberg equilibrium of the offloading game and provide a Q-learning based offloading solution for SMDs. This offloading scheme improves the utility of the SMD and reduces the malicious behavior rate.

4.6.2 Semi-Markov Based Offloading Schemes

In [139], the authors try to offload the local load, consider the adaptable executions of applications using an order- k semi-Markov model, and decide about offloading regarding probabilistic estimations of the offloading operation's energy saving. Their idea is to consider these patterns using the semi-Markov model and find the transitions according to

Markovian states' sojourn time distributions. As an advantage, this scheme's effectiveness is evaluated using experiments over Firefox, ChessWalk, and barcode applications.

4.6.3 Semi-Markov Decision-Based Schemes

This subsection provides a survey on the SMDP-based offloading schemes specially designed for the MCC environment. For instance, in [140], Wang et al. present an SMDP-based offloading solution that applies the bus-based cloudlets as a destination and considers it a delay-constrained shortest path problem. Then, they improve the performance of SMDs by mitigating the cost and increasing the offloading rate.

The approach presented in [141] performs wireless transmission scheduling to balance SMD's power efficiency and responsiveness. They consider causality and application runtime when delaying wireless transmissions to reduce the wireless energy cost and meet the application deadlines. It improves the power efficiency of the offloading over realistic SMDs.

In [134], Chen et al. apply DVFS in the SMD's local processor, where the modulations and bit rates of RF transmitter can be tuned. An accurate battery model is used to forecast the energy loss rate. It introduces an SMDP-based optimization approach, with the support for various DVFS levels, to mitigate the average latency in request servicing and power usage. The advantages include finding the optimum DVFS setting and offloading rate, using linear programming and a heuristic method.

4.6.4 Markov Decision Process-Based Schemes

This subsection reviews the offloading schemes which utilize MDP for in [86, 142, 143], the framework presented in [144], investigates data offloading in MCC, using the cellular network, WiFi, and SMD to SMD communications. It offloads a part of the data traffic via WiFi and D2D networks. They consider the data offloading as a finite horizon MDP and solve it using a hybrid offloading algorithm for both latency-sensitive and latency tolerant program.

In [145], Terefe et al. put forward a model for energy usage of multisite execution and use a DTMC model for fading wireless channels. They also apply an MDP model to consider the multisite partitioning as a delay-sensitive and least-cost shortest path problem on a transition graph. Their multisite offloading solution is built on a value iteration algorithm and considers the site capabilities to distribute components to mitigate data transfer power costs.

The work in [146], applies conceptual models by blending the positive outcomes of both methods using MDP to provide conceptual analysis. As an advantage, they introduce a hybrid solution using a Markov chain model for selecting the offloading server.

In [147], the authors present an adaptable offloading approach to decide offloading for a set of parallel tasks. They solve the MDP model for the user SMD to achieve an optimum offloading policy to reduce the processing and offloading cost. They improve the MDP model to a constrained MDP to handle the offloading problem while meeting the deadlines. Nevertheless, the cost model of this scheme should be tuned to deal with more realistic scenarios.

The work in [98] introduced a hybrid data offloading model, where SMD can use the WiFi network and D2D communication to offload mobile data. They formulated the mobile data offloading problem as an FHMDP and introduced a hybrid offloading algorithm for

delay-sensitive and delay-tolerant applications. They showed that their scheme achieves low offloading costs and increases the offloading ratio.

The scheme proposed in [148], presents MDPCO, in which the MDP method is applied for handling offloading computation while minimizing the power consumption by optimizing the offloading decisions and resource allocation. They indicated that the MDPCO algorithm could outperform other algorithms under different parameters.

In [149], the authors proposed GM-MCC or genetic Markov MCC for dealing with multisite computation offloading in MCCs that considers the environment changes in the lifecycle of the applications. This scheme uses the MDP to find the best possible decision for offloading and allocate application components to the target site by consuming less energy of SMDs by considering the cost and overhead factors. This model utilizes a GA-based algorithm to tune the MDP parameters to achieve the highest benefit. The authors indicated that this model considers different capabilities of sites to allocate appropriate components. There is a lower energy cost for data transfer from the mobile to the cloud.

4.6.5 Hidden Markov Model-Based Offloading Schemes

This subsection will provide a survey on the HMM-based offloading schemes, such as those designed for MCC. In [150], the authors propose an HMM-based scheduling solution to improve the processing delay accuracy and power usage in MCC. To identify the optimum configuration for various requirements, they consider network status, SMD battery, and CPU load as the hidden states. The SMD power efficiency, processing accuracy, and execution performance are used as outputs from the hidden states. They apply the Baum-Welch algorithm for the HMM learning process and introduce an HMM for each operation setting. By sorting the hidden states' output probability, they make an optimum based on the statistical features.

4.6.6 Queuing Model-Based Offloading Scheme

Queuing theory is one of the interesting analytical analysis techniques for modeling offloading approaches in the MCC. This subsection will review the Queuing Model-based Offloading schemes such as [8, 61, 101, 151–155] and their key contributions in the MCC. The approach presented in [156] provides an analytical queuing model for delay offloading with intermittent connectivity links, which consider deadlines to detect failures and service interruptions. As an advantage, they analyze power-saving and latency mitigation. Besides, in [157], Wang et al. develop a dynamic application offloading decision-transmission scheduling scheme using numerous target clouds. They use the Lyapunov optimization to have a low-overhead dynamic offloading scheduling solution. But, conducting more experiments are needed to evaluate it against other offloading solutions.

In [93], the authors proposed DTO, a lightweight distributed scheme to handle the workload offloading with deadline constraints in MCC. They indicated that their algorithm could meet the computational tasks' deadline and achieve a near-optimal system-wide utility. Furthermore, they conducted experiments on the SMDs to exhibit the effectiveness of their approach. Table 4 specifies the evaluation factors, the simulators, and environments applied in the regression-based offloading schemes.

4.7 Heuristic-Based Offloading Schemes

This subsection provides a review of the heuristic offloading approaches, such as [94], designed for the MCC. To be more specific, they mainly focus on the following categories of the offloading methods:

- Energy-aware schemes
- DVFS-based schemes
- Load-aware schemes
- Location-aware schemes
- Security-aware scheme
- Link selection-based schemes

In [97], Han et al. analyzed the outage probability and bottleneck for mobile computation offloading. They considered different offloading scenarios with task division, channel allocation methods, and the outage probability of the computation and transmission. The outage bottleneck is determined by the power usage ratio with high power consumption. However, they have not considered issues such as bottleneck and order of tasks.

Besides, in [158], Salehan introduced CMOD, an online context-aware mechanism to save power and reduce execution time while making context-aware offloading decisions. They use a nonlinear programming model defined to maximize the objective function. They indicated that their solution demonstrates satisfactory performance. However, they assumed that only one user intends to offload her tasks, which is not realistic.

In [159], the authors focused on the task offloading on a two-tier MCC and formulated this problem as a Non-linear binary integer programming. They proposed a distributed linear relaxation method using the Lagrangian decomposition method. They proposed a greedy heuristic to select the best cloudlet and allocate bandwidth. They also conducted the required experiments and considered factors such as CPU resources, different users, and applications with varying resource and traffic patterns. The authors indicated that their scheme outperforms other approaches.

4.7.1 DVFS-Based Schemes

The DVFS-based solution reduces power consumption by reducing the CPU's operating frequency, considering the specified deadlines [160].

The approach introduced in [55] tries to balance the execution time and power consumption by using DVFS in the local processor in the SMDs. The rate capacity impact of a battery and power conversion losses in the SMD is used to have a realistic model of the remaining battery life. The SMD is modeled using SMDP and the optimization issue to set the DVFS level. The transmission rate is resolved by linear programming aggregated with a one-dimensional heuristic search.

The solution presented in [25] proposes a DVFS-aware offloading algorithm to reduce makespan and migrate tasks between the local cores and the cloud. It determines the tasks that should be offloaded, tunes the frequencies for running local tasks, and schedules all tasks on the cores and the wireless channels. The precedence requirements and completion deadlines are met, and the SMD's power consumption is reduced.

Table 4 Properties of the Regression-based Offloading and Stochastic offloading schemes

Scheme	Evaluation factors													Properties			Offloading type											
	Energy consumption	Delay	Execution time	Security	Completion time	Average utility	Application completion ratio	Offloading ratio	Cost	Response time	Accuracy	Computational overhead	Execution energy	Response time	Cloud	Matlab	Real environment	Android	Amazon	OMNeT++	Two-tier architecture	Three-tier architecture	Multi-tier architecture	Pre-live	Data offloading	Code offloading	Computing offloading	
[136]	✓								✓							✓						✓						✓
[137]	✓		✓						✓								✓					✓						✓
[18]	✓	✓													✓							✓						✓
[59]				✓																✓								✓
[138]					✓															✓								✓
[139]	✓		✓								✓																	✓
[140]	✓				✓							✓																✓
[141]	✓											✓																✓
[134]	✓																											✓
[144]	✓				✓																							✓
[134]	✓																											✓
[146]	✓			✓																								✓
[147]	✓																											✓
[148]	✓																											✓
[98]	✓															✓												✓
[150]	✓				✓																							✓
[156]	✓												✓															✓
[157]	✓																			✓								✓
[93]			✓																									✓

In [161], Balakrishnan et al. employ DVFS in mapping and scheduling stages by considering both the task-resource and resource-frequency assignments. They estimated the worst-case global slack time for each task-resource assignment, distributes it over the Task Interaction Graph (TIG), and slowing down the execution of tasks using DVFS. This optimal slowdown increases the computation time of TIG without exceeding its worst-case completion time. This scheme model the code offloading as a Quadratic Assignment Problem and solves it using a two-level GA. The authors indicated that this scheme could reduce the power usage of SMDs.

