Efficient Offloading Schemes Using Markovian Models: A Literature Review

Article *in* Computing · July 2020

DOI: 10.1007/s00607-020-00812-x

2 authors:

Mohammad Masdari
Islamic Azad University of Urmia
61 PUBLICATIONS **801** CITATIONS
SEE PROFILE

Hemn Khezri
Young Researchers and Elite Club, Sardasht Branch, Islamic Azad University, Sard…
6 PUBLICATIONS **19** CITATIONS
SEE PROFILE

Computing
https://doi.org/10.1007/s00607-020-00812-x

REGULAR PAPER

Efficient offloading schemes using Markovian models: a literature review

Mohammad Masdari[1] · **Hemn Khezri**[2]

Received: 21 August 2019 / Accepted: 3 April 2020

Abstract
The increasing demand for new mobile applications puts a heavy demand for more processing power and resources in smart mobile devices (SMD). Offloading is a promising solution for these issues which tries to move data, code, or computation from the SMDs to the remote or nearby resourceful servers. To increase the effectiveness of the offloading process and make better decisions, various stochastic offloading schemes are proposed in the literature which has adapted different stochastic models. Although offloading issues are vastly studied in the literature, there is a lack of comprehensive paper to focus on stochastic offloading solutions. This paper presents a meticulous review and classification of the stochastic offloading frameworks designed for different environments such as mobile cloud computing, mobile edge computing), and Fog computing. Following this, it first presents the required background concepts and key issues regarding the offloading problem and stochastic models. It then puts forward a taxonomy of the stochastic offloading approaches according to their applied stochastic models and highlights their architectures and contributions. In addition, in each category, a comparison of the stochastic offloading schemes is provided to illuminate their features. Finally, the concluding remarks and open research areas.

Keywords Cloud computing · Fog · Edge · Offloading · Markov · Semi-Markov · MDP · HMM

Mathematics Subject Classification 68Uxx

✉ Hemn Khezri
HemnKhezri21@Gmail.com

Mohammad Masdari
M.Masdari@Iaurmia.ac.ir

[1] Computer Engineering Department, Urmia Branch, Islamic Azad University, Urmia, Iran

[2] Computer Engineering Department, Afagh Higher Education Institute, Urmia, Iran

Published online: 10 May 2020

1 Introduction

Cloud computing is a promising paradigm aimed at providing various virtualized resources to cloud users/applications, according to the pay for use pricing model [1, 2]. Ideally, it tries to host various applications and services regarding QoS attributes specified in the service level agreements (SLA) [3, 4]. Currently, various applications are designed for smart mobile devices (SMDs) and their demand for more computation and storage resources increases rapidly. For instance, various data analytics applications in different contexts such as e-healthcare, bio-medical, e-commerce, etc. are proposed which require access to large databases and need high processing power. However, the local execution of mobile applications can drain the battery power of the SMD and may lead to performance degradation [5]. This issue has inspired numerous researchers to further augment the SMDs' capabilities by offloading [6] their computation loads [7], data, codes, and applications [8, 9] on more powerful remote mobile cloud computing (MCC) servers. To mitigate the MCC load [9–12], newer technologies such as fog computing [13, 14], mobile edge computing (MEC) [15], and cloudlets [16] are used to handle the users' workloads in the nearby locations instead of remote servers. Nevertheless, offloading is a challenging NP-hard problem and it should be decided which components should be offloaded? Where they should be offloaded? When they should be offloaded, and how this process must be conducted? [17, 18]. To provide a power-efficient offloading service, different influencing factors such as network bandwidth, mobility [19], heterogeneity, coverage of the WAPs, the geographical distribution of SMDs, monetary cost, security [20], and QoS should be considered in the offloading process. For this purpose, various techniques such as multi-criteria decision making [21, 22], optimization algorithms [23], game-theoretic models [24], and stochastic modeling are applied to make optimal offloading decisions in the SMDs, which the latter is the focus of this article. Generally, a model can be defined as a simple representation of a system and as shown in Fig. 1, a system can be modeled using deterministic and stochastic models, which the former models do not take the chance factor into account.

However, the stochastic models are more realistic and consider the inherent uncertainty in real environments. Furthermore, the stochastic models can be further classified as static and dynamic models, in which the first category does not take time element into account while the second category does. For instance, a Markov chain model is

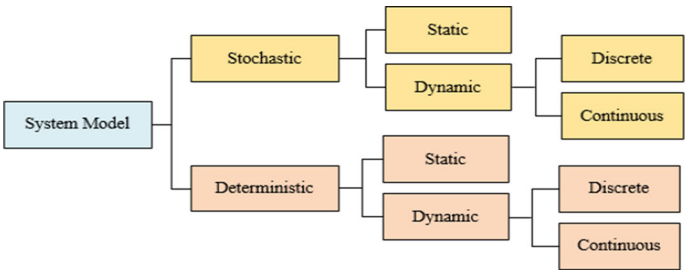


Fig. 1 Classification of system modeling [4]

dynamic because it is able to model the states change over time. Besides, dynamic models can be further classified into continuous-time and discrete-time models, wherein the discrete-time stochastic models, the time advance can be considered at discrete points, but in the continuous models, time advances continuously. Stochastic offloading schemes employ various stochastic models to make better or even optimal decisions by predicting the probability of future results and states using probability data. Although the offloading problem is extensively studied in the literature [11, 25–28], there is a lack of review paper to put forward an in-depth investigation of the stochastic offloading schemes. This paper provides a meticulous survey on the stochastic offloading schemes designed to handle computation, code, and data offloading in various environments such as MCC, MEC, fog computing, and cloudlets. It thoroughly discusses various issues and features concerning the offloading schemes and then provides a taxonomy of these schemes according to their utilized analytical models. It also illuminates the role of the stochastic models in the investigated offloading schemes and describes their architecture, communication models, their possible limitations, and advantages. Also, various properties of the stochastic offloading approaches are highlighted and compared in each section. Finally, the concluding issues and open challenging issues in the stochastic offloading domain are illuminated. According to our studies, this is the first survey aimed at providing a comprehensive review and classification of the stochastic offloading frameworks. Our contributions in this survey can be summarized as follows:

- Basic issues and the major challenges regarding the code, data, and processing of offloading in different environments are provided.
- A taxonomy and a meticulous review of the stochastic offloading literature are provided. In addition, the comparison of their applied evaluation factors, simulators, and various features are provided.
- Finally, based on the information resulted from the studied schemes, an elaborative comparison is supplied and the roadmap for future research and open problems in the stochastic offloading context are highlighted.

The remainder of this article is structured as follows: Sect. 2 presents the research methodology applied in this paper, Sect. 3 provides the required background knowledge about the stochastic offloading, and Sect. 4 introduces a taxonomy and review of the stochastic offloading solutions while detailed information about each scheme is compared with others. Also, Sect. 5 puts forward a comparison of the explored offloading solutions and finally, Sect. 6 presents the concluding remarks and future researches area. Table 1 indicates the abbreviations applied in the rest of this article.

2 Research methodology

This section introduces the systematic literature review methodology [29] conducted for the stochastic offloading schemes proposed in the cloud computing-related literature. At first, for finding review and survey articles in the stochastic offloading context, we employed the following search strings in the Google scholar:

- Stochastic offloading survey cloud computing

Table 1 Applied abbreviations

Abbreviation	Description
CSP	Cloud service provider
D2D	Device to device
DAG	Directed acyclic graph
DC	Data center
DTMC	Discrete time Markov chain
DVFS	Dynamic voltage and frequency scaling
ERTP	Energy-response time product
HMM	Hidden Markov model
MC	Markov chain
MCC	Mobile cloud computing
MDP	Markov decision process
MEC	Mobile edge computing
QoS	Quality of service
SLA	Service level agreement
SLAV	Service level agreement violation
SMC	Semi-Markov chain
SMD	Smart mobile device
WAP	WiFi access points

- Stochastic offloading review cloud computing

However, we found no paper satisfying these search strings. Subsequently, for finding the general review and survey articles in the offloading context in the cloud computing, fog computing, and edge computing the following expressions are searched in the title of the articles at the Google scholar:

- Offloading survey cloud computing
- Offloading review cloud computing
- Offloading survey fog computing
- Offloading review fog computing
- Offloading survey edge computing
- Offloading review edge computing

These searches result in some interesting review articles referenced in the introduction of this paper. Nevertheless, as outlined before, these articles study the general properties and features of the offloading schemes and do not focus on any specific method for dealing with the offloading problem. Besides, for finding the new proposals and research articles in the stochastic offloading context, the following search strings are searched in the Google scholar:

- Markov offloading cloud computing
- Markov decision process offloading cloud computing
- Hidden Markov model offloading cloud computing
- Semi-Markov offloading cloud computing

- Semi-Markov decision process offloading cloud computing
- Markov offloading fog computing
- Markov decision process offloading fog computing
- Hidden Markov model offloading fog computing
- Semi-Markov offloading fog computing
- Semi-Markov decision process offloading fog computing
- Markov offloading edge computing
- Markov decision process offloading edge computing
- Hidden Markov model offloading edge computing
- Semi-Markov offloading edge computing
- Semi-Markov decision process offloading edge computing

The results achieved from these searches are screened to find credible and original articles. For example, documents such as thesis, patents, and papers from the journals which were not provided by the publishers listed in Table 2 were excluded. The remaining of these articles are used in conducting this study and will be reviewed in the next section. Figure 2 depicts the number of the proposed stochastic offloading schemes published in each year since 2010 for the edge, fog, and MCC. As shown in this Figure, the number of these schemes is increasing and this context can be considered as an active research area. Furthermore, Table 3 describes the main research questions considered in this paper and the reasons which they are needed.

Table 2 Applied publishers

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/

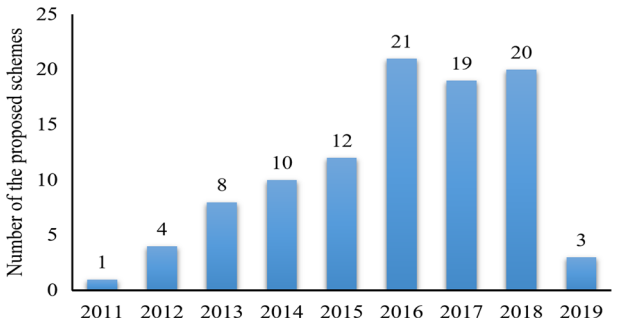


Fig. 2 Publication year of the stochastic offloading schemes

Table 3 Research questions

Index	Questions	Reasons
1	Which environments are supported by each offloading scheme?	The studied stochastic offloading schemes may be designed for fog computing, MEC, cloudlet assisted MCC and MCC. Each of these environments has different requirements and their different features are considered by each scheme. This question is answered in the comparison tables
2	Which kinds of stochastic models are used in each studied scheme?	Each examined offloading scheme may try to model the offloading process, the communication channel, or even SMDs' mobility, and so on. Also, some schemes may use multiple stochastic models to consider different issues which should happen in the offloading process to achieve their offloading policies. This question is answered in the specification of each scheme provided in Sect. 4
3	Which major contributions are provided by each stochastic offloading framework?	Each offloading solution has focused on some issues in the offloading context. Illuminating these contributions can be useful for finding the open issues in the stochastic offloading context and directing future researches in the proper directions. This question is answered in the specification of each scheme provided in Sect. 4
4	Which kinds of communication links and offloading methods are supported by each studied stochastic offloading scheme?	The offloading schemes may support various communication methods such as device-to-device (D2D) links and delayed offloading. Having this information is useful in recognizing the key points of the offloading schemes and determining the less investigated area. This question is answered in the specification of each scheme provided in Sect. 4
5	Which simulators are applied to evaluate the effectiveness of each offloading scheme?	Illuminates how the offloading schemes can be simulated and evaluated. It also determines that, is there any special purpose simulator for this context? This question is answered in the comparison tables provided in Sect. 4
6	Which metrics are considered in the evaluation and analysis of the stochastic offloading schemes?	Different metrics are used by the proposed offloading solutions in their simulation process, which identifying them can be useful in designing new offloading solutions and exploring other important factors. This question is answered in the comparison tables provided in Sect. 4

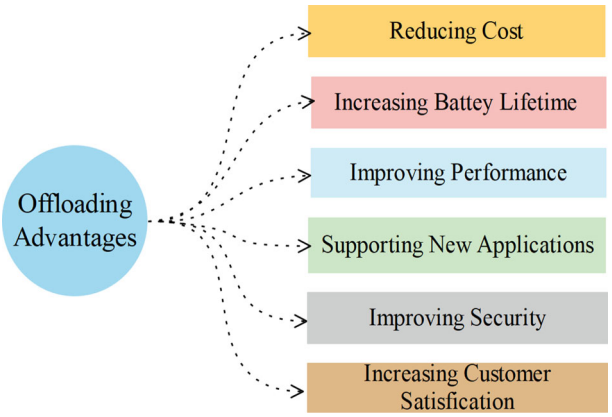


Fig. 3 Advantages of offloading

3 Properties of the offloading schemes

This section provides essential background knowledge about offloading and discusses various properties of the offloading frameworks.

