A comprehensive review of task offloading in edge computing

Marcos P. Konzen[+⋆], Arthur F. Lorenzon[†], Marcelo C. Luizelli[†], Fábio D. Rossi[+], Júlio C. B. Mattos[⋆]
[+]Federal Institute Farroupilha (IFFAR), Brazil
[†]Federal University of Pampa (UNIPAMPA), Brazil
[⋆]Federal University of Pelotas (UFPEL), Brazil
mkonzen@inf.ufpel.edu.br

***Abstract*—Nowadays, mobile applications are demanding compute-intensive use, in addition to the need for lower latency and lower computational costs. Thus, researchers are proposing to bring the computation of these applications closer to the users by offloading these applications to the Edge. In this work, we carried out a comprehensive literature review with the primary objective of investigating the offloading strategies used in the Edge Computing scenario, which restrictions are considered, and the security aspects considered by the strategies. From the selected works, we describe the main optimization objectives of the strategies, which models and algorithms were implemented, which computational constraints were considered, which types of applications, and the security requirements. Finally, we discussed some opportunities and open challenges.**

***Index Terms*—Edge computing, offloading strategies, task offloading.**

I. Introduction

In recent years, we have observed an increase and popularization in the use of smart mobile devices, such as smartphones, wearable devices, the Internet of Things (IoT), autonomous or unmanned vehicles, and intelligent embedded systems. It results from big data development that requires intensive resources, such as image recognition, augmented reality (AR), virtual reality (VR), semantic speech analysis, etc. These applications are latency-sensitive, require real-time processing of ever-increasing data streams, and generate high power consumption. In these cases, mobile devices alone are insufficient to run applications that require intensive use of data, as such devices have limited computational and energy resources [1].

However, due to their centralized processing feature, cloud infrastructures are far from end devices, which increases data transmission latency, delays processing responses, and can make costs unaffordable for end-users [2]. Edge Computing is an infrastructure that creates a layer of computing resources, such as the cloud, between mobile users and the cloud. One of the main advantages of edge computing is low latency, lower power consumption, and data transmission costs [3].

Thus, Computation Offloading (or simply offloading) has been proposed to use computational resources of this layer close to local devices, overcoming the limitations of these devices. The main challenges of offloading strategies are deciding which task, how much code is in the task, to which edge or cloud node the task will be placed, and in what order it will occur. These decisions are difficult, as the algorithms for an ideal strategy must consider heterogeneous resources, user requirements, task dependencies, the security requirements, increasing the complexity of the problem.

In the literature, it is possible to find works that propose different offloading strategies applied to Edge Computing. However, we observe a lack of studies that gather the main characteristics and objectives of these strategies, the algorithms used, which restrictions were considered, and what types of applications benefited. This work aims to review the literature on the offloading strategies of edge applications. The main contributions of this study are: (i) classification of the main objectives of offloading strategies at the Edge, (ii) an overview of the strategies and algorithms used to develop the offloading models of this study, (iii) classification of the main computational constraints considered in modeling offloading solutions and which security requirements are considered, (iv) and a discussion of research opportunities and challenges.

The rest of this article is organized as follows: Section II discusses Edge Computing and Computation Offloading; Section III presents the related work; Section IV states the applied methodology of this study; Section V shows the results and discussions; Section VI states some research opportunities and challenges; Section VII presents our conclusions.

II. Background

As mobile applications become more and more intense, the need for more processing and less response time becomes increasingly essential. Cloud computing has become a promising solution in offering unlimited computing resources scalable on-demand services. However, the centralization of cloud resources, the geographic distance of data centers, and the overload on communication networks ended up increasing the response time, harming real-time applications sensitive to delay [4]. To minimize this problem, Edge has a decentralized architecture based on layers, generally composed of the user devices layer, the edge layer, and the cloud layer. According to this paradigm, the edge layer resources are closer to the user devices, which can transfer the processing of their tasks and data to the edge layer, decreasing the overall computation delay and reducing the occurrence of bottlenecks in data centers [5], [2]. Tasks in edge computing can be partitioned into multiple subtasks, and each subtask can be offloaded to different edge devices for faster processing [3].