In [162], Guo et al. introduce a distributed offloading and resource scheduling algorithm to reduce energy consumption and completion time. It contains three algorithms for computation offloading selection, clock frequency tuning, and transmission power assignment. They find that the computation offloading selection depends on the computing load, its predecessors' maximum execution time, the clock frequency, and the SMD's transmission power. Advantages include reducing the energy consumption by DVFS-based CPU clock frequency tuning in the SMD and selecting the wireless channel's transmission energy.

4.7.2 Load-Aware Offloading Schemes

In [87], Shakkeera et al. present E-MACS, a power-aware SMD application scheduling algorithm for hybrid cloud. The application's consolidation reduces the power usage in the cloudlet, response latency, and migration cost. It improves factors such as QoS, throughput, and scalability among resources by load balancing.

4.7.3 Location-Aware Offloading Schemes

This subsection provides a survey on the location-aware offloading schemes such as [163], designed for MCC. For example, the solution presented in [164] proposes a level offloading mechanism that considers the mobility of the SMD and finds the best cloud instance. The application is divided at migration points where the execution thread is performed from the SMD to the cloud to improve performance and power cost. The SMD can finish the execution after the partitioned thread returns back to the SMD.

4.7.4 Energy-Aware Offloading Schemes

Many energy-aware offloading approaches, such as [81], are designed for MCC, which this part of the paper discusses.

In [44], Shiraz et al. propose EECOF, a distributed solution for the CPU-intensive programs in MCC, which reduces the data transmission size and energy cost in computational offloading. EECOF addresses energy usage and tries to mitigate the overheads of component offloading. But, the consistency of the simultaneous execution between the SMD and remote cloud server is not considered in this approach.

In [33], Lee et al. put forward SCCE, an energy-aware task scheduling in MCC, to mitigate the energy usage. Regarding the specified deadline, they apply a collaborative execution method. A threshold of the data size is employed for computation offloading and use the adaptive threshold to use degenerated Monte Carlo estimation.

The solution presented in [160] provides EETS, a power-aware task scheduling, and offloading in MCC. They evaluate the scheduler utilizing an Android smartphone to select the right action and mitigate energy usage. They present a task allocation method using the task data size, storage path, and an energy usage model. As an advantage, they apply a data compression method to reduce power consumption for the transferred data.

In [155], Zhang et al. provide EDTS, a power-aware scheduling solution to reduce the power usage of the SMDs regarding the specified deadline.

Besides, the offloading framework in [165] introduced an adaptive offloading model for data stream applications and performed code partitioning at thread-level. The application threads were further categorized as CPU intensive and I/O intensive tasks. This algorithm increases data stream applications' performance concerning makespan, power usage, computational load, and bandwidth. Finally, this scheme's performance is analyzed based on different resource models, but they have not considered data-intensive applications.

4.7.5 Security Aware Offloading Scheme

MCC is vulnerable to some security attacks. This subsection provides a survey on the security-aware offloading schemes, such as [7, 166–169] designed for the MCC environment.

In [35], Mahmoodi et al. provide a privacy-aware solution for the SMDs for server-based scheduling. Their method uses the holomorphic properties of cryptosystems to compute user availabilities. They specify the privacy requirements in such scheduling applications and use their solution on the real SMDs. They also indicated that their approach could satisfy the privacy properties and consider computation and communication factors.

The approach provided in [170] presents the OpenCL-based remote offloading solution for the SMD applications, which applies VPN to reduce the execution time and energy consumption. They implement their remote offloading framework prototype on an Android-based SMD and provide an adaptable resource discovery solution. The SMD is permitted to find the resources by IP multicasting via a virtual private network link.

In the offloading approach introduced in [171], Fadaraliki et al. propose a multi-agent approach on the Android-based SMD. This approach allows for the development of computation-intensive applications. Agents in the system are aware of resource limitations, battery power, and CPU consumption and tune their behaviors to obtain a tradeoff between resource consumption and security expectations.

4.7.6 Link Selection-Based Offloading Schemes

The offloading method in [135], Goudarzi et al. consider the power-aware link selection as a discrete stochastic adaptable program to improve the system throughput. As an advantage, they apply an approximate flexible programming algorithm that does not need the statistics of stochastic information.

The offloading approach, introduced in [172], presented a computation offloading method using deep learning for vehicular edge-clouds. This scheme considers the computational offloading and resource allocation and formulates the problem as a binary optimization aiming for minimizing the system cost. It uses a distributed deep learning algorithm that deep networks are used in parallel for achieving optimal computational

offloading. The authors proved that their approach could provide convergence factly and reduce resource consumption.

4.8 Learning Automata-Based Offloading Schemes

In [173], Krishna et al. present LADM, a reinforcement-based algorithm for offloading MCC tasks, which applies computation time and power usage for making the offloading decisions. This scheme supports the parallel execution of the job in mobile and cloud while reducing the job's overall cost.

4.9 Exponential Moving Average-Based Offloading Schemes

The framework provided in [56] introduces Phone2Cloud, a computation offloading method for power saving on the SMDs in MCC. Its objective is to improve the SMD's power efficiency and increase the application performance by mitigating its execution time. They evaluate their scheme on an Android and Hadoop environment, but not with other systems. Although this scheme provides semi-automatic offloading, it does not support automatic offloading. Table 5 specifies the simulation factors and the simulators/environment applied in the heuristic, learning automata-based, and exponential moving average-based offloading schemes.

4.10 Auction-Based Offloading Schemes

In [174], proposed a three-tier architecture for task offloading in MCC which tasks should be offloaded to near SMDs or cloudlets as the first two tiers and in the next step the MCC as the third tier. They applied the auction theory concepts to assign the task optimally to the SMDs, cloudlets, or MCC regarding factors such as power consumption and latency.

The approach presented in [175] introduces mCloudAuc, a reverse auction-based task offloading method for providing real-time auctions in the MMC. In this scheme, an SMD acts as a seller and competes with other SMDs by bidding on its computing resources. Also, an SMD that acts as a buyer may pay the price of bidding and then offload its tasks to the winning SMD. They indicated through simulations and proof the computation efficiency, individual rationality for the participants, and truthfulness. The authors implemented the mCloudAuc on the Android platform and provided its effectiveness.

In [176], Zhang et al. introduced a computation offloading method for the edge-computing system, consisting of energy harvesting SMDs and a dispatcher that collects the IoT devices' computation tasks. In this scheme, SMDs with limited computation capabilities to offload their loads on other SMDs for rewards. They introduced RoA, an online rewards-optimal auction, for optimizing rewards for handling the offloaded tasks while considering dynamic energy harvesting and arrivals of the computation tasks. RoA is based on Vickrey-Clarke-Groves auction and Lyapunov optimization. The authors validated their task assignment optimality by conducting several experiments and simulations.

Table 5 Properties of the Heuristic-based offloading, learning automata offloading schemes, exponential moving average, and auction-based offloading schemes

Scheme	Evaluation factors														Offloading type											
	Simulators/ environments							Properties							Code off-loading	Data off-loading	Computing off-loading									
	Energy consumption	Delay	Execution time	Cost	Profit	Completion time	Fitness value	Off-loading ratio	CPU usage	Processing time	Out-age probability	Resource usage	Open-Stack	Cloud-sim				Amaz-son	Real-environment	Mat-lab	Two-tier architecture	Three-tier architecture	Multi-site	Predictive		
[94]	✓		✓	✓									✓					✓							✓	
[97]	✓															✓										✓
[158]	✓	✓															✓									✓
[160]	✓		✓				✓										✓				✓					✓
[55]	✓			✓																						✓
[25]	✓		✓	✓																						✓
[161]	✓																									✓
[162]	✓																									✓
[87]	✓																									✓
[164]	✓																									✓
[163]	✓																									✓
[81]	✓		✓																							✓
[44]	✓			✓																						✓
[33]	✓																									✓
[160]	✓																									✓
[155]	✓		✓																							✓
[159]	✓		✓																							✓
[165]	✓		✓																							✓
[71]	✓																									✓
[35]	✓		✓																							✓
[170]	✓		✓																							✓
[171]	✓																									✓
[172]	✓																									✓
[173]	✓																									✓
[174]	✓		✓																							✓
[175]	✓																									✓

Table 5 (continued)

Evaluation factors	Simulators/ environments										Properties			Offloading type											
	Energy consumption	Delay	Execution time	Cost	Profit	Completion time	Fitness value	Offloading ratio	CPU usage	Processing time	Outputage probability	Resource usage	Open-Stack	Cloud-sim	Amazon	Real-environment	Matlab	Two-tier architecture	Three-tier architecture	Multisite	Predictive	Data offloading	Code offloading	Computing offloading	
[176]	✓				✓									✓		✓			✓					✓	✓
[56]	✓		✓													✓			✓					✓	✓