3.1 Offloading advantages

Figure 3 provides the advantages of the offloading for SMD users and CSPs [30]. As shown in this figure, offloading is able to mitigate the costs incurred to the users for using the cellular networks by transferring traffic on the available WiFi links as much as possible. Since, the SMD can be relieved from the offloaded processing, its battery power can be preserved and it can be utilized for longer time periods. Also, by offloading the processing load on the remote servers, applications can benefit from powerful processing resources and their performance can be improved. Furthermore, by enabling remote execution, more applications can now be supported leading to an increase in the usefulness of the SMDs.

In addition, from the security point of view, depending on the type of application, offloading can either decrease or increase security. For example, when the offloading leads to less transferred items, it can increase security. For example, consider a scenario in which an application on the mobile device needs data stored on the server-side; by offloading application on the server, there will be no need to transfer server-side data to the mobile node, which will increase the data security. On the other hand, when offloading causes further transfer of items such as code, data, applications on the network, it can mitigate the security of such items.

3.2 Offloading timing

Generally, offloading schemes can offload items such as data, applications, and code. In this context, regarding the offloading timing, they can be classified as on-the-

spot offloading and delayed offloading methods, which in the first case, offloading can be performed when the WiFi link is available, but in the second case, offloading will be performed using the cellular links. However, in the delayed offloading method, when a WiFi link is not available, offloading can be delayed regarding the deadlines. If until the deadlines no WiFi link can be reached, the offloading will be conducted using the cellular network. Generally, to reduce power consumption and monetary costs, it is always better to wait and find a WiFi link rather than local computation or offloading using cellular links, but this totally depends on the deadlines which should be guaranteed in the offloading process. Moreover, offloading components can be transmitted in advance to an offloading server (pre-offloading), or they can be transferred on-demand at the offloading time. The pre-offloading method obtains higher performance during the program execution, but it requires tuning an offloading server. The on-demand offloading method transmits the offloading components at the runtime and does not require a pre-configuration of a remote server. But, the overhead of transmitting the required components is higher. Also, it requires searching for a proper server. Moreover, offloading decisions can be made statically or dynamically, which in the first case, offloading decisions are made during the developments of the programs and in the second case, offloading decisions can be made during run-time according to different factors.

3.3 Offloading links

An SMD can benefit from cellular networks, WiFi links, or D2D communications [2, 27, 31] (Wi-Fi Direct and Bluetooth) to communicate to conduct offloading. However, when an SMD wants to contact an offloading destination using the cellular networks, there will be a queue because of the low transmission speed of the wireless cellular links. In general, offloading duration may consist of the following items:

- Uploading time of the data, code, or computation/application.
- Processing time in the offloading destination (offloading destination can benefit from other servers and may offload the received items from the SMDs to them).
- Downloading time to return the achieved results to the SMDs.

The WLAN-based offloading incurs a very low cost, but for mobile users, the WAP availability is intermittent and the offloading may encounter long delays until the SMD arrives at the coverage area of the next WAP. Because, in computation offloading, the execution time heavily depends on the cost of input/output data transfer over the wireless link, the data transmission rate will have a significant impact on the offloading decisions. When a user program or workflow is going to be executed, it should be decided to offload which parts of the program. In this case, based on the offloading policies or various factors such as Wi-Fi availability, all of the programs or some parts of them can be offloaded. In this context, various research articles have challenged the application portioning and several schemes are introduced in the offloading literature.

The wireless channel status is another aspect that can be considered in offloading solutions. As shown in Fig. 4, a channel may be assumed to be always accessible

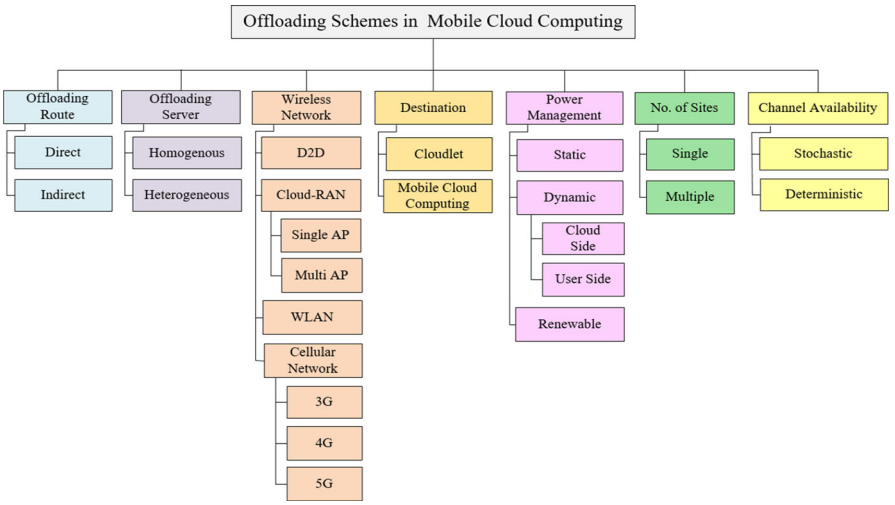


Fig. 4 Properties of the offloading schemes

or intermittently accessible (stochastic channel) [13, 16, 25]. Moreover, as shown in Fig. 4, offloading schemes may benefit from the 3G, 4G, or 5G [5] cellular networks.

3.4 Offloading route

With regard to the offloading route, offloading schemes can be classified 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.

But, in the dynamic offloading, various factors should be considered during runtime to make offloading more adaptable to the dynamic environments.

3.5 Offloading destination

Furthermore, offloading schemes may consider the offloading destination cloud to utilize one or multiple sites, and in the second case, in addition to the destination cloud, the destination offloading site should also be selected. Generally, the offloading destination can be remote MCC servers. However, when the MEC [32] servers are accessible, regarding the high latency of WAN links, the offloading process can be conducted on the Edge servers which are closer to the SMD [33]. In addition, the servers applied in the MCC, MEC [24], and cloudlets can be homogeneous or heterogeneous. As a result, the heterogeneity of the servers should be taken into account in the offloading destination selection.

3.6 Offloading type

With regard to the information required to conduct the offloading process, the offloading decisions can be made globally, based on the situation of the whole system or, locally, based on the situation perceived by SMD [26]. Also, the offloading decision making can be conducted by reactive, proactive, or hybrid methods. In reactive schemes, offloading takes place according to the current situation in the SMD, application, network, etc., while in the proactive schemes, the offloading decision is made based on both current and future situations of the SMD and network.

In addition, offloading approaches can be categorized as direct and indirect offloading schemes. With direct approaches, SMDs send their offloaded entity directly to a server, but in the indirect case, SMDs can utilize other SMDs for conducting offloading on behalf of them. Furthermore, offloading approaches can be categorized as full offloading and partial offloading methods. In the full offloading, a whole program will be offloaded from an SMD to the offloading server. However, when the program is large, it could cause a high network overhead. In partial offloading, only a subset of a program or workflow will be offloaded and incurs lower overheads.

As shown in Fig. 5, with regard to the architecture, offloading schemes can be classified as centralized, distributed, and hybrid approaches. Moreover, regarding the dependency of offloading tasks, they can be categorized as task offloading and workflow offloading which, in the second case for the user, may be considered to offload various scientific workflows. Furthermore, the algorithms exploited for offloading can be classified as single-objective and multi-objective algorithms. Also, the metaheuristic offloading schemes may use single swarm or multi-swarm optimization algorithms.

Regarding delay constraints in the offloading components, as shown in Fig. 5, the offloading operation can be categorized as delay-sensitive and delay-tolerant offloading methods. Generally, in delay-tolerant applications, response time is not very critical, but power consumption is more important. Nevertheless, in delay-sensitive

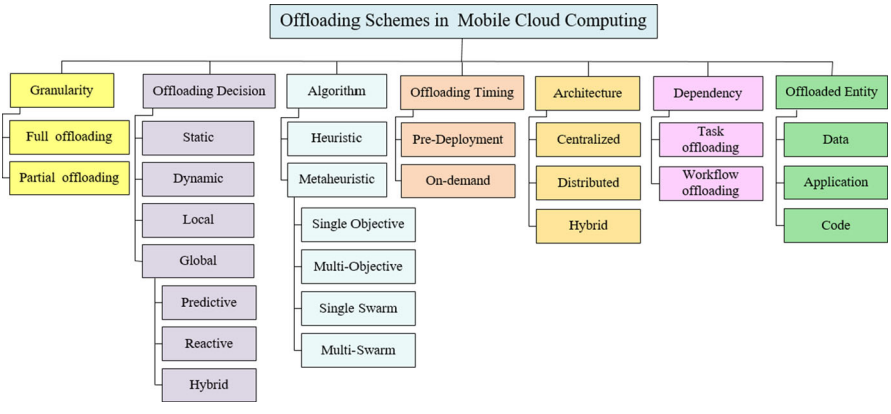


Fig. 5 Properties of the stochastic offloading schemes

applications, low response time is an important factor, and other offloading factors should be considered, while the specified deadlines are met.

3.7 Offloading power management

Offloading schemes may benefit from static and dynamic power management, and even they may also use renewable energies. In the dynamic power management-based offloading solutions, it is aimed at reducing the energy consumption in the SMD side or destination side by using techniques such as dynamic voltage and frequency scaling (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 the application of renewable energy in the destination site, for example on the cloud DCs. In addition, SMDs may select to offload their requests on the cloudlets, on the Fog computing or MCC. Also, the offloading decision can be made locally or globally.

3.8 Offloading architecture

In addition, the offloading schemes may use two-tier or three-tier architecture styles, in the first case of which, an SMD is able to directly offload to the offloading destination such as an MCC. But, in the three-tier architecture, a nearby cloudlet, fog, or MEC servers can be utilized to reduce the response time for the SMDs and mitigate the load on remote cloud servers. However, when the cloudlet cannot handle the workloads in the specified deadlines, it can forward them to the remote cloud server for processing.

3.9 Offloading factors

To achieve offloading objectives, each scheme should consider some factors in the offloading process. Figure 6 indicates some of the factors which can be considered in the offloading process. For instance, the required energy for local processing, processing, and communication delays, offloading costs, security consideration, bandwidth consumption, wireless links status, and reliability of the communication links and offloading destinations are of well-known factors considered by the various examined offloading schemes.

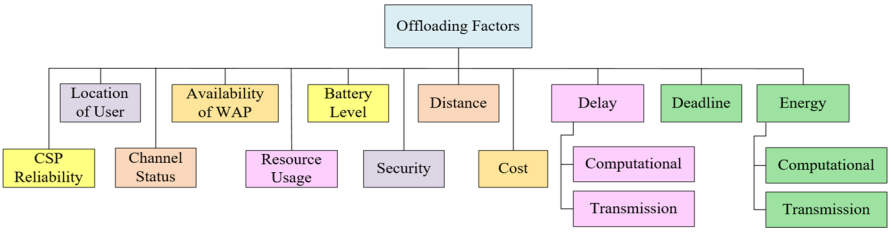


Fig. 6 Offloading factors

4 Stochastic models

This subsection briefly discusses the stochastic models utilized to make offloading decisions in stochastic environments.

4.1 Markov models

A Markov chain is a mathematical tool to model a system during the time which experiences the transition from one state to another according to certain probabilistic rules. Markov chains can be classified as discrete-time and continuous-time Markov chains. Also, based on the number of previous states which they consider for deciding the next state, they can be classified as the first order and high order Markov chains. By definition, in the first-order Markov, each state only depends on its previous state, while in the high order Markov, each state depends on some of its predecessors [15, 29].

4.2 Semi-Markov process

The semi-Markov process is a generalization of the Markov chain where the sojourn times in the states need not be exponentially distributed.

4.3 Markov decision process

MDP or Markov decision process is a well-known discrete-time mathematical framework applied for modeling decision making with uncertainty [34]. An MDP model contains items such as decision epochs, states, actions, transition probabilities, and costs. In addition, an MDP is able to determine agents' actions and their intersections with the environment. Several algorithms such as the value iteration algorithm, policy iteration algorithm, and linear programming are used to solve the MDP models. However, the computational complexity of an MDP model increases with the number of states and the action spaces.

4.4 Hidden Markov models

HMM, or hidden Markov models is one of the most widely applied statistical Markov modeling tools for discrete-time series [32]. In contrast to the Markov chain models where all states are visible, an HMM uses hidden states which are not observable, but its outputs which depend on the states are visible. On the other hand, each state has a probability distribution over the outputs and the term hidden refers to the first-order Markov process behind the observation. The HMM can be used to predict the future state of a stochastic variable.