Computational offloading has been studied by several researchers and is proposed to overcome the resource limitations of mobile devices. Offloading is defined as the process of sending one or more tasks and related data to be processed in a cloud servers or edge servers. Several factors affect the processing transfer decision, including local device resources such as CPU, storage, and battery life, as well as delays in data transmission over the network and the size of transferred data [6].

Offloading strategies are based on computational, communication, and mobility constraints. In addition, several criteria are considered for the building of offloading strategies, such as objectives, the granularity of the code to be executed, if the offloading is static or dynamic, if the offloading destination nodes are predetermined, or dynamically discovered if the offloading decision will be at runtime, among others [7]. A fair offloading policy should strike the right balance between overall compute delay, data throughput, and related performance metrics.

III. Related Work

Several studies have emerged in recent years on the offloading of edge applications. In [3] the authors present the study on offloading at the edge, covering offloading scenarios, influencing factors, and offloading strategies. Offloading based on machine learning (ML) is the focus of [8]'s study. The study reviews existing studies using various ML-based approaches and discusses possible AI problems in the MEC.

In the study proposed by [9], the objective is to review the literature on offloading in vehicular environments. In addition, they propose a taxonomy that is used to classify the works. They also present the leading tools, scenarios, themes, strategies, and objectives. A review on offloading in IoT environments is presented in [10], where they researched the benefits and challenges in implementing offloading strategies in an IoT application scenario at the edge through a case study. A review about application offloading studies in the Vehicular Edge Computing (VEC) scenario is addressed in the study of [11], where the authors examine offloading strategies in the VEC environment, proposing a new taxonomy of proposed offloading approaches.

The survey presented by [12] present studies that describe the computational offloading process and the MEC structure involved. In addition, they offer some models of offloading algorithms and evaluation parameters, giving some directions for future work. In [13] the authors carried out research on the application of game theory in the development of offloading strategies and resource allocation at the edge. The authors characterized the strategies based on the goals.

A taxonomy on offloading edge applications is also proposed in [14] to classify different articles on the topic. The focus was to investigate cooperative offloading strategies between the edge and the cloud. Finally, the work [15] carries out a literature review on edge computing offloading, focusing on approaches that propose to minimize energy consumption, guarantee the quality of services (QoS) and improve the quality of experience (QoE).

Our research differs from other studies by making a broader investigation, especially looking for works that describe the strategy's objectives, the implemented algorithm, which features and constraints were considered, and which types of applications were tested and are considered security constraints. In this sense, our work can be helpful to identify some gaps that are not yet well discussed in offloading strategies.

IV. Research Design

In order to produce a good understanding of offloading strategies in the Edge Computing scenario in a complementary way to the studies already presented in the related work, this study is characterized by being a literature review that proposes a research process according to the guidelines of [16]. According to the purpose of the study, the investigation process should be motivated by a key research question [17]. The central question of this work is "What offloading strategies are used in the Edge Computing scenario? To simplify the conduct of the research, the central question was divided into the following sub-questions and objectives:

Q1: What are the main offloading strategies involving Edge Computing? The objective is to investigate application offloading techniques in Edge Computing environments.

Q2: What are the objectives of these strategies? The objective is to investigate which performance metrics the offloading strategies intend to optimize.

Q3: What models and algorithms were used in offloading strategies? The objective is to identify which models served as the basis for constructing the algorithms used in the works.

Q4: What computational constraints are considered in offloading strategies? The objective is to investigate which constraints and computational resources are considered in offloading strategies.

Q5: What types of applications are impacted by offloading strategies? The objective is to investigate which applications are tested in the offloading strategies studied.

Q6: What security aspects are considered in offloading strategies? The objective is to verify if the studies consider security requirements in the proposed strategies.