5 Discussion

This subsection presents an extensive analysis and comparison of the offloading schemes in a cloud environment, outlined in the previous section. The results of this section can be used to highlight the directions of future research and help to develop new offloading solutions for cloud DCs. This section attempts to provide the following information about the offloading frameworks designed for the cloud:

- Evaluation metrics considered in the simulation process of the proposed offloading approaches.
- Simulators used to evaluate and analyze the proposed offloading schemes in various situations.
- Techniques adapted to handle the offloading process.
- Percentage of the two-tier and three-tier offloading approaches
- Percentage of the meta-heuristic algorithms applied in the offloading approaches
- Percentage of the heuristic offloading schemes
- Number of the stochastic offloading schemes

Figure 4 depicts the number of the proposed offloading schemes using various algorithms and techniques outlined in the previous section. As shown in this figure, to model the inherent uncertainty in the realistic offloading process, many offloading schemes based on the stochastic modeling are proposed. Figure 5 tries to depict the number of offloading solutions, which applies each kind of analytical modeling methods. As shown in this figure, Queueing theory is employed in most schemes to deal with offloading. However, less works have been conducted using HMM models and semi-Markov models. Figure 6 indicates the metaheuristic algorithms applied in the studied offloading approaches and the number of schemes that involve each kind of such algorithms. As shown in this figure, a few research types are performed to optimize different offloading features in MCC.

Also, Fig. 7 indicates the number of heuristic offloading frameworks outlined in the previous section. Regarding the limited battery power of the SMD's and the importance of the MCC's energy-saving, DVFS-based power management can be further investigated to alleviate the energy overheads in the SMDs, cloudlets, and the remote cloud DCs. Also, as shown in this figure, only a few numbers of offloading approaches have considered security issues.

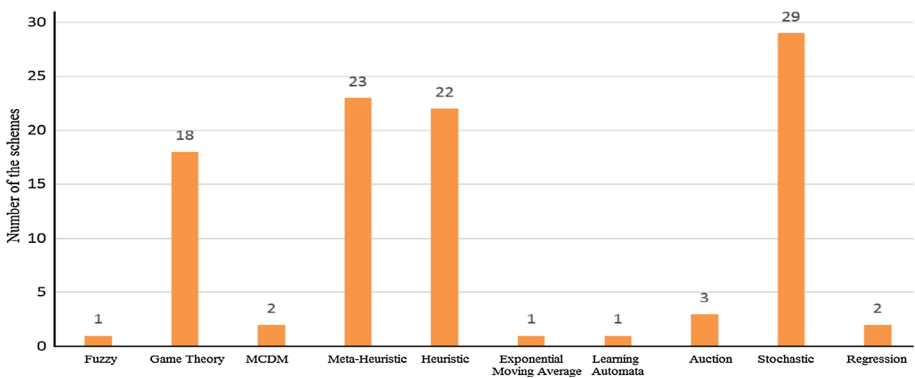


Fig. 4 Number of the offloading schemes proposed in each category

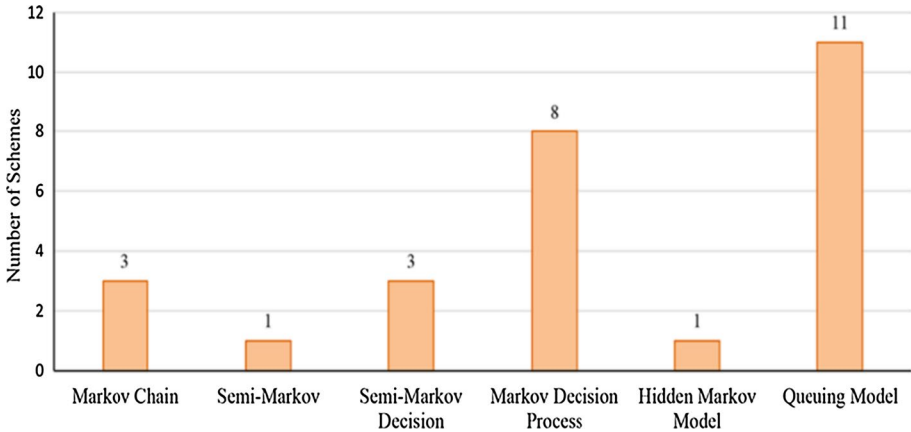


Fig. 5 Number of the stochastic offloading and scheduling schemes

Fig. 6 Percentage of the meta-heuristic offloading schemes

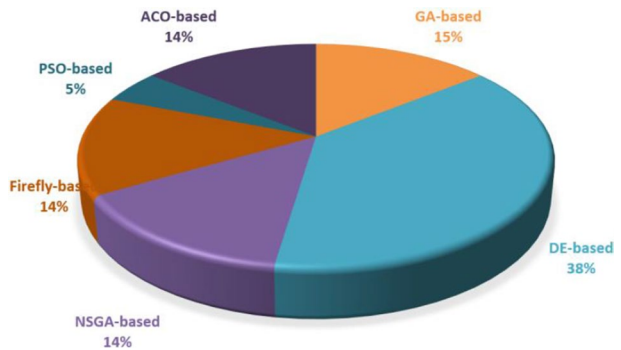
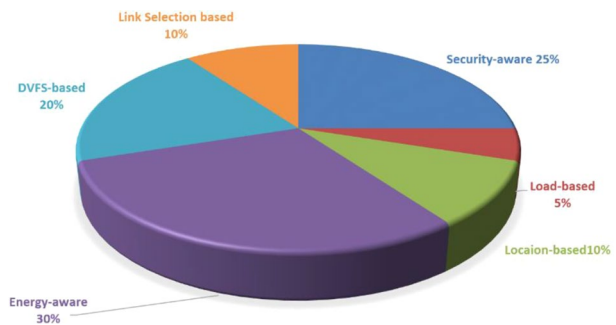


Fig. 7 Percentage of the heuristic offloading schemes



Considering the increasing number of security attacks on wireless networks, malicious behaviors, and security attacks should be further investigated in future studies.

Figure 8 exhibits the percentage of the two-tier and three-tier offloading schemes in the MCC environment. As outlined before, systems that benefit from the three-tier architecture benefit from one or more cloudlets in the middle layer, which alleviate the MCC load and reduce the offloading delays for the SMDs. As shown in this figure, fewer schemes have used

Fig. 8 Percentage of the two-tier and three-tier offloading schemes

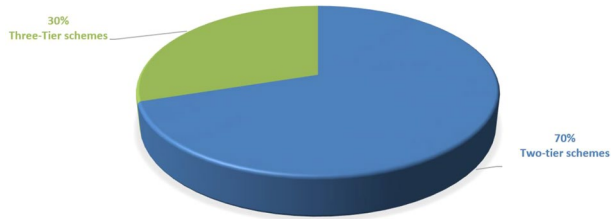
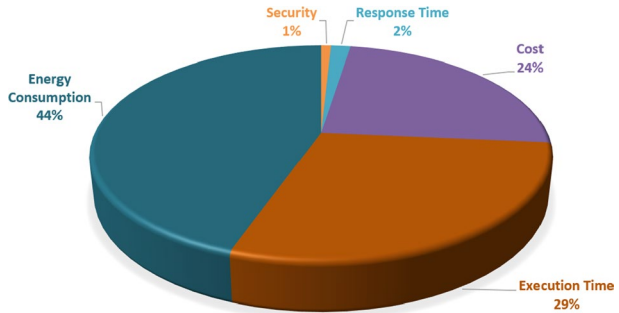


Fig. 9 Percentage of the applied evaluation metrics



cloudlets in their architecture, and this area can be further investigated in future studies. In this context, issues such as cloudlets' mobility, security, and architecture can focus on the next research. Figure 9 show the percentage of the evaluation metrics applied in the investigated offloading schemes. As shown in this figure, energy consumption, execution, and cost are three main metrics considered in the offloading articles. However, as shown in this figure, important metrics such as security-related metrics are neglected and few researches have been conducted in the security context. Also, Fig. 10 indicates the number of simulation environments and programming languages which have been used in evaluation of the proposed offloading approaches. As exhibited in the following figure, most of the schemes have used real environments in their experiments and applied Amazon EC2, OpenStack, or other cloud computing environments. Besides, the Cloudsim simulator and Matlab programming environment.