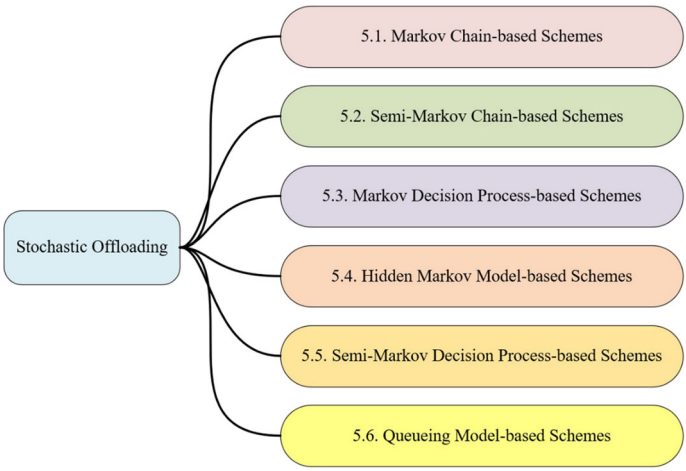


Fig. 7 Classification of the stochastic offloading schemes

5 Proposed stochastic offloading schemes

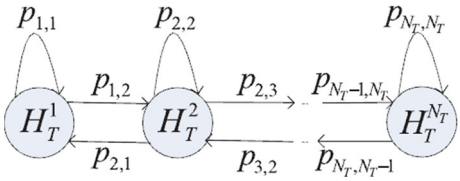
Numerous stochastic offloading schemes such as [22, 35] are provided in the literature, which, as shown in Fig. 7, we classify them based on their applied Markov model. All of the examined stochastic offloading schemes benefit from one or more stochastic models to help them in making better offloading decisions.

5.1 Markov chain-based schemes

Numerous Markov chain-based stochastic offloading schemes such as [36–38] are provided in the literature, which this section addresses them.

In [39], Xiao et al. presented a game theory-based method for mobile offloading and considered three players in which an SMD tries to select its offloading rate, a smart attacker, and a security agent that decides protection for the serving WAP during the offloading process. They found Nash and Stackelberg equilibrium of the offloading game and proposed a Q-learning-based mobile offloading framework for SMDs that are unaware of system parameters, like the channel conditions. As an advantage, for evaluating the effectiveness of their solution, they have conducted security analysis for jamming and spoofing attacks. Figure 8 indicates the Markov model applied for modeling the channel gain between WAPs and SMDs in the offloading process, which also can be used to model the channel gain between the WAPs and attackers.

Fig. 8 Channel model in [39]



The stochastic offloading approach provided in [40], presented a Markov model for performance analysis of Wi-Fi offloading which permits analyzing the non-saturation throughput, the interaction between interfering stations, and the queuing behavior of each SMD. It can predict the throughput and the mean-field approximation is valid in the WiFi offloading. However, this Markov model does not explain the behavior of the contention window for each station.

The work in [41], tried to model an offloading system using a hybrid CTMC and a queuing model which can be under timing attacks. They carried out a quantification analysis for the steady-state behavior of their CTMC model to optimize the weighted-sum cost measure. By transforming the security model to an absorbing model, they computed the meantime to security failure measure. They further computed a security and performance measure based on the system model and found the optimum parameters for the system.

The offloading approach proposed in [42], supports two delayed offloading policies, a partial offloading model where jobs can leave the slow offloading to be executed locally, and a full offloading model, where jobs can be offloaded directly via the cellular network. In both cases, they tried to reduce the ERTP (energy-response time-weighted product) metric. For delay-sensitive applications, the partial offloading model is preferred, while for delay-tolerant ones with the larger deadline, the full offloading can preserve the SMD power. They developed queuing models for delayed offloading to leverage the WiFi and cellular networks by choosing wireless interfaces offloading. Besides, they carried out an optimality analysis of the power and performance trade-off for MCC based on the ERWP or energy-response time-weighted product metric, which captures both energy and performance. In their experiments, they considered the intermittently available links and found that when the availability of the WiFi is low, the percentage of jobs that abandon the queue is very high. For delay-sensitive applications, the partial offloading model is preferred for medium deadlines, while the full offloading model is better for delay-tolerant applications and can reduce power consumption. The authors indicated that an optimal deadline to abort offloading in the partial offloading model and for the WiFi in full offloading model can be found.

In [43], Zhang et al. provided an energy-aware task scheduling by taking into account the collaborative execution among the SMDs and a cloud through task offloading for conserving the power usage. They aimed for power conserving in the SMD under a Markovian stochastic channel. They modeled the minimum-energy task scheduling problem as a constrained stochastic shortest path on a DAG and applied the canonical Lagrangian relaxation based aggregated cost algorithm for solving this problem approximately. They indicated that compared to the execution in the SMD or in the MCC, their proposed collaborative execution approach can mitigate power consumption on the SMD.

The offloading approach in [44], focused on two kinds of delayed offloading models, the partial model where workflows are able to leave from the slow stage of the offloading process, executed on the SMD, and the full offloading model where workflows are able to abandon the WiFi queue, offloaded by the cellular queue. They found that when the availability of the WLAN is small, workflows often abandon the queue.

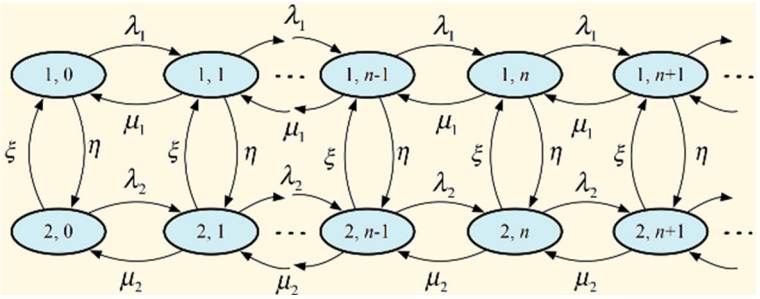


Fig. 9 2D Markov model in [46]

The offloading solution presented in [45], proposed a distributed approach to realize the SNE and quantify its performance gap based on the social optimal algorithm. They provided a distributed reinforcement learning solution that converges to a ϱ-SNE.

In [46], the authors provided two communication methods, the first one of which (interrupted method) chooses the best interface for sending packets and can have interrupts in the WiFi connections causing delays for packets. The other offloading model tries to multiplex data in the available channels and is denoted as an uninterrupted method. Figure 9 shows the 2D Markov model exploited in this scheme for uninterrupted offloading. In this Markov model, two categories or two lines of states are used in which the first line or the (1, n) states determines the states of the cellular network, and the second line, consisting of (2, n) states, indicates the states of the WiFi connectivity. In this Model, *n* denotes the number of jobs in the queues and in service (or the number of jobs in the system).

The solution presented in [47], proposed a fault tolerance technique and a mobility algorithm to optimize the offloading decision in offloading workflows from SMD in the MCC. Their method is able to mitigate the mobile application execution time. They introduced mobility and a fault tolerance technique for calculating the computation offloading cost.

The stochastic framework presented in [48], provided a queuing analytic model for understanding the performance improvements which can be achieved via WiFi-based data offloading, as a function of WiFi performance, availability, traffic workload, user mobility, and the coverage ratio of cellular networks. They focused on the total time a user request spends in the system for queueing and receiving service. They conducted experiments using different traffic patterns, file sizes, and WiFi availability time periods.

The scheme introduced in [49], proposed a hierarchical data offloading solution, which employed different offloading options with various priorities. Figure 10 depicts the four states Markov chain model applied for the SMD data offloading. As shown in this figure, the SMD offloads the cellular traffic on the WiFi connections, but gives less priority to the base station and D2D-based offloading, since the D2D connections may suffer from unreliable communications. By prioritizing WiFi over base station offloading, backhaul traffic load can be reduced.

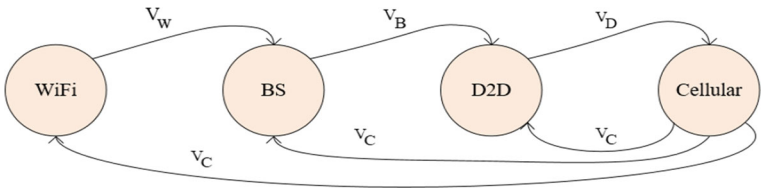


Fig. 10 Markov chain model for the SMD data offloading in [49]

In [50], the authors proposed a preemptive method for code offloading and improve the time to create and transmit safe points to decrease the messaging overhead and mitigate power usage. They applied a predictive method that estimates the mobile link quality to send safe-points before any disconnection of the network.

In [51], Wu et al. presented a delayed offloading model for leveraging the complementary advantage of cellular networks and WiFi when selecting heterogeneous wireless interfaces for offloading. Optimality analysis of the power-delay balance is carried out through employing a queuing model with service interruptions and delay-sensitive workflows, which considers energy and performance metrics, and intermittently available links.

The offloading framework introduced in [52], provided an online prefetching method which combines task level computation prediction and prefetching in the application. It mitigates the SMD's power usage by preventing unnecessary prefetching and lessens the applications' execution time through parallel fetching and computing.

The approach provided in [53], proposed the architecture of a power-aware offloading approach for MCC with regard to the stochastic wireless channels. The architecture of this offloading scheme is depicted in Fig. 11. As shown in this figure, to reduce the power consumption of the SMD and handle the stochastic channel conditions, they exploited DVFS in the SMDs to tune the frequency of CPU or change the data transmission rate dynamically. They considered scheduling problem as a constrained

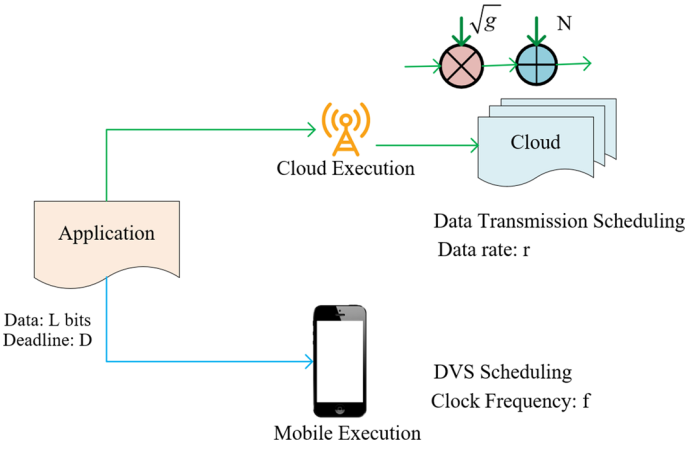


Fig. 11 Mobile application execution in [53]

optimization problem and achieved closed-form solutions for optimal scheduling policies.

In [54], Kim et al. try to find the minimum number of WAPs need for effective offloading by conducting the mathematical analysis. They set the target average per-user throughput when a WiFi network is able to offload traffic from a given cellular network.

The scheme provided in [55], applied a high-order Markov model to provide a data offloading approach with maximum throughput regarding the cellular budget. They presented a dynamic policy that takes into account the SMDs' mobility pattern, cellular budget, and network bandwidth usage. Furthermore, this scheme used an adaptive model to estimate the throughput of WAPs and network usage by SMDs.

5.2 Semi-Markov process-based schemes

This subsection intends to discuss the semi-Markov chain based stochastic offloading schemes. For instance, in [56], the authors proposed to offload the computational load regarding the application features. They handled the dynamic executions of SMD applications utilizing a Semi- Markov chain model and to decide about offloading according to probabilistic estimations of the offloading power saving. They introduced analytical modeling of the execution time and offloading cost.

Furthermore, the stochastic offloading scheme in [57], tried to balance the power usage and performance of the SMDs through application-driven transmission scheduling. They applied the run-time features of applications in delaying wireless transmissions to decrease the power usage and meet the deadlines.

In [58], the authors presented a state transition model for the evaluation of security for an offloading system under the timing attacks, in which the attacker can deduce information about a secret key from successive requests. They developed a semi-Markov chain model for an offloading system under the timing attack. It exhibits the system behavior for a specific attack and system behavior to regularly renew its key to prevent security attacks. Figure 12 shows the state transition model of the proposed semi-Markov chain and described the events which triggered transitions. In this scheme, it is assumed that the system is under the timing attacks conducted by some attackers. After initialization, the system is assumed to be in the good state, denoted by *I,* in which its the sojourn time is the system lifetime before an attacker launches a timing attack or a key renewing process is conducted. When an attack occurs, the system goes to state T, in which the timing attack happens and the attacker

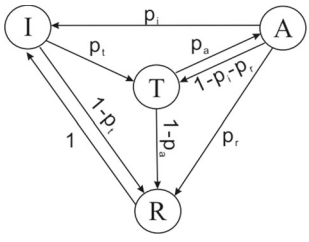


Fig. 12 Semi-Markov chain state transition graph in [58]

decrypts the encryption key. When the system is in state T, the attacker is not able to access private information. In-state R, it is considered that the system to perform rekeying, in which finding an optimal rekeying interval is a challenging issue. At last, in state A, the system is considered to be compromised.

Table 4 presents the properties, evaluation factors and their applied simulators in the Markov chain and semi-Markov chain-based stochastic offloading schemes.

5.3 Markov decision process-based schemes

MDP or Markov decision process is a well-known discrete-time mathematical framework applied for modeling decision making with uncertainty [34]. The MDP model contains items such as decision epochs, states, actions, transition probabilities, and costs. However, the computational complexity of an MDP model increases with the number of states and the action spaces. An MDP can be solved by the value iteration algorithm, policy iteration algorithm, and linear programming. Numerous MDP-based stochastic schemes are provided in the offloading literature.

The work in [63], tried to use energy harvesting in MEC and proposed a reinforcement learning-based resource management algorithm, which learns the optimal policy of workload offloading and MEC server provisioning to reduce the delay and operational cost. This scheme applies an online learning algorithm that uses a decomposition of the value iteration and reinforcement learning, for improving learning rate and performance. They proved the convergence of their algorithm and analytically showed that the learned policy has a monotone structure.