We used PICO systematic model (Population, Intervention, Comparison, and Outcomes) to identify key terms from the research questions according [16]. The basic string search is then defined by concatenating the strings generated by each part of the PICO model using boolean AND and OR keys. We identify synonyms of the keywords so that the search can return articles that use similar terms and use wildcard characters. We included the term "Fog Computing" in the string due to the existence of some authors in the literature conceptualizing Edge Computing as Fog Computing and vice-versa. The following basic search string was used to build and obtain the initial set of primary studies:

```
("offload*" AND ("edge computing" OR "fog
computing" OR "mobile edge computing" OR
"mec"))
```

We chose the bases IEEEExplore, ACM Digital Library, and Scopus were chosen as digital research bases for this study, according [16] and [17]. Refining the search results is critical to ensuring the quality of the results. The use of inclusion criteria (IC) establishes characteristics that a study must contain to be relevant in the research context. Likewise, exclusion criteria (EC) establish characteristics to exclude studies that do not cover the topic of interest. We used the following inclusion criteria: (IC1) articles published in the last 5 years, (IC2) primary articles dealing with offloading in the Edge context, (IC3) articles that describe the estratagy and the algorithm used and (IC4) articles that demonstrate results od experiments of the techniques. Exclusion criteria: (EC1) duplicate articles, (EC2) inaccessible articles, (EC3) articles that do not detail the offloading process or is not applied to the Edge context, (EC4) articles that do not describe the objective of the strategy and (EC5) articles that do not describe the modeling of the considered computational resources.

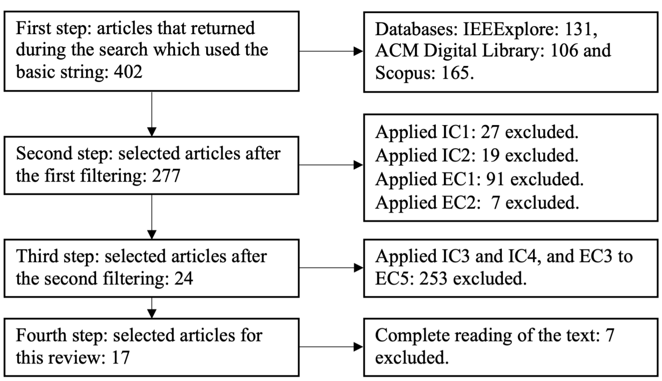

Fig. 1. Research steps.

According to Fig. 1, the investigation was carried out in four steps: In the first step, the search was carried out from the basic string of the search in the selected data sources. A total of 402 articles were returned in the first step. The second step applied inclusion criteria IC1 and IC2, and exclusion criteria EC1 and EC2 in the initial search. The search tools allow using search filters, helping to apply IC1, EC1, and EC2 criteria. The IC2 inclusion criterion was applied by reviewing the text's title, abstract and diagonal reading. After filtering in step 2, it resulted in 277 articles. In the third step, the selected works were analyzed in more detail through the complete reading of the text. For this, inclusion criteria IC3 and IC4, and exclusion criteria EC3 to EC5 were applied. After step 3, 24 articles were selected for the next step. Finally, in the fourth step, a complete reading of the articles was carried out and 17 were selected for this review according to the information necessary to achieve the research objectives. Seven articles were excluded at this stage because they did not present consistent results for this study. The following section presents the results and discussions of the study.

V. Results and Discussion

This study focuses on investigating the main characteristics of computing offloading strategies used in the Edge Computing scenario, such as offloading objectives, the strategy used, the algorithm implemented, which restrictions were considered, what types of applications were tested in the studies and what security constraints were considered.

Offloading Objectives. Minimizing task execution latency is the main objective of the vast majority of studies. Minimizing computing costs is cited by 6 studies, followed by 5 studies that also try to minimize energy consumption, and one work aims to optimize the time of the offloading process. The studies of [18] and [19] focus only on minimizing energy consumption, whose strategy is modeled in such a way that the offloading decision is based on the restrictions of processing energy and data transmission. The models of [6], [20], [21], [22], [4], [7] and [23] aim to minimize only the computing latency, prioritizing the offloading of applications that are sensitive to latency, such as real-time applications. In these strategies, the main constraint is communication latency.

Some studies use multi-objectives to decide offloading tasks, and generally, they look for a better trade-off between different objectives. The studies of [24], [25] and [26] aim to minimize latency and energy consumption. In [27], the proposal is to minimize energy consumption and computing costs. Minimizing latency and costs are proposed in [28]. [29] propose an offloading strategy in which the user can choose to minimize latency, costs, or energy consumption. In the work of [5], in addition to minimizing latency, it focuses on an algorithm that minimizes the offloading decision time.