6 Conclusion

Emerging new processing-intensive applications and ever-growing wireless networks have enabled data, code, and applications offloading from the SMDs onto the MCC servers. Offloading empowers the SMDs with unlimited resources of MCCs and saves their battery power. Typically, the offloading decision should be made considering factors such as delay, energy consumption, and the access ability of WiFi and cellular wireless links;

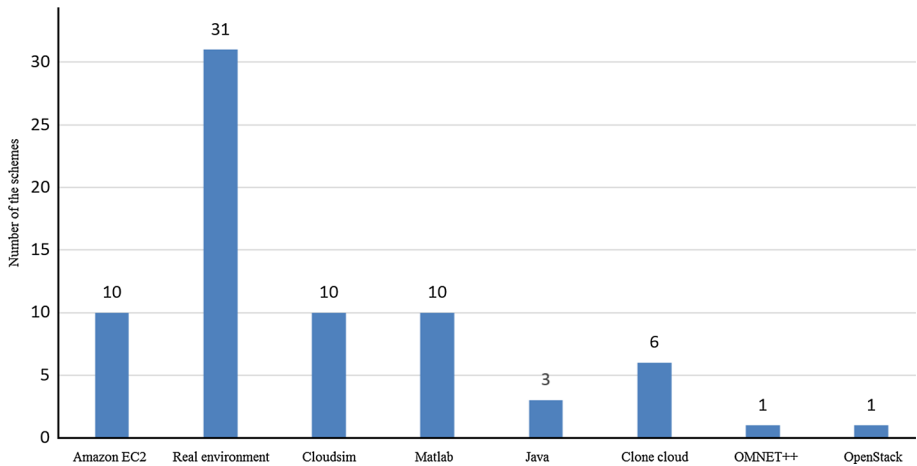


Fig. 10 Applied simulation and programming environments

For handling these issues, numerous scheduling and offloading approaches are introduced in the literature, which this paper is aimed to provide a comprehensive survey and analysis of them. We focus on the algorithms employed in making the offloading decisions and present a taxonomy of the offloading frameworks regarding their applied algorithms. Furthermore, a comparison of these approaches are presented for highlighting their datasets, evaluation factors applied in the simulation process, architecture, and offloading methods.

Also, in future studies, the following issues should be discussed in the offloading domain:

- Nevertheless, offloading data, code, and computation to the remote cloud servers makes them vulnerable to different kinds of attacks. Thus, in future studies, providing automatic offloading solutions according to the organizations' security policies and users' security levels should be further analyzed. In this context, offloading decisions can be investigated regarding the security policies of the organizations.
- Integrating offloading with the trust management schemes is another interesting issue that can be employed to blacklist selfish nodes or malicious ones that offload harmful codes and applications to the cloud or cloudlets.
- Also, for mitigating the energy consumption, dynamic energy management techniques such as DVFS should be investigated on the SMDs and MCC servers. Besides, less research is conducted in this context, and it should be studied more in future investigations.
- For improving the offloading performance, various prediction methods to predict the workload on the MCC data centers can be applied. In this context, the workload prediction schemes designed for MCC environments can be integrated with the offloading approaches to provide more robust offloading solutions.
- Variation in the offloading destinations' processing capabilities, which may be under a distributed denial of service (DDoS) attacks, has not been investigated and it should be addressed in the future. Also, regarding the DDoS attack and their impact of the bandwidth of the cloud datacenters, proper investigations should be conducted to

- Dealing with malicious SMD's which may be infected by the growing viruses and malware to conduct DDoS attacks on the offloading destinations has not been discussed in examined literature and should be focused further.

References

1. Debnath, H., Gezzi, G., Corradi, A., Gehani, N., Ding, X., Curtmola, R. *et al.* (2018). Collaborative offloading for distributed mobile-cloud apps," in *2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, 2018, pp. 87–94.
2. Masdari, M., & Jalali, M. (2016). A survey and taxonomy of DoS attacks in cloud computing. *Security and Communication Networks*, 9, 3724–3751.
3. Masdari, M., Nabavi, S. S., & Ahmadi, V. (2016). An overview of virtual machine placement schemes in cloud computing. *Journal of Network and Computer Applications*, 66, 106–127.
4. Masdari, M., ValiKardan, S., Shahi, Z., & Azar, S. I. (2016). Towards workflow scheduling in cloud computing: a comprehensive analysis. *Journal of Network and Computer Applications*, 66, 64–82.
5. Masdari, M., Salehi, F., Jalali, M., & Bidaki, M. (2017). A survey of PSO-based scheduling algorithms in cloud computing. *Journal of Network and Systems Management*, 25, 122–158.
6. Junior, W., Oliveira, E., Santos, A., & Dias, K. (2019). A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment. *Future Generation Computer Systems*, 90, 503–520.
7. Elgendy, I., Zhang, W., Liu, C., & Hsu, C.-H. (2018). An efficient and secured framework for mobile cloud computing. *IEEE Transactions on Cloud Computing*.
8. Liu, L., Guo, X., Chang, Z., & Ristaniemi, T. (2018). Joint optimization of energy and delay for computation offloading in cloudlet-assisted mobile cloud computing. *Wireless Networks*, pp. 1–14.
9. Beraldi, R., Massri, K., Abderrahmen, M., & Alnuweiri, H. (2013). Towards automating mobile cloud computing offloading decisions: An experimental approach. in *ICSNC 2013: The Eighth International Conference on Systems and Networks Communications*.
10. Luzuriaga, J., Cano, J. C., Calafate, C., & Manzoni, P. (2013). Evaluating computation offloading trade-offs in mobile cloud computing: A sample application. In *CLOUD COMPUTING 2013, The Fourth International Conference on Cloud Computing, GRIDS, and Virtualization*, 2013, pp. 138–143.
11. Byun, H., Park, B.-K., & Jeong, Y.-S. (2016). Mobile Agent Oriented Service for Offloading on Mobile Cloud Computing. In *Advances in Computer Science and Ubiquitous Computing*. Heidelberg, Springer, pp. 920–925.
12. Yang, S., Kwon, Y., Cho, Y., Yi, H., Kwon, D., Youn, J., *et al.* (2013). Fast dynamic execution offloading for efficient mobile cloud computing. In *Pervasive Computing and Communications (PerCom)*. *IEEE International Conference on*, pp 20–28.
13. Shiraz, M., & Gani, A. (2014). A lightweight active service migration framework for computational offloading in mobile cloud computing. *The Journal of Supercomputing*, 68, 978–995.
14. Wu, H. (2018). Multi-objective decision-making for mobile cloud offloading: A survey. *IEEE Access*, 6, 3962–3976.
15. Noor, T. H., Zeadally, S., Alfazi, A., & Sheng, Q. Z. (2018). Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications*, 115, 70–85.
16. Jiang, Y., He, J., Li, Q., & Xiao, X. (2014). A dynamic execution offloading model for efficient mobile cloud computing. In *Global Communications Conference (GLOBECOM)*. *IEEE*, pp 2302–2307.
17. Zhou, B., Dastjerdi, A. V., Calheiros, R. N., & Buyya, R. (2018). An online algorithm for task offloading in heterogeneous mobile clouds. In *ACM Transactions on Internet Technology (TOIT)*, vol. 18, p. 23, 2018.
18. Gu, F., Niu, J., Qi, Z., & Atiquazzaman, M. (2018). Partitioning and offloading in smart mobile devices for mobile cloud computing: State of the art and future directions. *Journal of Network and Computer Applications*.
19. Enzai, M., Idawati, N., & Tang, M. (2016). A heuristic algorithm for multi-site computation offloading in mobile cloud computing. *Procedia Computer Science*, 80, 1232–1241.