In [64], the authors proposed a code offloading approach based on a deep Q-learning management framework for MEC and mobile fog. They benefited from a distributed network controller which seeks for the free fog/MEC resources. The availability of resources and various options for allocating those resources for offloading fits for modeling through MDP and solution through reinforcement learning. As it is a multi-agent-based distributed method, agents learn from the environment through reinforcements. This framework benefits from two modules denoted as a code analyzer and code offloader, in which the first module determines the code blocks that must be offloaded and the second offloads the code blocks to the Fog. If the CPU and memory needs of the code block are less than the available resources then the application manager executes; otherwise, it determines the expected execution time of host processing based on the available memory and CPU of the host smart device. The trained code offloader makes offloading according to the resource demand, availability, and network status to reduce the latency. By simulations conducted in the MATLAB, the authors indicated that their method can mitigate the delay, execution time, and power usage regarding different offloading decisions. It is considering oscillated resource demands and mobility of end-user devices.

The offloading method introduced in [25], presented an optimal offloading solution for the SMD with the intermittently connected cloudlet, regarding the workload of users and cloudlets availability. They provided an MDP model for obtaining an optimal policy for the SMD to mitigate offloading costs and computation. The architecture of this framework is provided in Fig. 13.

Table 4 Comparison of the Markov chain and semi Markov chain-based stochastic offloading schemes

Simulation factors															
Scheme	Normalized latency	The average utility of the SMD	Through-put	Average queue length	Energy consump-tion	Response time	ERWP	Normalized perfor-mance gap	Total cost	Stopping time	Number of Offloading users	Energy-Response time weighted sum	Execution time	Total traffic offloaded	Arrival rate
[59]	✓														
[39]		✓													
[40]			✓	✓											
[60]			✓												
[43]				✓											
[44]					✓	✓	✓								
[37]								✓	✓	✓					
[45]											✓				
[36]									✓						
[46]					✓		✓					✓			
[47]													✓		
[61]														✓	
[48]															✓
[49]															
[50]					✓								✓		
[51]						✓	✓								
[41]			✓												
[38]					✓										✓
[52]					✓										
[62]											✓				
[42]					✓										
[53]															
[54]															
[55]			✓												
[56]					✓										
[57]					✓										

Table 4 continued

Simulation factors	Properties							Environments				Simulators/environments		
Scheme	Data offloading	Code offloading	Computation offloading	DVFS-based	D2D communication	Delayed offloading	Heterogeneous	MCC		MEC	Fog computing	MATLAB	Samsung Nexus S	LG Nexus 4
								With cloudlet	Without cloudlet					
[59]			✓						✓					
[39]			✓						✓					
[40]			✓											
[60]			✓											
[43]	✓								✓					
[44]	✓					✓	✓		✓					
[37]	✓						✓		✓					
[45]			✓					✓						
[36]	✓						✓		✓					
[46]			✓				✓		✓					
[47]			✓						✓			✓		
[61]			✓			✓			✓					
[48]	✓					✓	✓		✓					
[49]	✓				✓				✓					
[50]		✓							✓			✓		
[51]	✓					✓			✓					
[41]			✓						✓					
[38]			✓				✓		✓					
[52]			✓						✓					
[62]			✓						✓					
[42]	✓					✓	✓	✓						
[53]	✓								✓					
[54]			✓						✓					
[55]	✓					✓			✓					
[56]			✓				✓		✓				✓	
[57]			✓			✓	✓		✓					✓

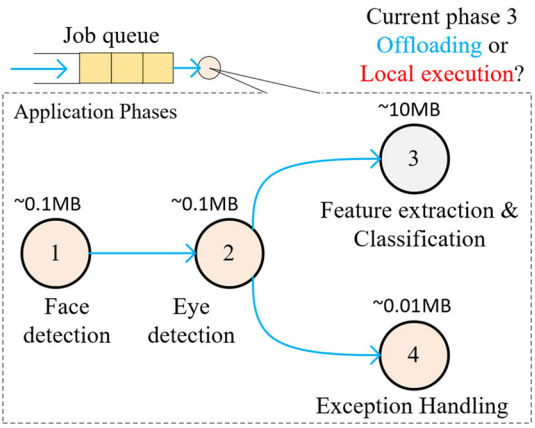


Fig. 13 Application phases in [25]

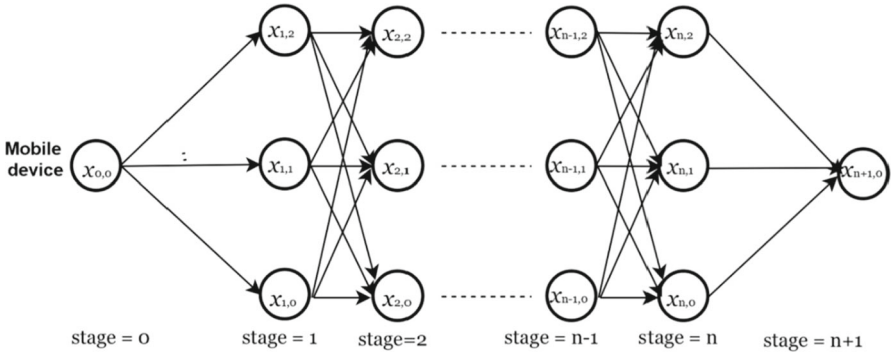


Fig. 14 State transitions in the MDP model in [65]

The authors in [65], the authors presented a multisite offloading method for MEC computing using MDP methodology and provided an MDP-based solution to make optimal offloading decisions and optimizing several goals. They added an edge device between the SMD and CSP to conduct data processing at the network edge to decrease delay, energy consumption, and also network congestion. Following this, they used factors such as computation time and energy consumption in the reward function and utilized the value iteration Algorithm to find a near-optimum value function. Figure 14 exhibits the state transition diagram applied for multisite offloading. They considered two offloading sites in a scenario in which the first site is an MEC server and the second site is an MCC server that has a database. The application of SMD has some components with a linear topology, in which each component can be offloaded into one of the two offloading sites or remain on the SMD in any given step. In this system, each state at each decision epoch indicates the location of the executed component. The decision-maker should evaluate the current state receive a reward depending on the current state. Then, it selects to migrate or continue execution. However, they have not conducted elaborated evaluations to compare their solutions.

In [66], the authors investigated the SMD's policy to mitigate his monetary cost and power usage under time-dependent pricing. They considered wireless LAN offloading

as a finite-horizon discrete-time MDP and provide an optimum policy using dynamic programming.

The framework provided in [67], presented an offloading approach using a deep reinforcement learning method to find near-optimal offloading regarding SMD and cloudlets' mobility and cloudlets' resource availability. They formulated offloading by an MDP which takes into account the SMD's state, cloudlets' queues, and the distance among the SMD and cloudlets. They are aimed at finding the optimal number of tasks which must be processed locally or remotely on the cloudlets to increase the user's utility and reduce the power consumption, delay, cost, and task loss. The deep Q-network (DQN) is exploited to learn the optimal offloading decision for the MDP-based offloading problem since DQN can achieve good performance in the high-dimensional decision-making. In the DQN-based offloading, the user learns online and improves its offloading decisions through a learning method. The state information and learning data are combined in a nonlinear function approximator to form a neural network. But, this scheme is not compared with other offloading solutions to highlight its possible merits.

In [68], Wang et al. provided the edge caching issue as an MDP model and presented a distributed cache replacement method according to Q-learning. To illustrate the D2D offloading effect on the cellular network, they defined the cellular serving ratio, computed through the iterative maximum weighted independent sets for static networks and the stochastic geometry for high dynamic networks.

The scheme introduced in [69], proposed deterministic and randomized methods to detect the optimal radio scheduling-offloading policy. They are able to adapt the processing decisions among local processing, staying idle, and offloading, through applying their knowledge on the channel conditions and the application properties.

In [70], the authors proposed an analytical model to investigate an offloading method for the fog DCs in heavy loads that perform forwarding with some probability request blocked at a DC to another DC. They assumed that the MCC can investigate the offload of requests blocked from a big DC onto a small DC. As an advantage, this method is able to reduce blocking at a small DC without affecting the bigger DC.

The work presented in [71], provided a conceptual model utilizing MDP to decision making and a hybrid approach using a Markov chain model for better prediction of the next suitable offloading server. However, they did not provide the required numerical evaluation results for their offloading solution.

The approach provided in [72], presented a model for offloading computation and data-intensive applications on multiple heterogeneous sites. They also have used MDP for formulating the multisite application partitioning as a delay-aware and least-cost shortest path problem on a state transition graph. In this scheme, the MDP is used to achieve an optimal policy indicating sites to migrate the execution given the current state. They considered minimizing the energy consumption of the application executions and their data transfer in the offloading process regarding the specified deadlines. The authors have benefited the value iteration algorithm to find an optimal solution for the proposed MDP for multi-site offloading. Moreover, they employed a DTMC for modeling the fading wireless mobile channels and considered good and bad states for channel states. The proposed DTMC is applied to compute the average bandwidth of the communication channel. Figure 15 exhibits the state transmission

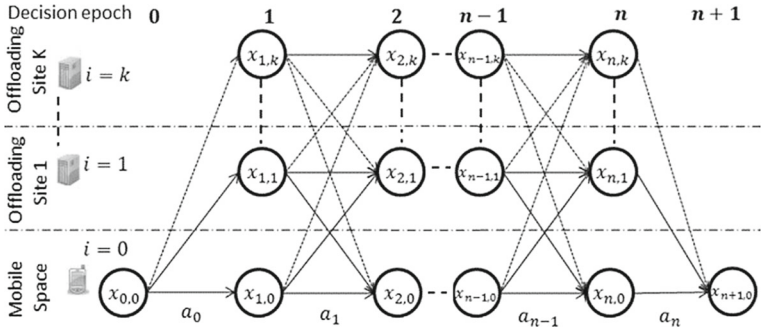


Fig. 15 State transitions of the offloading system in [72]

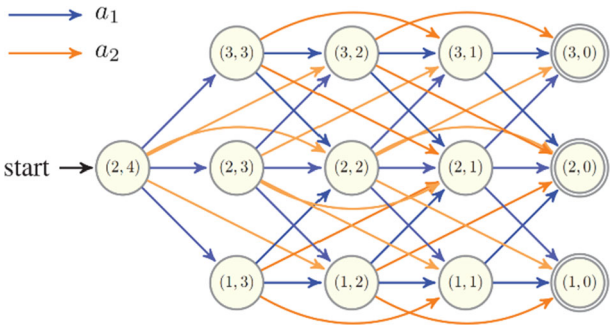


Fig. 16 A reduced state transition diagram for offloading approach in [73]

diagram applied in this scheme. As shown in this Figure, in addition to the SMD, each site can offload its load on the other sites and the SMD itself. However, this scheme does not consider the data structure of a component in the offloading process and may offload different components at the same site. Although they have considered the data transfer costs between offloading source and destination, they assumed that all applications are already replicated on the remote sites and as a result, they did not consider the cost of application offloading. In addition, the authors only have considered a simple linear workflow and failed to provide evaluations on the more complicated workflows such as scientific workflows. The authors have carried out experiments on both data-intensive and computation-intensive applications.

The framework provided in [73], Liu et al. reduced the cost of data delivery tasks while satisfying delay requirements. A portion of the cellular data traffic is offloaded by D2D and WiFi networks. They handled the data offloading problem as a finite horizon MDP and resolved it by employing a hybrid offloading solution. A reduced state transition diagram for SMD is shown in Fig. 16, where the first digit denotes the location of SMD, and the second digit denotes the data size. In addition, in this figure, a1 and a2 are two actions in which the former can send one data size each time whereas the latter can send two data sizes. Also, the states indicated by two circles are terminal states.

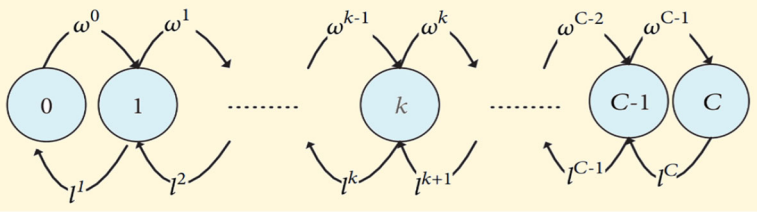


Fig. 17 State transition graph of the Markov model in [77]

In [74], Hyytia et al. provided an MDP-based model to investigate the dynamic offloading in the MCC, which considers different performance metrics and intermittently available wireless links. They applied the queueing theory and used its results to achieve near-optimal offloading policies that concern with the various performance metrics, the current state of the queues, and future tasks.

The approach provided in [75], proposed a dynamic opportunistic offloading solution that enables the user to decide about the offloading or deferring it. They provided an MDP model for the mobile user for obtaining an optimal offloading policy while reducing the cost of processing and offloading.

The offloading approach in [76], presented FDTMDP, a finite-horizon discrete-time MDP with a specific mobility pattern for SMD which uses dynamic programming in offloading solutions. Since SMD's mobility pattern may not be known in advance, they presented a reinforcement learning-based offloading solution, which is able to handle various SMD's mobility patterns. But, they only used one SMD in their scheme, while multiple SMDs should be considered.