20. Rodriguez, C. P., & Souza, G. (2011). Decision-Making Model for Offshore Offloading Operations Based on Probabilistic Risk Assessment. In *Vulnerability, Uncertainty, and Risk: Analysis, Modeling, and Management*, pp. 385–393.
21. Fiandrino, C., Kliazovich, D., Bouvry, P., & Zomaya, A. Y. (2015). “Network-assisted offloading for mobile cloud applications,” in *Communications (ICC). IEEE International Conference on, 2015*, 5833–5838.
22. Shuja J., Gani, A., Naveed, A., Ahmed, E., & Hsu, C.-H. (2017). Case of ARM emulation optimization for offloading mechanisms in Mobile Cloud Computing. *Future Generation Computer Systems*, vol. 76, pp. 407–417
23. Jia, M., Cao, J., & Yang, L. (2014). “Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing,” in *Computer Communications Workshops (INFOCOM WKSHPs). IEEE Conference on, 2014*, 352–357.
24. Kumar, K., & Lu, Y.-H. (2010). Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43, 51–56.
25. Shiraz, M., Sookhak, M., Gani, A., & Shah, S. A. A. (2015). A study on the critical analysis of computational offloading frameworks for mobile cloud computing. *Journal of Network and Computer Applications*, 47, 47–60.
26. Chen, M.-H., Dong, M., & Liang, B. (2016). “Joint offloading decision and resource allocation for mobile cloud with computing access point,” in *Acoustics, Speech and Signal Processing (ICASSP). IEEE International Conference on, 2016*, 3516–3520.
27. Wang, X., Wang, J., Wang, X., & Chen, X. (2017). Energy and delay tradeoff for application offloading in mobile cloud computing. *IEEE Systems Journal*, 11, 858–867.
28. Chen, M.-H., Liang, B., & Dong, M. (2018). Multi-user multi-task offloading and resource allocation in mobile cloud systems. *IEEE Transactions on Wireless Communications*, 17, 6790–6805.
29. Goudarzi, M., Zamani, M., & Toroghi Haghghat, A. (2017). A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing. *International Journal of Communication Systems*, 30, e3241.
30. Mahmoodi, S. E., Subbalakshmi, K., & Uma, R. (2019). Classification of Mobile Cloud Offloading. In *Spectrum-Aware Mobile Computing*. Heidelberg: Springer, 2019, pp. 7–11.
31. Boukerche, A., Guan, S., & Grande, R. E. D. (2019). Sustainable Offloading in Mobile Cloud Computing: Algorithmic Design and Implementation. *ACM Computing Surveys (CSUR)*, vol. 52, p. 11, 2019.
32. La, H. J., & Kim, S. D. (2014). A taxonomy of offloading in mobile cloud computing. In *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on, 2014*, pp. 147–153.
33. Lee, H.-S., & Lee, J.-W. (2018). Task offloading in heterogeneous mobile cloud computing: Modeling, analysis, and cloudlet deployment. *IEEE Access*, 6, 14908–14925.
34. Kumar, D., & Sharma, R. (2017). Data synchronization and offloading techniques for energy optimization in mobile cloud computing. In *Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS). International Conference on, 2017*, pp 633–638.
35. Mahmoodi, S. E., Subbalakshmi, K., & Uma, R. (2019). Cognitive Cloud Offloading Using Multiple Radios. In *Spectrum-Aware Mobile Computing*. Heidelberg: Springer, 2019, pp. 23–33.
36. Akherfi, K., Gerndt, M., & Harroud, H. (2018). Mobile cloud computing for computation offloading: Issues and challenges. *Applied computing and informatics*, 14, 1–16.
37. Kumari, R., & Kaushal, S. (2017). Application Offloading Using Data Aggregation in Mobile Cloud Computing Environment. In *Leadership, Innovation and Entrepreneurship as Driving Forces of the Global Economy*. Heidelberg: Springer, 2017, pp. 17–29.
38. Su, W.-T., & Ng, K. S. (2013). Mobile cloud with smart offloading system. In *Communications in China (ICCC). IEEE/CIC International Conference on, 2013*, pp 680–685.
39. Flores, H., & Srirama, S. N. Adaptive code offloading and resource-intensive task delegation for mobile cloud applications. In *Proceeding 4th ACM workshop on Mobile Cloud Computing and services (MCS '13)*, pp. 9–16.
40. Huang, T., Chen, Y., Xue, Ji, H., Xue, Y., Qi, L., et al. (2018). Energy-Efficient Computation Offloading for Multimedia Workflows in Mobile Cloud Computing. In *International Conference on Testbeds and Research Infrastructures*, 2018, pp. 113–123.
41. Liwang, M., Wang, J., Gao, Z., Du, X., & Guizani, M. (2019). Game theory based opportunistic computation offloading in cloud-enabled IoV. *IEEE Access*, 7, 32551–32561.
42. Magurawalage, C. M. S., Yang, K., Hu, L., & Zhang, J. (2014). Energy-efficient and network-aware offloading algorithm for mobile cloud computing. *Computer Networks*, 74, 22–33.

43. Lin, T.-Y., Lin, T.-A., Hsu, C.-H., & King, C.-T. (2013). Context-aware decision engine for mobile cloud offloading. In *Wireless Communications and Networking Conference Workshops (WCNCW), IEEE, 2013*, pp 111–116.
44. Shiraz, M., Gani, A., Shamim, A., Khan, S., & Ahmad, R. W. (2015). Energy efficient computational offloading framework for mobile cloud computing. *Journal of Grid Computing*, 13, 1–18.
45. Choi, K., Lee, J., Kim, Y., Kang, S., & Han, H. (2015). Feasibility of the computation task offloading to GPGPU-enabled devices in mobile cloud. In *Cloud and Autonomic Computing (ICCAC), International Conference on, 2015*, pp 244–251.
46. Tseng, F.-H., Cho, H.-H., Chang, K.-D., Li, J.-C., & Shih, T. K. (2018). Application-oriented offloading in heterogeneous networks for mobile cloud computing. *Enterprise Information Systems*, 12, 398–413.
47. Lee, K., & Shin, I. (2015). User mobility model based computation offloading decision for mobile cloud. *Journal of Computing Science and Engineering*, 9, 155–162.
48. Kwon, Y., Yi, H., Kwon, D., Yang, S., Cho, Y., & Paek, Y. (2016). Precise execution offloading for applications with dynamic behavior in mobile cloud computing. *Pervasive and Mobile Computing*, vol. 27, pp. 58–74, 2016/04/01/ 2016.
49. Junior, W., França, A., Dias, K., & de Souza, J. N. (2017). Supporting mobility-aware computational offloading in mobile cloud environment. *Journal of Network and Computer Applications*, 94, 93–108.
50. Khanna, A., Kero, A., & Kumar, D. (2016). Mobile cloud computing architecture for computation offloading. In *Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on, 2016*, pp. 639–643.
51. Tao, Y. Zhang, Y., & Ji, Y. (2015). Efficient Computation Offloading Strategies for Mobile Cloud Computing. In *Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on, 2015*, pp. 626–633.
52. Pu, L., Xu, J., Jin, X., & Zhang, J. (2013). “SmartVirtCloud: virtual cloud assisted application offloading execution at mobile devices”’ discretion,” in *Wireless Communications and Networking Conference (WCNC), IEEE, 2013*, 4398–4403.
53. Barbera, M. V., Kosta, S., Mei, A., Perta, V. C., & Stefa, J. (2014). “Mobile offloading in the wild: Findings and lessons learned through a real-life experiment with a new cloud-aware system,” in *INFOCOM. Proceedings IEEE, 2014*, pp 2355–2363.
54. Goyal, M., & Saini, P. (2016). “A fault-tolerant energy-efficient computational offloading approach with minimal energy and response time in mobile cloud computing,” in *Parallel, Distributed and Grid Computing (PDGC), Fourth International Conference on, 2016*, pp 44–49.
55. Chen, S., Wang, Y., & Pedram, M. (2013) A semi-Markovian decision process based control method for offloading tasks from mobile devices to the cloud. In *GLOBECOM, 2013*, pp. 2885–2890.
56. Xia, F., Ding, F., Li, J., Kong, X., Yang, L. T., & Ma, J. (2014). Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing. *Information Systems Frontiers*, 16, 95–111.
57. Mahmoodi, S. E., Uma, R., & Subbalakshmi, K. (2016). Optimal joint scheduling and cloud offloading for mobile applications. *IEEE Transactions on Cloud Computing*, 2016.
58. Shi, Y., Chen, S., & Xu, X. (2018). MAGA: a mobility-aware computation offloading decision for distributed mobile cloud computing. *IEEE Internet of Things Journal*, 5, 164–174.
59. Meng, T., Wolter, K., Wu, H., & Wang, Q. (2018). A secure and cost-efficient offloading policy for Mobile Cloud Computing against timing attacks. *Pervasive and Mobile Computing*, vol. 45, pp. 4–18, 2018/04/01/ 2018.
60. Zhang, W., & Wen, Y. (2018). Energy-efficient task execution for application as a general topology in mobile cloud computing. *IEEE Transactions on cloud Computing*, 6, 708–719.
61. Sundararaj, V. (2019). Optimal task assignment in mobile cloud computing by queue based Ant-Bee algorithm. *Wireless Personal Communications*, 104, 173–197.
62. Liu, D., Khoukhi, L., & Hafid, A. (2017). “Decentralized data offloading for mobile cloud computing based on game theory,” in. *Second International Conference on Fog and Mobile Edge Computing (FMEC), 2017*, 20–24.
63. Chen, M.-H., Liang, B., & Dong, M. (2017). Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017*, pp. 1–9.
64. Ma, X., Lin, C., Xiang, X., & Chen, V. (2015). Game-theoretic analysis of computation offloading for cloudlet-based mobile cloud computing. In *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2015, pp. 271–278.

65. Wang, T., Wei, X., Tang, C., & Fan, J. (2018). Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints. *Peer-to-Peer Networking and Applications*, 11, 793–807.
66. Chen, M.-H., Dong, M., & Liang, B. (2018). Resource Sharing of a Computing Access Point for Multi-user Mobile Cloud Offloading with Delay Constraints. *IEEE Transactions on Mobile Computing*, 2018.
67. Cheng, J., Shi, Y., Bai, B., & Chen, W. (2016). Computation offloading in cloud-RAN based mobile cloud computing system. In *Communications (ICC). IEEE International Conference on*, 2016, pp 1–6.
68. Ellouze, A., Gagnaire, M., & Haddad, A. (2015). A mobile application offloading algorithm for mobile cloud computing. In *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on*, 2015, pp. 34–40.
69. Li, J., Bu, K., Liu, X., & Xiao, B. (2013). Enda: Embracing network inconsistency for dynamic application offloading in mobile cloud computing. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, 2013, pp. 39–44.
70. Wu, H., Wang, Q., & Wolter, K. (2013). Tradeoff between performance improvement and energy saving in mobile cloud offloading systems. In *Communications Workshops (ICC). IEEE International Conference on*, 2013, pp 728–732.
71. Song, J., Cui, Y., Li, M., Qiu, J., & Buyya, R. (2014). Energy-traffic tradeoff cooperative offloading for mobile cloud computing. In *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of*, 2014, pp. 284–289.
72. Zhou, Z., Zhang, H., Ye, L., & Du, X. (2016). Cuckoo: flexible compute-intensive task offloading in mobile cloud computing. *Wireless Communications and Mobile Computing*, 16, 3256–3268.
73. Deng, S., Huang, L., Taheri, J., & Zomaya, A. Y. (2015). Computation offloading for service workflow in mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 26, 3317–3329.
74. Dhanya, N., & Kousalya, G. (2015). Adaptive and Secure Application Partitioning for Offloading in Mobile Cloud Computing. In *International Symposium on Security in Computing and Communication*, 2015, pp. 45–53.
75. Goudarzi, M., Movahedi, Z., & Nazari, M. (2016). Efficient multisite computation offloading for mobile cloud computing. In *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld). Intl IEEE Conferences*, 2016, pp 1131–1138.
76. Chen, X., Jiao, L., Li, W., & Fu, X. (2016). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, pp. 2795–2808.
77. Tianze, L., Muqing, W., & Min, Z. (2016). A stackelberg game based task offloading mechanism for ad-hoc based mobile cloud computing. In *Wireless Communication Systems (ISWCS), 2016 International Symposium on*, 2016, pp. 618–622.
78. Kchaou, H., Kechaou, Z., & Alimi, A. M. (2015). Towards an offloading framework based on Big Data analytics in Mobile Cloud Computing Environments. *Procedia Computer Science*, 53, 292–297.
79. Haghighi, V., & Moayedian, N. S. (2018). An offloading strategy in mobile cloud computing considering energy and delay constraints. *IEEE Access*, 6, 11849–11861.
80. Kuang, Z., Guo, S., Liu, J., & Yang, Y. (2018). A quick-response framework for multi-user computation offloading in mobile cloud computing. *Future Generation Computer Systems*, 81, 166–176.
81. Pan, S., Liu, C., Zeng, D., Yao, H., & Qian, Z. (2018). “Fine-Grained Task-Dependency Offloading in Mobile Cloud Computing. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications* (pp. 977–982). Cloud & Big Data Computing: Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI).
82. Chen, M.-H., Dong, M., & Liang, B. (2018). Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints. *IEEE Transactions on Mobile Computing*, 17, 2868–2881.
83. Guo, S., Liu, J., Yang, Y., Xiao, B., & Li, Z. (2019). Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing. *IEEE Transactions on Mobile Computing*, 18, 319–333.
84. Zhang, Y., He, J., & Guo, S. (2018). Energy-Efficient Dynamic Task Offloading for Energy Harvesting Mobile Cloud Computing. In *2018 IEEE International Conference on Networking, Architecture and Storage (NAS)*, 2018, pp. 1–4.

85. Hasan, R., Hossain, M., & Khan, R. (2018). Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading. *Future Generation Computer Systems*, *86*, 821–835.
86. Flores, H., Hui, P., Nurmi, P., Lagerspetz, E., Tarkoma, S., Manner, J., et al. (2018). Evidence-aware mobile computational offloading. *IEEE Transactions on Mobile Computing*, *17*, 1834–1850.
87. Wang, K., Yang, K., & Magurawalage, C. S. (2018). Joint energy minimization and resource allocation in C-RAN with mobile cloud. *IEEE Transactions on Cloud Computing*, *6*, 760–770.
88. Somula, R., Anilkumar, C., Venkatesh, B., Karrothu, A., Kumar, C.P., & Sasikala, R. (2019). Cloudlet Services for Healthcare Applications in Mobile Cloud Computing. In *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, 2019, pp. 535–543.
89. Nakahara, F. A., & Beder, D. M. (2018). A context-aware and self-adaptive offloading decision support model for mobile cloud computing system. *Journal of Ambient Intelligence and Humanized Computing*, *9*, 1561–1572.
90. Mahmoodi, S. E., Subbalakshmi, K., & Uma, R. (2019). Joint Scheduling and Cloud Offloading Using Single Radio. In *Spectrum-Aware Mobile Computing*. Heidelberg: Springer, 2019, pp. 13–21.
91. Zhao, J., Li, Q., Gong, Y., & Zhang, K. (2019). Computation Offloading and Resource Allocation for Cloud Assisted Mobile Edge Computing in Vehicular Networks. *IEEE Transactions on Vehicular Technology*, 2019.
92. Guo, K., Yang, M., & Zhang, Y. (2018). Computation offloading over a shared communication channel for mobile cloud computing. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp 1–6.
93. Zhou, C., & Tham, C.-K. (2018). Deadline-aware peer-to-peer task offloading in stochastic mobile cloud computing systems. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–9.
94. Jeevan, A. G., & Mohamed, M. M. (2018). DyTO: Dynamic task offloading strategy for mobile cloud computing using surrogate object model. *International Journal of Parallel Programming*, pp. 1–17, 2018.
95. Vankadara, S., & Dasari, N. (2019). Energy-aware dynamic task offloading and collective task execution in mobile cloud computing. *International Journal of Communication Systems*, p. e3914, 2019.
96. Qin, A., Cai, C., Wang, Q., Ni, Y., & Zhu, H. (2019). Game Theoretical Multi-user Computation Offloading for Mobile-Edge Cloud Computing. In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019, 328–332.
97. Han, D., Chen, W., Bai, B., & Fang, Y. (2019). Offloading Optimization and Bottleneck Analysis for Mobile Cloud Computing. *IEEE Transactions on Communications*, 2019.
98. Liu, D., Khoukhi, L., & Hafid, A. (2018). Prediction-based mobile data offloading in mobile cloud computing. *IEEE Transactions on Wireless Communications*, *17*, 4660–4673.
99. Wu, H., & Huang, D. (2014). Modeling multi-factor multi-site risk-based offloading for mobile cloud computing. In *Network and Service Management (CNSM)*, 2014 10th International Conference on, 2014, pp. 230–235.
100. Cao, H., & Cai, J. (2018). Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach. *IEEE Transactions on Vehicular Technology*, *67*, 752–764.
101. Zheng, J., Cai, Y., Wu, Y., & Shen, X. (2019). Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach. *IEEE Transactions on Mobile Computing*, *18*, 771–786.
102. Guo, X., Liu, L., Chang, Z., & Ristaniemi, T. (2018). Data offloading and task allocation for cloudlet-assisted ad hoc mobile clouds. *Wireless Networks*, *24*, 79–88.
103. Ahn, S., Lee, J., Park, S., Newaz, S. S., & Choi, J. K. (2018). Competitive partial computation offloading for maximizing energy efficiency in mobile cloud computing. *IEEE Access*, *6*, 899–912.
104. Jošilo, S., & Dán, G. (2019). Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks. *IEEE Transactions on Mobile Computing*, *18*, 207–220.
105. Chen, M.-H., Dong, M., & Liang, B. (2016). Multi-user mobile cloud offloading game with computing access point. In *Cloud Networking (Cloudnet)*, 2016 5th IEEE International Conference on, 2016, pp. 64–69.
106. Cardellini, V., Personé, V. D. N., Di Valerio, V., Facchinei, F., Grassi, V., Presti, F. L., et al. (2016). A game-theoretic approach to computation offloading in mobile cloud computing. *Mathematical Programming*, *157*, 421–449.
107. Kim, S. (2015). Nested game-based computation offloading scheme for mobile cloud IoT systems. *EURASIP Journal on Wireless Communications and Networking*, 2015, 229.
108. Chen, X. (2015). Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, *26*, 974–983.