In [77], Liu et al. provided COWO, a WiFi offloading approach according to the subgradient technique, which aims at obtaining the optimal offloading ratio for each WAP to decrease the network congestion and increase the throughput. Due to the complexity of the subgradient technique, they enhanced the COWO approach by the equivalent transformation. The state transition graph of the Markov model proposed in this solution is indicated in Fig. 17. As shown in this figure, they formulated the offloading model of WiFi using the discrete Markov process and formulated the channel occupation by a discrete Markov state, the service arriving indicates that the user enters the network service sequence, and left indicates that the user discontinues the network service. In this figure, the number of channels denoted by indicates the maximum number of users which can be served by the network. However, they did not consider SMDs' mobility pattern which is very crucial for solving the congestion problem in WLAN.

The work in [78], handled the problem of multi-flow offloading in which an SMD has some traffic flow with various deadlines and loads. They considered the multi-flow rate control as a discrete and finite-horizon Markov decision problem and developed an optimal rate control solution using dynamic programming to improve offloading efficiency as well as meeting deadlines. As an advantage, their solution supported delayed offloading. However, they have only focused on the download traffic and did not consider the upload traffic.

In addition, CEMMO is a cost-aware stochastic offloading framework aimed at improving the offloading process using multi-hop communications [79]. It uses cellular delivery, delay-tolerant offloading, and peer-assisted offloading. Using user mobility data and WiFi, CEMMO assists the cellular operator in selecting its best mode to decrease the cost based on the financial settlement, power usage, and user satisfaction.

In [80], Liu et al. studied the congestion offloading with SMD's random mobility model and proved it could be resolved in a subgradient technique with equivalent transformation. They presented a congestion-aware WiFi offloading solution for a scenario with multiple WAPs to tradeoff the capacity increase with congestion detected through blocking probability.

In [81], Le et al. assumed that the SMD has a bag of tasks in which a number of them can be executed locally and the others should be offloaded to the mobile cloudlets. They introduced an MDP-based offloading optimization approach that enables the SMD to perform optimal offloading to increase its utility and reducing the overheads. It considers the effects of the user's load, the diverse connectivity of cloudlets and the wireless environment on the offloading action. They developed an MDP for the SMD to obtain an optimal offloading policy for workload distribution to decrease the communication and processing overheads and improve the user's utility. The MDP considers the heterogeneous connectivity to the mobile cloudlets, the distance to the cloudlets, and the fading wireless channel to estimate the cloudlet's availability and offloading cost. They determined an optimal workload distribution to assign the tasks which should be processed locally or offloaded to the mobile cloudlets. The semi-Markov smooth (SMS) mobility model is exploited and MCs mobility model. In the SMS model, each movement has three consecutive moving phases.

The offloading solution presented in [82], applies reinforcement learning and a game-theoretical model for resource management in the edge computing. In addition, it provides a minority game-based model for power-efficient distributed edge server activation and investigated many learning methods in their approach. In this scheme, a pool of some edge servers is considered which receives a fixed number of offloaded tasks that are delay-sensitive and have a deadline. At each time, some servers are active and the offloaded tasks are divided among them. Every active server is reimbursed for its performed tasks. Thus, the number of tasks per server shall be large enough to ensure an acceptable revenue. Also, the required number of servers at each round can be determined regarding the system energy-efficiency and the user's quality of experience with high probability.

The stochastic scheme introduced in [83], presented LODCO, a Lyapunov optimization-based low complexity dynamic computation offloading approach to decide about the offloading on the MEC servers, the CPU frequencies for SMD, and the transmit energy for offloading. As an advantage, its decisions depend on the current state and it does not require any historical data. In this scheme, the execution cost is based on the delay and task failure is adopted as the performance metric, while DVFS and power control are adopted to optimize the mobile execution process and data transmission for computation offloading. Also, the execution cost minimization problem is formulated as a high-dimensional Markov decision problem. In each time slot, the system operation, including the offloading decision, the CPU-cycle frequencies for mobile execution, and the transmit power for computation offloading, only

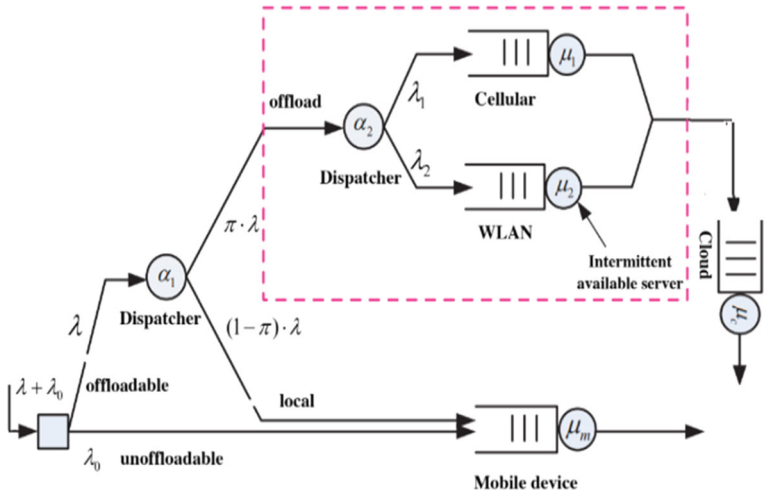


Fig. 18 A queuing model for MCC offloading in [85]

depends on the optimal solution of a deterministic optimization problem, which can be obtained either in closed form or by bisection search. The authors showed that LODCO can reduce the execution cost and task failures. But, the authors have only evaluated the delay and number of local executions, while more factors should be evaluated to indicate energy efficiency.

In [84], Hyytia et al. provided a stochastic model using the MDP which considers different performance metrics and available access links in the MCC and assumed that an offloaded workflow via a WAP cannot be re-assigned to the cellular links. They achieved queuing theoretic results to find near-optimal offloading policies.

In [85], the authors applied the queueing models to mitigate the weighted sum of power usage and performance expressed in the ERWS metric. Added to this, they take various offloading policies into account, where arriving workflows are processed locally or remotely in the cloud. Offloading can be conducted through WLAN or by a cellular network. The queuing architecture of this scheme is shown in Fig. 18. As shown in this figure, offloadable and unoffloadable jobs arrive at the SMD. In this scheme, the system behaves like an M/M/1 queue for the offloadable jobs. Also, an SMD user moves around the coverage area of the WiFi and this time variation of the WiFi connection state is modeled by an ON–OFF renewal process.

The work in [86], introduced ST-CODA, a spatial and temporal computation offloading algorithm in MEC, in which an SMD may decide to offload to the cloud, the edge cloud, or delay the processing of tasks. In ST-CODA, an SMD decides where and when to process tasks using an MDP according to the processing time and energy consumption of different computation nodes and the transmission cost in the heterogeneous MEC. The authors obtained the optimal policy on the spatial and temporal offloading by the value iteration algorithm and optimized the performance of ST-CODA. Furthermore, they evaluated the effectiveness of ST-CODA with the optimal

policy according to transmission cost, the energy efficiency of the SMD, complexity, and the number of processed tasks.

Table 5 exhibits the evaluation factors, the simulator software, and various properties of the MDP-based stochastic offloading approaches.

5.4 Semi-MDP based schemes

This subsection studies the stochastic offloading schemes which have used semi-MDP models in the offloading process.

In [90], the authors provided a cloud broker model to provide an optimal allocation of several cloud resources to the SMDs' requests with various requirements. The cloud brokering is dealt with as a semi-MDP model considering the average system cost which consists of the communication costs, the cost of computing resources, the request traffic, and security risk degrees, and resource for the mobile users.

The approach presented in [91], provided a cloudlet cooperation method, where the bus-based cloudlets are intended to handle the SMDs' workflows. They used a semi-MDP architecture for formulating this problem as a delay-constrained shortest path problem.

The offloading method provided in [92], proposed a model based on semi-MDP to estimate the amount of future offloaded workload. User satisfaction is modeled as a satisfaction function of the delay period. They proposed a series of approaches for detecting the optimal handing-back time from offloading.

The approach provided in [93], aims to handle optimal task dispatch, transmission, and execution in the MCC. To mitigate power consumption, they have exploited DVFS in the SMD's processor, whereas the RF transmitter can select different modulation approaches and bit rates. Also, they applied an accurate and realistic battery model to predict the battery power loss rate in a more accurate way. The block diagram of this offloading framework is shown in Fig. 19.

In [94], Zhuo et al. are aimed at providing a trade-off between the offloaded traffic and the users' satisfaction. They proposed an incentive architecture to motivate SMDs to delay their cellular traffic offloading. Added to this, they prioritized users with high delay tolerance and large traffic offloading, to decrease the incentive cost for an offloading target.

The offloading approach in [95], proposed a resource allocation model using a semi-MDP regarding the average cost and resolving it by applying linear programming. With increasing the long-term reward while meeting the request blocking probability and service time latency, an optimal resource allocation policy is computed.

The stochastic offloading method provided in [96], tried to optimize the dynamic resource sharing of users in MCC hotspot with a cloudlet by using a semi-MDP and transformed it into a linear programming model and considered QoS for various classes of the SMDs. The authors claimed that their admission control approach is able to obtain good performance and improve the throughput of an MCC hotspot.

Table 6 lists the evaluation factors, environments, offloading types, and other properties of the semi-MDP based stochastic offloading solutions.

Table 5 Comparison of the MDP-based stochastic offloading schemes

Simulation factors													Properties		
Scheme	Throughput	Memory cost	Energy consumption	Response time	ERWP	Offloading ratio	Total ccost	Deadline	Satisfaction gain	Completion time	Network usage deviation	Task loss ratio	Data offloading	Code offloading	Computation offloading
[25]			✓			✓									✓
[87]			✓												✓
[65]			✓												✓
[66]	✓	✓	✓										✓		
[67]			✓									✓			✓
[68]															✓
[69]													✓		
[70]													✓		
[55]	✓										✓	✓			✓
[71]														✓	
[72]			✓												✓
[73]						✓	✓			✓		✓			✓
[74]												✓			✓
[75]						✓							✓		
[76]	✓		✓					✓					✓		
[77]	✓														✓
[78]									✓						✓
[79]						✓							✓		
[88]															✓
[80]	✓														✓
[81]			✓												✓
[82]															✓
[83]										✓					✓
[86]															✓
[63]															✓
[64]			✓	✓											✓
[89]															✓
[84]													✓		
[34]				✓											✓
[85]															✓

Table 5 continued

Simulation factors	Properties							Environments				Simulators/environments		
Scheme	Data offloading	Code offloading	Computation offloading	DVFS-based	D2D communication	Delayed offloading	Heterogeneous	MCC		MEC	Fog computing	Python	ONE simulator	MATLAB
								With cloudlet	Without cloudlet					
[25]			✓					✓						
[87]			✓				✓		✓					
[65]			✓							✓				
[66]	✓								✓			✓		
[67]			✓		✓			✓						
[68]			✓		✓				✓					
[69]	✓								✓					
[70]	✓										✓			
[55]			✓			✓			✓					
[71]		✓							✓					
[72]			✓						✓					
[73]			✓		✓	✓			✓					
[74]			✓						✓					
[75]	✓							✓						
[76]	✓								✓			✓		
[77]			✓						✓					
[78]			✓			✓			✓					
[79]	✓								✓				✓	
[88]			✓							✓				
[80]			✓				✓		✓					
[81]			✓					✓						
[82]			✓							✓				
[83]			✓	✓						✓				
[86]			✓			✓	✓			✓				
[63]			✓							✓				
[64]			✓							✓				
[89]			✓							✓				
[84]	✓								✓					
[34]			✓						✓					✓
[85]			✓				✓		✓			✓		

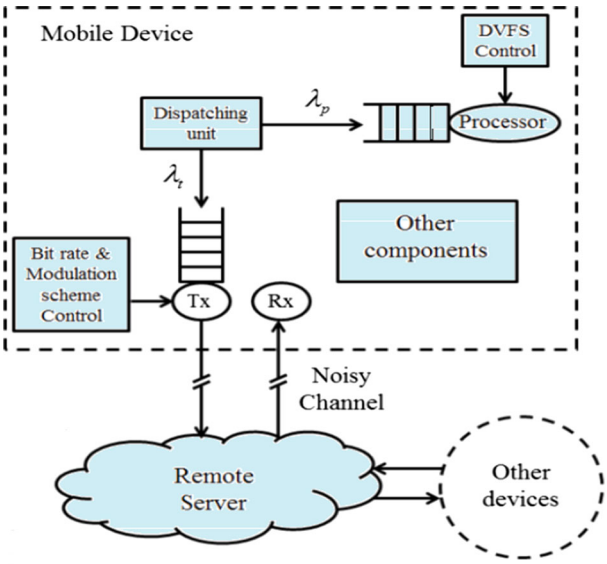


Fig. 19 Offloading framework in [93]

5.5 Hidden Markov model-based schemes

Several HMM-based stochastic offloading schemes such as [98] are provided in the literature, which this section focuses on them. The offloading scheme introduced in [99], proposed an HMM-based dynamic scheduling method to permit the system to adapt to the changing requirements while optimizing the processing latency, diagnosis accuracy, and power usage. They simulated several scenarios and evaluated their scheme regarding various performance factors. The applied HMM model in this scheme is provided in Fig. 20. In this model, various network situations, SMD battery status, and CPU load are considered as the hidden states. Also, the SMD's power efficiency, execution performance, and processing accuracy are considered as outputs from the hidden states. They applied the Baum-Welch algorithm to deal with the HMM learning and by sorting hidden states' output probability, make optimum decisions regarding the statistical features.