109. Zhang, Y., Shi, Y., Shen, F., Yan, F., & Shen, L. (2019). Price-based joint offloading and resource allocation for ad hoc mobile cloud. *IEEE Access*, 7, 62769–62784.
110. Li, Y., Liu, J., Li, Q., & Xiao, L. (2015). Mobile cloud offloading for malware detections with learning. In *Computer Communications Workshops (INFOCOM WKSHPs). IEEE Conference on, 2015.*, pp 197–201.
111. Zheng, J., Cai, Y., Wu, Y., & Shen, X. (2018). Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach. *IEEE Transactions on Mobile Computing*, 18, 771–786.
112. Kuang, Z., Shi, Y., Guo, S., Dan, J., & Xiao, B. (2019). Multi-user offloading game strategy in OFDMA mobile cloud computing system. *IEEE Transactions on Vehicular Technology*, 68, 12190–12201.
113. Mahini, H., Rahmani, A. M., & Mousavirad, S. M.. (2020). An evolutionary game approach to IoT task offloading in fog-cloud computing. *The Journal of Supercomputing*, pp. 1–28, 2020.
114. Shen, B., Xu, X., Dar, F., Qi, L., Zhang, X., & Dou, W. (2020). Dynamic Task Offloading with Minority Game for Internet of Vehicles in Cloud-Edge Computing. In *IEEE International Conference on Web Services (ICWS), 2020*, pp 372–379.
115. Guo, S., Wang, Y., & Wang, H. (2019). Multi-factor based Dynamic Offloading with Coalitional Game in Mobile Cloud Computing. In *IEEE/CIC International Conference on Communications in China (ICCC), 2019*, pp 863–868.
116. Wu, H., Wang, Q., & Wolter, K. (2012). Methods of cloud-path selection for offloading in mobile cloud computing systems. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, 2012*, pp. 443–448.
117. Singla, C., & Kaushal, S., (2015). Cloud path selection using fuzzy analytic hierarchy process for offloading in mobile cloud computing. In *Recent Advances in Engineering & Computational Sciences (RAECS), 2015 2nd International Conference on, 2015*, pp. 1–5.
118. Shukla, A. K., Pippal, S. K., & Chauhan, S. S. (2019). An empirical evaluation of teaching–learning-based optimization, genetic algorithm and particle swarm optimization," *International Journal of Computers and Applications*, pp. 1–15, 2019.
119. Sheikh, I., & Das, O. (2018). Modeling the effect of parallel execution on multi-site computation offloading in mobile cloud computing. In *European Workshop on Performance Engineering, 2018*, pp. 219–234.
120. Guo, S., Wang, Y., Meng, S., & Ma, N. (2018). Delay Optimization for Mobile Cloud Computing Application Offloading in Smart Cities. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2018*, pp. 456–466.
121. Abd, S. K., Al-Haddad, S. A. R., Hashim, F., Abdullah, A. B., & Yussof, S. (2017). Energy-aware fault tolerant task offloading of mobile cloud computing. In *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2017*, pp. 161–164.
122. Kaushi, N., & Kumar, J. (2014). A computation offloading framework to optimize energy utilisation in mobile cloud computing environment. *International Journal of Computer Applications and Information Technology*, 5, 61–69.
123. Roostaei, R., & Movahedi, Z. (2016). Mobility and Context-Aware Offloading in Mobile Cloud Computing. In *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld). Intl IEEE Conferences, 2016.*, pp 1144–1148.
124. Goudarzi, M., Movahedi, Z., & Nazari, M. (2016). Mobile cloud computing: a multisite computation offloading. In *Telecommunications (IST), 2016 8th International Symposium on, 2016*, pp. 660–665.
125. Tout, H., Talhi, C., Kara, N., & Mourad, A. (2016). Selective mobile cloud offloading to augment multi-persona performance and viability. *IEEE Transactions on Cloud Computing*, 2016.
126. Manukumar, S. T., & Muthuswamy, V. (2019). A Novel Multi-Objective Efficient Offloading Decision Framework in Cloud Computing for Mobile Computing Applications. *Wireless Personal Communications*, pp. 1–18, 2019.
127. Wang, Y., Wu, L., Yuan, X., Liu, X., & Li, X. (2019). An Energy-Efficient and Deadline-Aware Task Offloading Strategy Based on Channel Constraint for Mobile Cloud Workflows (May 2019). *IEEE Access*, 2019.
128. Alli, A. A., & Alam, M. M. (2019). SecOFF-FCIoT: Machine learning based secure offloading in Fog-Cloud of things for smart city applications. *Internet of Things*, 7, 100070.
129. Huang, T., Ruan, F., Xue, S., Qi, L., & Duan, Y. (2019). Computation offloading for multimedia workflows with deadline constraints in cloudlet-based mobile cloud. *Wireless Networks*, pp. 1–15, 2019.

130. Adhikari, M., & Gianey, H. (2019). Energy efficient offloading strategy in fog-cloud environment for IoT applications. *Internet of Things*, 6, 100053.
131. Xu, X., Liu, Q., Luo, Y., Peng, K., Zhang, X., Meng, S., et al. (2019). A computation offloading method over big data for IoT-enabled cloud-edge computing. *Future Generation Computer Systems*, 95, 522–533.
132. Xu, X., Gu, R., Dai, F., Qi, L., & Wan, S. (2019). Multi-objective computation offloading for internet of vehicles in cloud-edge computing. *Wireless Networks*, pp. 1–19, 2019.
133. Xu, X., Fu, S., Yuan, Y., Luo, Y., Qi, L., Lin, W., et al. (2019). Multiobjective computation offloading for workflow management in cloudlet-based mobile cloud using NSGA-II. *Computational Intelligence*, 35, 476–495.
134. Chen, S., Wang, Y., & Pedram, M. (2014). Optimal offloading control for a mobile device based on a realistic battery model and semi-Markov decision process. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, 2014, pp. 369–375.
135. Goudarzi, M., Zamani, M., & Haghghat, A. T. (2017). A fast hybrid multi-site computation offloading for mobile cloud computing. *Journal of Network and Computer Applications*, 80, 219–231.
136. Dhanya, N., Kousalya, G., & Balakrishnan, P. (2017). Dynamic mobile cloud offloading prediction based on statistical regression. *Journal of Intelligent & Fuzzy Systems*, 32, 3081–3089.
137. Khoda, M. E., Razzaque, M. A., Almogren, A., Hassan, M. M., Alamri, A., & Alelaiwi, A. (2016). Efficient computation offloading decision in mobile cloud computing over 5G network. *Mobile Networks and Applications*, 21, 777–792.
138. Xiao, L., Xie, C., Chen, T., Dai, H., & Poor, H. V. (2016). A mobile offloading game against smart attacks. *IEEE Access*, 4, 2281–2291.
139. Gao, W., Li, Y., Lu, H., Wang, T., & Liu, C. (2014). On exploiting dynamic execution patterns for workload offloading in mobile cloud applications. In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, 2014, pp. 1–12.
140. Wang, Z., Zhong, Z., & Ni, M. (2017). A semi-Markov decision process-based computation offloading strategy in vehicular networks. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*, 2017, pp. 1–6.
141. Tong, L., & Gao, W. (2016). Application-aware traffic scheduling for workload offloading in mobile clouds. in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, IEEE, 2016, pp. 1–9.
142. Van Le, D., & Tham, C.-K. (2018). Quality of service aware computation offloading in an ad-hoc mobile cloud. *IEEE Transactions on Vehicular Technology*, 67, 8890–8904.
143. Chen, X., Wu, C., Zhou, Y., & Zhang, H. (2015). A learning approach for traffic offloading in stochastic heterogeneous cellular networks. In *Communications (ICC). IEEE International Conference on*, 2015, pp 3347–3351.
144. Liu, D., Khoukhi, L., & Hafid, A. (2017). Data offloading in mobile cloud computing: A markov decision process approach. In *Communications (ICC). IEEE International Conference on*, 2017, pp 1–6.
145. Terefe, M. B., Lee, H., Heo, N., Fox, G. C., & Oh, S. (2016). Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing. *Pervasive and Mobile Computing*, 27, 75–89.
146. Zannat, H., & Hossain, M. S. (2016). A hybrid framework using Markov decision process for mobile code offloading. In *Computer and Information Technology (ICCIT), 2016 19th International Conference on*, 2016, pp. 31–35.
147. Truong-Huu, T., Tham, C.-K., & Niyato, D. (2014). To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud. In *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, 2014, pp. 182–189.
148. Gao, Z., Hao, W., Zhang, R., & Yang, S. (2020). Markov decision process-based computation offloading algorithm and resource allocation in time constraint for mobile cloud computing. *IET Communications*, 14, 2068–2078.
149. Zalat, M. S., Darwish, S. M., & Madbouly, M. M. (2020). An Effective Offloading Model Based on Genetic Markov Process for Cloud Mobile Applications. In *International Conference on Advanced Intelligent Systems and Informatics*, 2020, pp. 38–50.
150. Wang, X., Xu, W., & Jin, Z. (2017). A hidden Markov model based dynamic scheduling approach for mobile cloud telemonitoring. In *Biomedical & Health Informatics (BHI). IEEE EMBS International Conference on*, 2017, pp 273–276.
151. Wu, H., Sun, Y., & Wolter, K. (2015). Analysis of the energy-response time tradeoff for delayed mobile cloud offloading. *ACM SIGMETRICS Performance Evaluation Review*, 43, 33–35.