The offloading scheme in [100], employs machine learning approaches to handle adaptive scheduling in mobile offloading architecture with high accuracy. The block diagram of this scheme is exhibited in Fig. 21. By using machine learning techniques in the offloading process, a scheduler is trained by past offloading behaviors and also by considering current conditions can decide if the load must be offloaded or handled. It takes into account several machine learning approaches and four loads, with a dataset achieved by the deployment of an Android-based offloading architecture on actual SMD and cloud resources.

Table 6 Comparison of the semi-MDP based stochastic offloading schemes

Simulation factors											Properties	
Scheme	Offloaded traffic	Mean service time	Expected waiting time	Percentage Of offloaded	Energy consumption	Average downloading delay	Trade	Offloading ratio	Offloaded volume	Normalized round trip time	Data offloading	Code offloading
[90]												
[91]					✓			✓				
[58]							✓					
[92]									✓		✓	
[93]					✓					✓		
[94]	✓			✓		✓						
[97]			✓									
[95]		✓										
[96]												

Simulation factors	Properties					Environments				Simulators/environments		
Scheme	Computation offloading	DVFS-based	D2D communication	Delayed offloading	Heterogeneous	MCC		MEC	Fog computing	Python	Qualcomm snapdragon	MATLAB
						With cloudlet	Without cloudlet					
[90]	✓						✓					
[91]	✓					✓						
[58]	✓						✓					
[92]				✓			✓					
[93]	✓	✓					✓				✓	
[94]	✓						✓					
[97]	✓								✓			
[95]	✓						✓					✓
[96]	✓						✓					

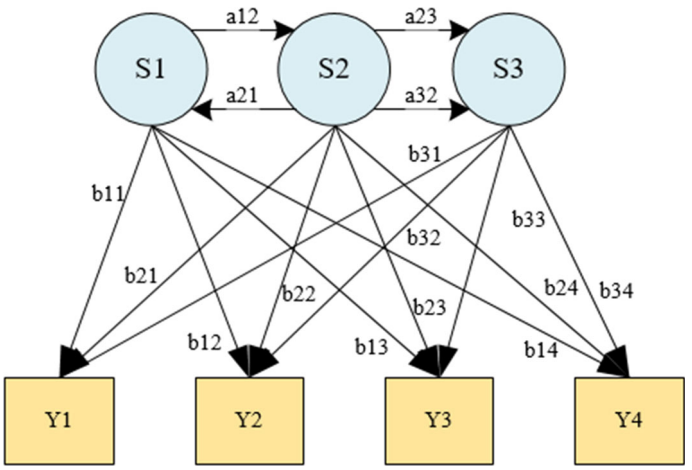


Fig. 20 The applied HMM in [99]

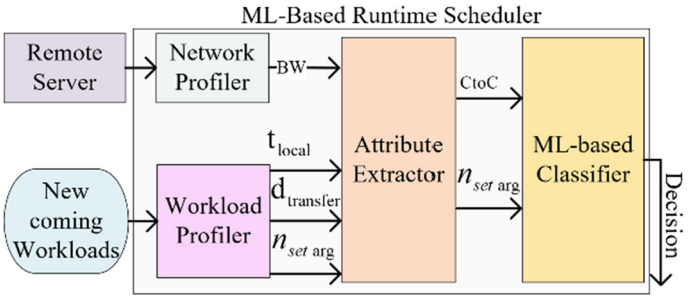


Fig. 21 The block diagram of the solution in [100]

6 Discussion

This subsection presents a detailed comparison of the stochastic offloading schemes, outlined and explored in the previous section. The results of this section can demonstrate future research directions and help to develop new stochastic offloading methods. This section highlights and compares the following features regarding the stochastic offloading schemes:

- Environments applied for evaluation of stochastic offloading schemes.
- Evaluation factors considered in the simulation process.
- Applied simulators and environments.
- Offloading types.
- Power management in stochastic offloading solutions.
- Type of stochastic models.
- D2D support in stochastic offloading schemes.
- Support for delayed-offloading.

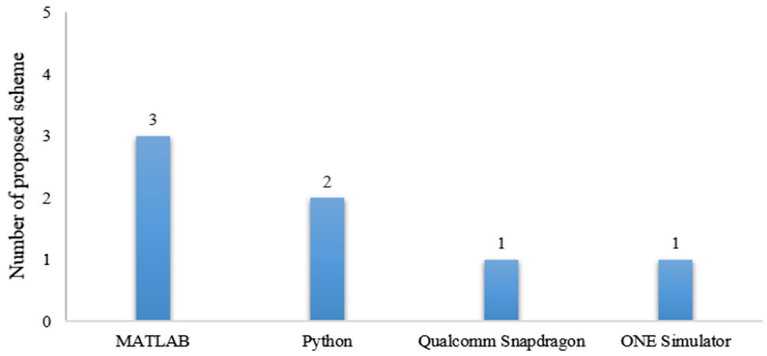


Fig. 22 Simulators applied in the investigated stochastic offloading schemes

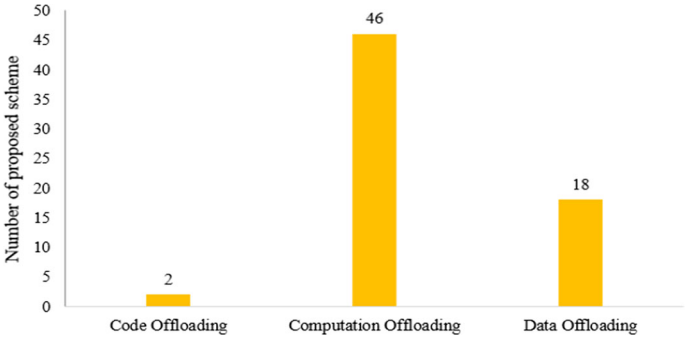


Fig. 23 Offloading types

Figure 22 depicts the type of simulator software and programming environments used to evaluate the proposed stochastic offloading schemes and the number of schemes that have used each of them.

As shown in this Figure, most of the schemes have not specified their applied simulators and a few numbers of the schemes have applied the general-purpose programming language such as MATLAB and Python. This indicates a lack of special-purpose simulation environments to support various types of communications link, protocol, analytical modeling, and different offloading features. Consequently, designing and implementation such as offloading simulators can be challenged in the future. Obviously, such an environment should support various kinds of Markovian models and different types of Queuing models to enable realistic modeling of the offloading environment. Figure 23 exhibits the number of stochastic schemes designed and proposed for code offloading, data offloading, and computation offloading. As shown in this Figure, most of the proposed schemes support the computation offloading and because of the security issues with the code offloading, fewer researches have been devoted to it. To this end, at future studies code offloading can be addressed, while considering its security challenges to provide secure offloading without endangering the offloading destinations.

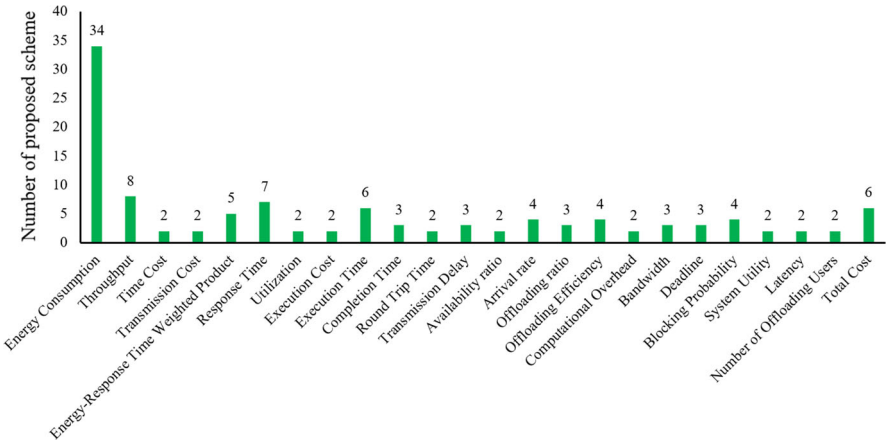


Fig. 24 Parameters applied to evaluate the stochastic offloading schemes

Selecting proper factors for the evaluation of the stochastic offloading schemes is very important. As shown in Fig. 24, a wide range of evaluation criteria is used to assess the performance of the studied schemes. Moreover, this figure exhibits the number of schemes which have applied each evaluation factor. It indicates that energy consumption, response time, and execution time are evaluated by more stochastic offloading schemes.

Figure 25 indicates the type of stochastic models applied in the investigated offloading schemes. As shown in this Figure, Markov chain models and MDP models are applied more than other stochastic models. Mainly, since MDP can be better adapted to the stochastic offloading process, the most number of schemes are devoted to use it and try to find an optimal offloading policy for the SMDs using value iteration and policy iteration algorithms. Furthermore, as shown in Fig. 25, some of the offloading schemes have applied queueing models, in addition to the Markov models. These queueing models are used to model the queues created in the offloading destination for processing the offloading request, or in the intermediate node for routing and processing requests, or in the offloading sources for modeling requests placed in the various available offloading links.

Figure 26a depicts the power management techniques applied in the stochastic offloading solutions. As shown in this Figure, only two DVFS-based stochastic offloading solutions are presented in the literature and most of the introduced offloading approaches use static power management. In DVFS-based Dynamic power management, the operating frequency of the CPU will be tuned to reduce its speed and power consumption while considering the deadlines. Generally, DVFS can be applied at the SMD's side, Server's side, or both of them [16]. Figure 26b depicts the D2D communication support in the stochastic offloading schemes. Regarding the limited coverage of the WLANs, using D2D communications can be very useful in increasing the WiFi coverage. However, as shown in this figure due to various challenging issues of the D2D communications, fewer stochastic offloading schemes are employed it and in the next studies, this type of communications should be further examined. In this context,

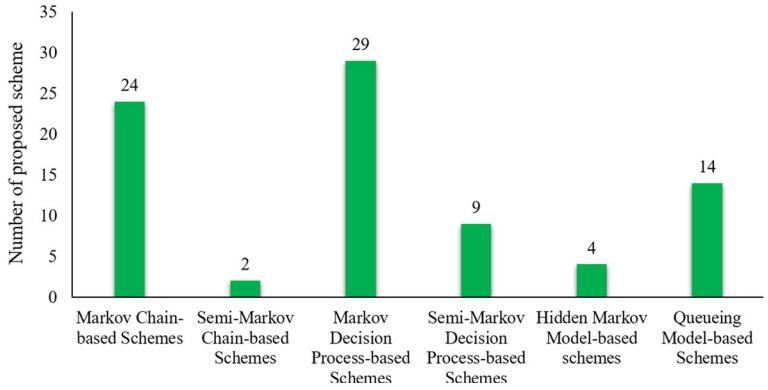


Fig. 25 Type of stochastic models

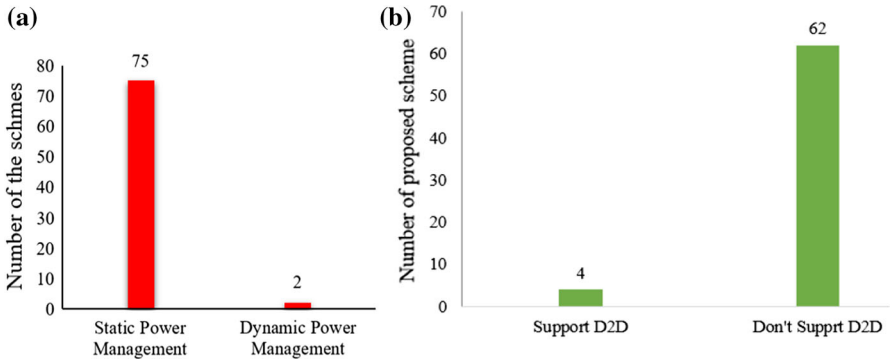


Fig. 26 **a** Power management, **b** D2D support

issues such as security, heterogeneity, and multi-hop D2D communications can be probed and analyzed.

As shown in this figure, conducting offloading using device to device (D2D) communications is one of the areas which has been focused by fewer schemes and in future D2D-based offloading schemes should be further investigated. For example, the following issues in this context can be further studied:

- Which nodes should have the privilege to be a relay node in the D2D offloading?
- What actions should be taken regarding the black-hole security attacks which may be conducted using the relaying SMDs? The same question can be asked for other types of security attacks such as grey-hole attacks.
- How should we deal with the selfish SMDs which do not cooperate with other SMDs to relay the offloading requests, but asks other SMDs to offload their requests?
- How to avoid hotspot problem in the D2D offloading pattern?
- How much battery power can be dedicated to relaying the offloading requests?
- How to deal with SMDs' which try to conduct DDoS attacks by launching a flood of invalid offloading requests?
- How to deal with SMDs' heterogeneity in D2D communications?

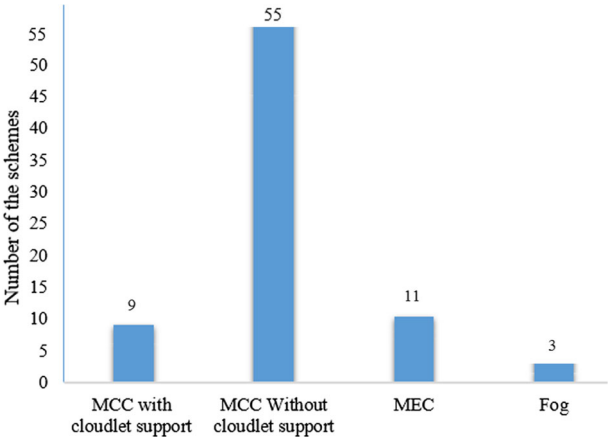


Fig. 27 Stochastic schemes environments

- Mobility pattern is another important issue in the D2D offloading which has a direct effect on the performance and reliability of the offloading process.
- How the privacy of the D2D offloading schemes can be provided? The intermediate and relay nodes can capture offloading traffic for themselves and analyze it.
- Most of the offloading schemes which have supported D2D communications, only support one-hop communications and in the next studies, multihop D2D communications based offloading can be practiced.

Figure 27 indicates the environments in which the stochastic offloading schemes are designed for them. As shown in this figure, most of the schemes are designed for MCC and in these schemes, most of them do not support cloudlets in their architecture. Consequently, using cloudlets in the MCC environment and 3-tier offloading architectures should be analyzed and discussed further in future investigations. In addition, the least stochastic offloading solutions are designed for fog computing environments. As Internet-of-Things (IoT) gains popularity, research on fog computing which is the back-end of the IoT systems seems to be necessary. In general, cloudlets, MEC, and fog computing provide 3-tier offloading architecture. However, as depicted in Fig. 27, more researches have done on the 2-tier offloading methods in which SMDs are in direct contact with the cloud computing servers. Figure 28 indicates the support for the delayed-offloading in the investigated schemes. As shown in this Figure, fewer offloading schemes are designed to support delayed-offloading; however, regarding the importance of the delayed offloading in saving SMDs' power and improving their battery lifetime, this feature should be further focused and explored in the future studies.

Figure 29 depicts the percentage of secure offloading schemes. As shown in the figure, very few schemes have taken security issues into account and in the following studies, secure offloading should be focused further. Various security solutions are proposed for MCC in the literature [101, 102]. From the security point of view, both offloading and local executions are vulnerable to a variety of security attacks and even offloading cases can incur security risks on the offloading destinations.

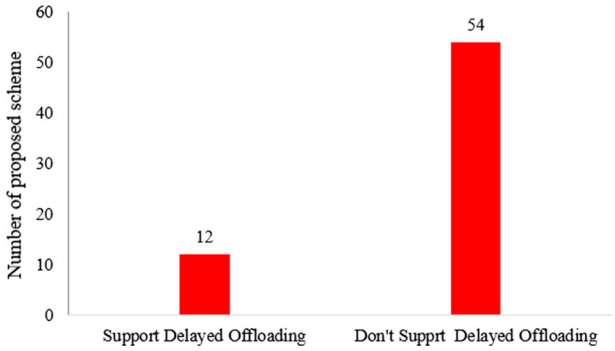


Fig. 28 Support for delayed-offloading

Fig. 29 Percentage of secure offloading schemes

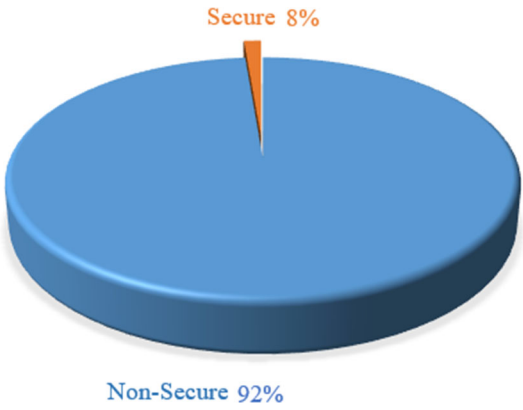


Figure 30 exhibits the number of secure and insecure stochastic offloading schemes studied in the previous section. As shown in this chart, secure schemes are only provided for the cloudlet and MCC environments and no secure stochastic offloading schemes are presented for the Fog computing and MEC environments. In the following, security threats in various steps of the offloading process are illuminated:

- *Computation/Application offloading*
  - Data transfer between SMD and the offloading destination: Application can be modified during this offloading step. Although this problem can be detected in the server-side by integrity checking tools, attackers may use this as a mechanism for DoS attacks on the offloading servers.
  - Remote execution: Offloading on the untrusted and unreliable destination, endangers the privacy of the achieved results and can impose long delays.
  - Data transfer among servers: In the case of which an offloading destination server sends the application or part of it to another remote site, privacy and integrity of the application and its data can be endangered. Thus, using secure communication links can be considered to mitigate security risks.

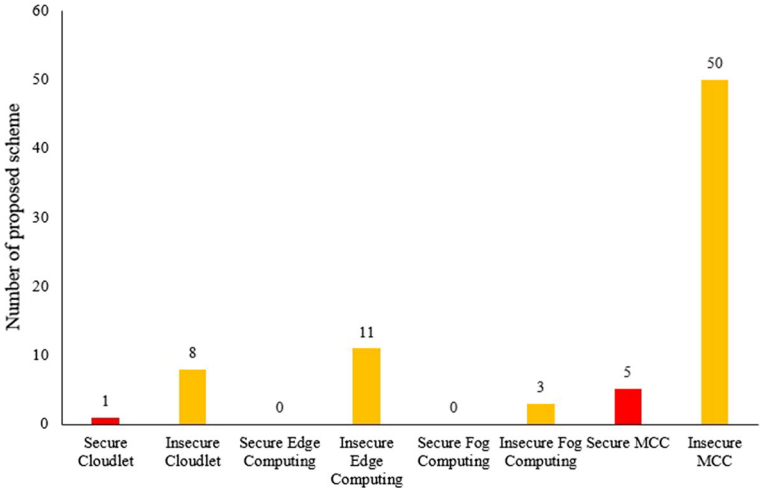


Fig. 30 Secure and insecure stochastic offloading schemes

- *Code offloading*
  - Data transfer between SMD and the offloading destination: Code can be modified during the offloading to produce results that are good for the attacker.
  - Remote execution: Remote code execution is a serious issue and it can be used to launch various attacks against the offloading destination. Thus, the required security considerations, such as execution in the sandbox or other container environments should be taken into account.
  - Data transfer among servers: Attackers may use code offloading to launch attacks in which the offloading destination be not able to handle its workload and send this load to other servers and sites, leading to propagation of attack.
- *Data offloading*
  - Local: By keeping data locally, the risk of losing whole data in various unexpected accidents should be taken into account. Also, when an SMD is stolen, its private data may be accessible to the attacker. Although this problem can be mitigated by encrypting the critical data, this process can be useful only for small data and for large data, it can incur long and tedious delays. Thus, offloading data on the remote trusted repository can improve its security and availability.
  - Data transfer between SMD and the offloading destination: Data can be modified in this step to produce results that are desired by the attacker. Using secure links can be useful in preventing such attacks, but when the SMD is affected by the hidden malware installed as part of a typical mobile application, the secure links cannot guarantee data privacy.
  - Remote storage: By transferring data to the remote servers, the privacy and integrity of data can be at risk during the data transmission phases. In addition, data repositories themselves are not immune to the security attacks and unauthorized access from CSPs.

- Data transfer among servers: In this case, to prevent security attacks, secure links should be used, however, the security of offloading source and destination should be taken into account which in the case of mobile cloudlets and mobile edge servers, ensuring this can be arduous.

6.1 Open research directions

In the future researches and studies, the following issues can be considered to further improve the stochastic offloading context:

- Regarding the importance of the power saving in the SMDs and server-side systems, efficient power management methods should be deployed to increase the battery life of the SMDs and reduce the energy costs of the MCC, cloudlets, and other offloading destinations. In this context, dynamic power management techniques such as DVFS can be further investigated because as outlined before, only a few DVFS-based stochastic offloading solutions are presented in the literature.
- The other possible direction for future researches is increasing the security of approaches designed for using direct communication with the other SMDs. For example, a node which assists offloading may be a malicious node and may conduct attacks such as a black hole.
- In addition, because of user mobility or other reasons, SMDs and wireless links sometimes may become unavailable and as a result, failure may happen during the offloading process. Thus, to conduct reliable offloading, factors such as SMDs mobility pattern, wireless link availability, and intermittently accessible offloading servers should be further considered in future studies.
- Workload oscillation is another challenging research issue.
- Variation in the processing capabilities of the offloading destinations which may be under a distributed denial of service (DDoS) attacks has not investigated and should be addressed in the future.
- 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 examined literature and should be focused further.
- Almost all of the studied schemes are greedy offloading solutions aimed at providing an optimal set of offloading decisions to optimize some factors. In this context, there is a lack of offloading schemes to make decisions globally by considering the states of the offloading destination and states of the other SMDs.

7 Conclusion and open research issues

The application of smart mobile devices or SMDs is growing in everyday life in recent years and various interesting applications ranging from online gaming to the e-healthcare applications are designed for their platform. This incurs a heavy demand for more processing power, memory, storage, and network bandwidth to enable application execution and satisfy the SMDs' users. Offloading is an interesting paradigm that enables the relocation of applications, codes, and data from resource-constrained

SMDs to the resource-rich near-by or remote servers. By growing coverage of WiFi technology, SMDs can benefit from free bandwidth of the WLAN to alleviate data sent over the cellular networks and reduce the costs of using them. However, regarding various affecting factors, offloading is considered as an NP-hard problem and it should be decided to offload what, when, where, and how.

Numerous offloading approaches are proposed to handle offloading problems in various recently proposed technologies such as edge computing, fog computing, and MCC. This paper focuses on the recently proposed state of the art offloading schemes which have applied stochastic models to aid SMDs in making better or optimal offloading decisions dynamically. It first discusses the advantages of the offloading and then illustrates various features of the offloading methods and describes the stochastic models applied in the investigated offloading approaches. Then, it classifies the studied offloading solutions based on their utilized stochastic models and provides an extensive literature review on them, while illuminating their main contributions, objectives, and any possible limitations. At last, a comparison of the evaluation factors, the simulators along with various features of the investigated stochastic offloading solutions such as their support for dynamic power management, delayed offloading, heterogeneity, and also offloading type are outlined to provide an in-depth insight into these schemes.

References

1. Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl 1(1):7–18
2. Chen X et al (2017) Exploiting massive D2D collaboration for energy-efficient mobile edge computing. IEEE Wirel Commun 24(4):64–71
3. Masdari M et al (2017) A survey of PSO-based scheduling algorithms in cloud computing. J Netw Syst Manag 25(1):122–158
4. Masdari M, Zangakani M (2019) Green cloud computing using proactive virtual machine placement: challenges and issues. J Grid Comput. https://doi.org/10.1007/s10723-019-09489-9
5. Zhang K et al (2016) Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE Access 4:5896–5907
6. Shiraz M et al (2015) A study on the critical analysis of computational offloading frameworks for mobile cloud computing. J Netw Comput Appl 47:47–60
7. Kwon Y et al (2016) Precise execution offloading for applications with dynamic behavior in mobile cloud computing. Pervasive Mob Comput 27:58–74
8. Masdari M, Jalali M (2016) A survey and taxonomy of DoS attacks in cloud computing. Secur Commun Netw 9(16):3724–3751
9. Masdari M, Nabavi SS, Ahmadi V (2016) An overview of virtual machine placement schemes in cloud computing. J Netw Comput Appl 66:106–127
10. Masdari M et al (2016) Towards workflow scheduling in cloud computing: a comprehensive analysis. J Netw Comput Appl 66:64–82
11. Shiraz M et al (2015) Energy efficient computational offloading framework for mobile cloud computing. J Grid Comput 13(1):1–18
12. Lordan F, Badia RM (2017) Compss-mobile: parallel programming for mobile cloud computing. J Grid Comput 15(3):357–378
13. Panigrahi CR, Sarkar JL, Pati B (2018) Transmission in mobile cloudlet systems with intermittent connectivity in emergency areas. Digit Commun Netw 4(1):69–75
14. Ghobaei-Arani M, Souri A, Rahmanian AA (2020) Resource management approaches in fog computing: a comprehensive review. J Grid Comput 18:1–42. https://doi.org/10.1007/s10723-019-09491-1
15. Douc R et al (2018) Markov chains. Springer, Berlin