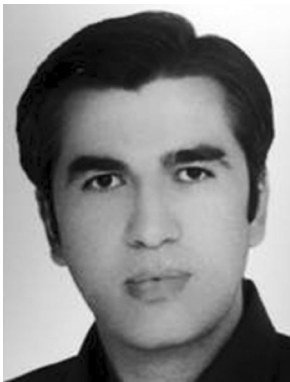
152. Wu, H., Knottenbelt, W., & Wolter, K. (2015). Analysis of the energy-response time tradeoff for mobile cloud offloading using combined metrics. In *Teletraffic Congress (ITC 27), 2015 27th International*, 2015, pp. 134–142.
153. Hyytiä, E., Spyropoulos, T., & Ott, J. (2013). Optimizing offloading strategies in mobile cloud computing. *Cryptanalyst*, 2013.
154. Wu, H., Sun, Y., & Wolter, K. (2018). Energy-efficient decision making for mobile cloud offloading. *IEEE Transactions on Cloud Computing*, 2018.
155. Zhang, J., Zhou, Z., Li, S., Gan, L., Zhang, X., Qi, L., et al. (2018). Hybrid computation offloading for smart home automation in mobile cloud computing. *Personal and Ubiquitous Computing*, 22, 121–134.
156. Wu, H. (2018). Performance Modeling of Delayed Offloading in Mobile Wireless Environments with Failures," *IEEE Communications Letters*, 2018.
157. Wang, J., Peng, J., Wei, Y., Liu, D., & Fu, J. (2017). Adaptive application offloading decision and transmission scheduling for mobile cloud computing. *China Communications*, 14, 169–181.
158. Salehan, A., Deldari, H., & Abrishami, S. (2019). An online context-aware mechanism for computation offloading in ubiquitous and mobile cloud environments. *The Journal of Supercomputing*, pp. 1–41, 2019.
159. Mazouzi, H., Boussetta, K., & Achir, N. (2019). Maximizing mobiles energy saving through tasks optimal offloading placement in two-tier cloud: A theoretical and an experimental study. *Computer Communications*, 144, 132–148.
160. Kumari, R., Kaushal, S., & Chilamkurti, N. (2018). Energy conscious multi-site computation offloading for mobile cloud computing. *Soft Computing*, 22, 6751–6764.
161. Balakrishnan, P., & Tham, C.-K. (2013). Energy-efficient mapping and scheduling of task interaction graphs for code offloading in mobile cloud computing. In *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, 2013, pp. 34–41.
162. Guo, S., Xiao, B., Yang, Y., & Yang, Y. (2016). Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, IEEE, 2016, pp. 1–9.
163. Singh, A., & Madan, N. (2016). A Region Based Offloading Mechanism in Mobile Cloud Computing Environment, 2016.
164. Farra, N., Raffa, G., Nachman, L., & Hajj, H. (2011). Energy-efficient mobile gesture recognition with computation offloading. In *Energy Aware Computing (ICEAC). International Conference on, 2011*, pp 1–6.
165. Kero, A., Khanna, A., Kumar, D., & Agarwal, A. (2019). An adaptive approach towards computation offloading for mobile cloud computing. *International Journal of Information Technology and Web Engineering (IJITWE)*, 14, 52–73.
166. Dwivedi, S., Vardhan, M., & Tripathi, S. (2020). An Effect of Chaos Grasshopper Optimization Algorithm for Protection of Network Infrastructure. *Computer Networks*, p. 107251, 2020.
167. Dwivedi, S., Vardhan, M., & Tripathi, S. (2020). Defense against distributed DoS attack detection by using intelligent evolutionary algorithm. *International Journal of Computers and Applications*, pp. 1–11, 2020.
168. Dwivedi, S., Vardhan, M., & Tripathi, S. (2020). Incorporating evolutionary computation for securing wireless network against cyberthreats. *The Journal of Supercomputing*, pp. 1–38, 2020.
169. Shukla, A. K., & Singh, P. (2019). Building an effective approach toward intrusion detection using ensemble feature selection. *International Journal of Information Security and Privacy (IJISP)*, 13, 31–47.
170. Eom, H., Juste, P. S., Figueiredo, R., Tickoo, O., Illikkal, R., & Iyer, R. (2013). Opencl-based remote offloading framework for trusted mobile cloud computing. In *Parallel and Distributed Systems (ICPADS). International Conference on, 2013*, pp 240–248.
171. Fadaraliki, D. I., & Rajendran, S. (2015). Secure mobile cloud computation offloading node privacy and security.
172. Khayyat, M., Elgendy, I. A., Muthanna, A., Alshahrani, A. S., Alharbi, S., & Koucheryavy, A. (2020). Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks. *IEEE Access*, 8, 137052–137062.
173. Krishna, P. V., Misra, S., Nagaraju, D., Saritha, V., & Obaidat, M. S. (2016). Learning automata based decision making algorithm for task offloading in mobile cloud. In *Computer, Information and Telecommunication Systems (CITS). International Conference on, 2016*, pp 1–6.
174. Misra, S., Wolfinger, B. E., Achuthananda, M., Chakraborty, T., Das, S. N., & Das, S. (2019). Auction-based optimal task offloading in Mobile cloud computing. *IEEE Systems Journal*, 13, 2978–2985.
175. Zhou, B., Srirama, S. N., & Buyya, R. (2019). An auction-based incentive mechanism for heterogeneous mobile clouds. *Journal of Systems and Software*, 152, 151–164.

176. Zhang, D., Tan, L., Ren, J., Awad, M. K., Zhang, S., Zhang, Y., et al. (2019). Near-optimal and truthful online auction for computation offloading in green edge-computing systems. *IEEE Transactions on Mobile Computing*, 19, 880–893.

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



Amir Masoud Rahmani received his BS in Computer Engineering from Amir Kabir University, Tehran, in 1996, the MS in Computer Engineering from Sharif University of Technology, Tehran, in 1998 and the PhD degree in Computer Engineering from IAU University, Tehran, in 2005. Currently, he is a Professor in the Department of Computer Engineering at the IAU University. He is the author/co-author of more than 200 publications in technical journals and conferences. His research interests are in the areas of distributed systems, Internet of things and evolutionary computing.



Mokhtar Mohammadi received the B.S. degree in computer engineering from Shahed University, Tehran, Iran, in 2003, the M.S. degree in computer engineering from Shahid Beheshti University, Tehran, Iran, in 2012, and the Ph.D. degree in computer engineering from Shahrood University of Technology, Shahrood, Iran, in 2018. His current research interests include signal processing, time-frequency analysis, and machine learning. He is currently with the Department of Information Technology, Lebanese French University-Erbil, Iraq.



Adil Hussein Mohammed born in Baghdad, Iraq 14/6/1964, Finished all studies in Baghdad, working research and development center from 1987 to 2003, then moved to Syria and work in Al Kalamoon University KOU before returning back home to Erbil in 2012 to work in Cihan University, nowadays, I am the head of communication and computer engineering department.



Sarkhel H. Taher Karim received a BSC degree in Computer Science from the University of Sulaimani, Sulaymaniyah, Iraq, in 2007 3rd out of twenty graduated. I started working at the University of Sulaimani 2007–2009. I did my master degree 2011 in Computer Science, Faculty of Management and Information Technology, Jamia Hamdard University, New Delhi, India. Lecturer at University of Sulaimani, College of Science, Computer Department 2011–2016. Currently, I'm Head of Computer Department College of Science, University of Halabja, Halabja, Iraq Since 2016. My current research interests include Recommender Systems, Social Network Analysis, speech, dialogue and natural language processing, neural network, deep learning and Internet of Things.



Mohammed Kamal Majeed received the B.Sc. degree in Computer Engineering from Tishk International University, Iraq in 2013, and M.Sc. degree in Computer Engineering from Near East University, Cyprus in 2018. Currently, he works as Lecturer at Tishk International University, Iraq. His current research interests are Pattern Recognition, Machine Learning, and Cyber Security.




Mohammad Masdari received his B.Tech. degree in Computer Software Engineering from Islamic Azad University, Qazvin Branch, Iran, in 2001, and M.Tech degree in Computer Software Engineering from Islamic Azad University, South Tehran Branch, Tehran, Iran, in 2003. He received his Ph.D. degree in Computer Software Engineering from Islamic Azad University, Science and Research Branch, Tehran, Iran, in 2014. Since 2003, he worked as a faculty member of Islamic Azad University, Urmia Branch, Iran. Presently he is an Assistant Professor in the Department of Computer Engineering of Islamic Azad University, Urmia Branch, Iran. His research interests include Distributed Systems and Network Security.



Mehdi Hosseinzadeh received his B.S. degree in computer hardware engineering, from Islamic Azad University, Dezfol branch, Iran in 2003. He also received his M.Sc. and the Ph.D. degree in computer system architecture from the Science and Research Branch, Islamic Azad University, Tehran, Iran in 2005 and 2008, respectively. He is currently an Associate professor in Iran University of Medical Sciences (IUMS), Tehran, Iran. He is the author/co-author of more than 120 publications in technical journals and conferences, and his research interests include SDN, Information Technology, Data Mining, Big data analytics, E-Commerce, E-Marketing, and Social Networks.

Authors and Affiliations

Amir Masoud Rahmani^{1,2} · **Mokhtar Mohammadi**³ · **Adil Hussein Mohammed**⁴ · **Sarkhel H. Taher Karim**^{5,6} · **Mohammed Kamal Majeed**⁷ · **Mohammed Masdari**⁸ · **Mehdi Hosseinzadeh**^{9,10} 

Amir Masoud Rahmani
arahmani@khazar.org

Mokhtar Mohammadi
Mukhtar@lfu.edu.krd

Adil Hussein Mohammed
adil.mohammed@cihanuniversity.edu.iq

Sarkhel H. Taher Karim
Sarkhel.kareem@uoh.edu.iq

Mohammed Kamal Majeed
mohammed.kamal@tiu.edu.iq

Mohammed Masdari
M.Masdari@iaurmia.ac.ir

¹ Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan, R.O.C.

² Department of computer science, Khazar University, Baku, Azerbaijan

³ Department of Information Technology, Lebanese French University, Kurdistan Region, Erbil, Iraq

⁴ Department of Communication and Computer Engineering, Cihan university-Erbil, Kurdistan Region, Erbil, Iraq

⁵ Computer Department, College of Science, University of Halabja, Halabja, Iraq

⁶ Technical College of Informatics, Computer Networks Department, Sulaimani Polytechnic University, Sulaymaniyah, Iraq

⁷ Science Department: Information Technology, Tishk International University, Erbil, Iraq

⁸ Department of computer science, Urmia Branch, Islamic Azad University, Urmia, Iran

⁹ Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

¹⁰ Computer Science, University of Human Development, Sulaimaniyah, Iraq