16. Hu M et al (2019) Quantifying the influence of intermittent connectivity on mobile edge computing. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2019.2926702
17. Amiri M, Mohammad-Khanli L (2017) Survey on prediction models of applications for resources provisioning in cloud. J Netw Comput Appl 82:93–113
18. Li W et al (2019) Opportunistic computing offloading in edge clouds. J Parallel Distrib Comput 123:69–76
19. Yu F, Chen H, Xu J (2018) DMPO: dynamic mobility-aware partial offloading in mobile edge computing. Future Gener Comput Syst 89:722–735
20. Meng T et al (2018) A secure and cost-efficient offloading policy for mobile cloud computing against timing attacks. Pervasive Mob Comput 45:4–18
21. Nădăban S, Dzitac S, Dzitac I (2016) Fuzzy TOPSIS: a general view. Procedia Comput Sci 91:823–831
22. Zhang J et al (2018) Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks. IEEE Internet Things J 5(4):2633–2645
23. Shuja J et al (2017) Case of ARM emulation optimization for offloading mechanisms in mobile cloud computing. Future Gener Comput Syst 76:407–417
24. Tao X et al (2017) Performance guaranteed computation offloading for mobile-edge cloud computing. IEEE Wirel Commun Lett 6(6):774–777
25. Zhang Y, Niyato D, Wang P (2015) Offloading in mobile cloudlet systems with intermittent connectivity. IEEE Trans Mob Comput 14(12):2516–2529
26. Gu F et al (2018) Partitioning and offloading in smart mobile devices for mobile cloud computing: state of the art and future directions. J Netw Comput Appl 119:83–96
27. Pan Y et al (2017) On consideration of content preference and sharing willingness in D2D assisted offloading. IEEE J Sel Areas Commun 35(4):978–993
28. Flores H et al (2015) Mobile code offloading: from concept to practice and beyond. IEEE Commun Mag 53(3):80–88
29. Masdari M (2017) Markov chain-based evaluation of the certificate status validations in hybrid MANETs. J Netw Comput Appl 80:79–89
30. Akherfi K, Gerndt M, Harroud H (2018) Mobile cloud computing for computation offloading: issues and challenges. Appl Comput Inform 14(1):1–16
31. Pu L et al (2016) D2D fogging: an energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. IEEE J Sel Areas Commun 34(12):3887–3901
32. Chen X et al (2015) Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans Netw 24(5):2795–2808
33. Kuang Z et al (2018) A quick-response framework for multi-user computation offloading in mobile cloud computing. Future Gener Comput Syst 81:166–176
34. Barrett E, Howley E, Duggan J (2013) Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. Concurr Comput Pract Exp 25(12):1656–1674
35. Goudarzi M, Zamani M, Haghighat AT (2017) A fast hybrid multi-site computation offloading for mobile cloud computing. J Netw Comput Appl 80:219–231
36. Oo TZ et al (2016) Traffic offloading via Markov approximation in heterogeneous cellular networks. In: NOMS 2016–2016 IEEE/IFIP network operations and management symposium
37. Oo TZ et al (2017) Offloading in HetNet: a coordination of interference mitigation, user association, and resource allocation. IEEE Trans Mob Comput 16(8):2276–2291
38. Zhang S et al (2016) Energy-aware traffic offloading for green heterogeneous networks. IEEE J Sel Areas Commun 34(5):1116–1129
39. Xiao L et al (2016) A mobile offloading game against smart attacks. IEEE Access 4:2281–2291
40. Li X et al (2015) Light-weight performance analysis of Wi-Fi offload using mean-field approximation. In 2015 21st Asia-Pacific conference on communications (APCC)
41. Meng T, Wolter K, Wang Q (2015) Security and performance tradeoff analysis of mobile offloading systems under timing attacks. Springer, Cham
42. Wu H, Wolter K (2018) Stochastic analysis of delayed mobile offloading in heterogeneous networks. IEEE Trans Mob Comput 17(2):461–474
43. Zhang W, Wen Y, Wu DO (2013) Energy-efficient scheduling policy for collaborative execution in mobile cloud computing. In: 2013 proceedings IEEE INFOCOM
44. Wu H, Wolter K (2016) Analysis of the energy-performance tradeoff for delayed mobile offloading. In: Proceedings of the 9th EAI international conference on performance evaluation methodologies

and tools. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)
45. Tang L, Chen X, He S (2016) When social network meets mobile cloud: a social group utility approach for optimizing computation offloading in Cloudlet. IEEE Access 4:5868–5879
46. Wu H, Knottenbelt W, Wolter K (2015) Analysis of the energy-response time tradeoff for mobile cloud offloading using combined metrics. In 2015 27th international teletraffic congress
47. Roostaei R, Movahedi Z (2016) Mobility and context-aware offloading in mobile cloud computing. In: 2016 International IEEE conferences on ubiquitous intelligence and computing, advanced and trusted computing, scalable computing and communications, cloud and Big data computing, internet of people, and smart world congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)
48. Mehmeti F, Spyropoulos T (2017) Performance analysis of mobile data offloading in heterogeneous networks. IEEE Trans Mob Comput 16(2):482–497
49. Zhang X, Cao Y (2018) Mobile data offloading efficiency: a stochastic analytical view. In: 2018 IEEE international conference on communications workshops (ICC Workshops)
50. Berg F, Dürr F, Rothermel K (2014) Optimal predictive code offloading. In: Proceedings of the 11th international conference on mobile and ubiquitous systems: computing, networking and services. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)
51. Wu H, Sun Y, Wolter K (2015) Analysis of the energy-response time tradeoff for delayed mobile cloud offloading. ACM SIGMETRICS Perform Eval Rev 43(2):33–35
52. Ko S et al (2017) Energy efficient mobile computation offloading via online prefetching. In: 2017 IEEE international conference on communications (ICC)
53. Zhang W et al (2013) Energy-optimal mobile cloud computing under stochastic wireless channel. IEEE Trans Wirel Commun 12(9):4569–4581
54. Kim J et al (2013) Placement of WiFi access points for efficient WiFi offloading in an overlay network. In: 2013 IEEE 24th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)
55. Kim S et al (2016) Prediction-based personalized offloading of cellular traffic through WiFi networks. In: 2016 IEEE international conference on pervasive computing and communications (PerCom)
56. Gao W et al (2014) On exploiting dynamic execution patterns for workload offloading in mobile cloud applications. In: 2014 IEEE 22nd international conference on network protocols
57. Tong L, Gao W (2016) Application-aware traffic scheduling for workload offloading in mobile clouds. In: IEEE INFOCOM 2016—the 35th annual IEEE international conference on computer communications
58. Meng T, Wang Q, Wolter K (2015) Model-based quantitative security analysis of mobile offloading systems under timing attacks. Springer, Cham
59. Yamamoto H et al (2014) Modeling of dynamic trend of latency variations on mobile network using markov regime switching. In: 2014 IEEE 38th international computer software and applications conference workshops
60. Yu P et al (2015) Energy harvesting personal cells—traffic offloading and network throughput. In: 2015 IEEE international conference on communications (ICC)
61. Cheng N et al (2016) Opportunistic WiFi offloading in vehicular environment: a game-theory approach. IEEE Trans Intel Transp Syst 17(7):1944–1955
62. Wei Y et al (2016) The offloading model for green base stations in hybrid energy networks with multiple objectives. Int J Commun Syst 29(11):1805–1816
63. Xu J, Chen L, Ren S (2017) Online learning for offloading and autoscaling in energy harvesting mobile edge computing. IEEE Trans Cognit Commun Netw 3(3):361–373
64. Alam MGR et al (2019) Autonomic computation offloading in mobile edge for IoT applications. Future Gener Comput Syst 90:149–157
65. Alasmari KR, Green RC, Alam M (2018) Mobile edge offloading using markov decision processes. In: International conference on edge computing. Springer
66. Zhang C et al (2017) Cost-and energy-aware multi-flow mobile data offloading under time dependent pricing. In: 2017 13th international conference on network and service management (CNSM). IEEE
67. Le DV, Tham C (2018) A deep reinforcement learning based offloading scheme in ad-hoc mobile clouds. In: IEEE INFOCOM 2018—IEEE conference on computer communications workshops (INFOCOM WKSHPS)
68. Wang W et al (2017) Edge caching at base stations with device-to-device offloading. IEEE Access 5:6399–6410

69. Labidi W, Sarkiss M, Kamoun M (2015) Energy-optimal resource scheduling and computation offloading in small cell networks. In: 2015 22nd international conference on telecommunications (ICT)
70. Fricker C et al (2016) Analysis of an offloading scheme for data centers in the framework of fog computing. ACM Trans Model Perform Eval Comput Syst (TOMPECS) 1(4):16
71. Zannat H, Hossain MS (2016) A hybrid framework using Markov decision process for mobile code offloading. In; 2016 19th international conference on computer and information technology (ICCIT)
72. Terefe MB et al (2016) Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing. Pervasive Mob Comput 27:75–89
73. Liu D, Khoukhi L, Hafid A (2017) Data offloading in mobile cloud computing: a Markov decision process approach. In: 2017 IEEE international conference on communications (ICC)
74. Hyytiä E, Spyropoulos T, Ott J (2015) Offload (only) the right jobs: robust offloading using the Markov decision processes. In; 2015 IEEE 16th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)
75. Truong-Huu T, Tham C, Niyato D (2014) To offload or to wait: an opportunistic offloading algorithm for parallel tasks in a mobile cloud. In: 2014 IEEE 6th international conference on cloud computing technology and science
76. Zhang C et al (2016) A reinforcement learning approach for cost- and energy-aware mobile data offloading. In; 2016 18th Asia-Pacific network operations and management symposium (APNOMS)
77. Liu B et al (2018) Congestion-optimal WIFI offloading with user mobility management in smart communications. Wirel Commun Mob Comput. https://doi.org/10.1155/2018/9297536
78. Kim Y et al (2016) Multi-flow rate control in delayed Wi-Fi offloading systems. In; 2016 international conference on information networking (ICOIN)
79. Komnios I, Tsapeli F, Gorinsky S (2015) Cost-effective multi-mode offloading with peer-assisted communications. Ad Hoc Netw 25:370–382
80. Liu B, Zhu Q, Zhu H (2017) CAWO: congestion-aware WiFi offloading for 5G heterogeneous wireless network. In: 2017 13th international wireless communications and mobile computing conference (IWCMC)
81. Le DV, Tham C (2017) An optimization-based approach to offloading in ad-hoc mobile clouds. In: GLOBECOM 2017—2017 IEEE global communications conference
82. Ranadheera S, Maghsudi S, Hossain E (2017) Mobile edge computation offloading using game theory and reinforcement learning. arXiv preprint arXiv:1711.09012
83. Mao Y, Zhang J, Letaief KB (2016) Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J Sel Areas Commun 34(12):3590–3605
84. Hyytiä E, Spyropoulos T, Ott J (2013) Optimizing offloading strategies in mobile cloud computing. In: Cryptanalyst
85. Wu H, Wolter K (2014) Tradeoff analysis for mobile cloud offloading based on an additive energy-performance metric. In: Proceedings of the 8th international conference on performance evaluation methodologies and tools. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)
86. Ko H, Lee J, Pack S (2018) Spatial and temporal computation offloading decision algorithm in edge cloud-enabled heterogeneous networks. IEEE Access 6:18920–18932
87. Chen X et al (2015) A learning approach for traffic offloading in stochastic heterogeneous cellular networks. In: 2015 IEEE international conference on communications (ICC). IEEE
88. He X et al (2017) Privacy-aware offloading in mobile-edge computing. In; GLOBECOM 2017—2017 IEEE global communications conference
89. Liu J et al (2016) Delay-optimal computation task scheduling for mobile-edge computing systems. In; 2016 IEEE international symposium on information theory (ISIT)
90. Carvalho GHS et al (2017) *A* Semi-Markov decision model-based brokering mechanism for mobile cloud market. In 2017 IEEE international conference on communications (ICC)
91. Wang Z, Zhong Z, Ni M (2017) A semi-Markov decision process-based computation offloading strategy in vehicular networks. In: 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)
92. Zhang D, Yeo CK (2012) Optimal handing-back point in mobile data offloading. In: 2012 IEEE vehicular networking conference (VNC)

93. 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. IEEE Press
94. Zhuo X et al (2014) An incentive framework for cellular traffic offloading. IEEE Trans Mob Comput 13(3):541–555
95. Liu Y, Lee MJ, Zheng Y (2016) Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system. IEEE Trans Mob Comput 15(10):2398–2410
96. Hoang DT, Niyato D, Wang P (2012) Optimal admission control policy for mobile cloud computing hotspot with cloudlet. In: 2012 IEEE wireless communications and networking conference (WCNC)
97. Wang Z, Zhong Z, Ni M (2018) Application-aware offloading policy using SMDP in vehicular fog computing systems. In: 2018 IEEE international conference on communications workshops (ICC Workshops)
98. Ramakrishnan AK et al (2012) Federated mobile activity recognition using a smart service adapter for cloud offloading. Springer, Dordrecht
99. Wang X, Xu W, Jin Z (2017) A hidden Markov model based dynamic scheduling approach for mobile cloud telemonitoring. In: 2017 IEEE EMBS international conference on biomedical and health informatics (BHI). IEEE
100. Eom H et al (2013) Machine learning-based runtime scheduler for mobile offloading framework. In; Proceedings of the 2013 IEEE/ACM 6th international conference on utility and cloud computing. IEEE Computer Society
101. Lordan F, Jensen J, Badia RM (2018) Towards mobile cloud computing with single sign-on access. J Grid Comput 16(4):627–646
102. Kashyap R, Vidyarthi DP (2013) Security driven scheduling model for computational grid using NSGA-II. J Grid Comput 11(4):721–734