Strategies and algorithms. To achieve the goals of offloading, different strategies and algorithms are proposed in the articles studied. Most of them apply models and algorithms known in the literature and adapt them according to constraints and objectives. Strategies based on heuristic algorithms are used in works on [6], [29] and [2], and meta-heuristics on [18]. In order to improve task execution time, [6] uses compute reuse strategy to improve task execution time, and [24] uses data caching strategy. [27] uses a three-layer architecture strategy with sub-algorithms that solve a multi-objective optimization problem. The collaborative offloading strategy based on the Stackelberg game is used in the work of [28]. Game theory is also used in the [26] strategy. Strategies based on Deep Learning are proposed by [5], [22], [19]. In [21], and [2] algorithms based on Artificial Intelligence are developed to increase the granularity of application offloading.

Strategies based on Data Streaming Processing (DSP) are presented in works by [23], [7] and [25]. A characteristic of these strategies is that proposed algorithms extend the functionality of frameworks for data stream processing to choose the best node for processing according to the defined constraints. In the work of [20], the method used is the A Library for Support Vector Machines (LibSVM) which identifies the different types of applications, dynamically alternating execution in the cloud or at the edge through a dynamic switching algorithm. Finally, a mathematical model using Integer Linear Programming (ILP) and Aspect-Oriented Programming is used by [4] to verify the feasibility of edge devices in performing user mobile application tasks.

Computing constraints. It is possible to verify that even strategies with common goals use different models and algorithms to decide what, when, and where to offload applications. One of the issues that impact the choice and modeling of the problem is the computational constraints related to the computational resources available at the edge and the constraints imposed by the applications. For example, applications that require intensive data processing impose restrictions on the processing capacity of the devices, which is considered in 15 strategies studied.

Latency-sensitive applications generally impose communication latency restrictions, addressed in 4 studies, and are impacted by the transmission rate and bandwidth, considered in 5 and 8 studies, respectively. Strategies that aim to minimize energy consumption need to consider devices' energy consumption and battery life. There is a need to improve the trade-off between performance and energy savings, considered in 8 strategies. The computational monetary cost, such as the cost of cloud services or transmission cost, is a constraint considered in the strategies that aim to reduce the cost for the end-user, addressed in 7 works, as well as the complexity of the tasks and the size of the data to be processed. Node storage capacity is also considered by 4 works, as they influence the total latency of task execution and data caching. The amount of node memory and processor capacity is also mentioned by 3 works, which can influence the execution of tasks with a higher processing load.

Restrictions related to nodes, tasks, and data security are cited in only 1 work. It means that these constraints need to be better explored in the development of offloading strategies since the number of tasks and the number of nodes can directly influence the model's performance. Likewise, given that tasks and data are migrated to other nodes over data networks, security constraints are crucial to protect data and applications. The number of constraints considered can create smarter strategies. However, the more restrictions are modeled, the greater the complexity of the algorithms, which can impact the performance of the algorithms, compromising the objectives of the offloading strategies.

Types of applications. Different types of applications may impose different computational constraints and, therefore, may perform better or worse depending on the offloading strategy used. Thus, one of the objectives of this study was to identify what type of applications were used to evaluate the proposed offloading strategies. Most of the studies carried out tests of the algorithms through simulated experiments in mathematical software or simulators of Edge environments. Most of the studies do not specify the type of application considered in the simulations or specify only as generic tasks, which indicates that few strategies are developed focusing on a certain type of application.

The proposals of [29] and [4] focus on mobile applications without specifying the type of applications. The strategies of [20], [23], [7] and [25] are aimed at optimizing Big Data and Data Streaming Processing (DSP) applications. In [21] and [26] the focus is on improving the performance of real-time image processing applications. Offloading of autonomous vehicle applications is proposed by [24], and also proposes an offloading strategy for industrial applications in [27].

As many works do not specify or test only one type of application, it is still a challenge to determine the best offloading solution based on the type of application. An opportunity for future work is to carry out experiments with different types of applications, with different restrictions on certain strategies, to assess the models' generality or specificity.

Security aspects. Offloading strategies mainly aim to migrate tasks and data to be processed in other network nodes or the cloud. In this sense, security restrictions are crucial to protect data and applications. As we discussed earlier, only few studies consider security restrictions in implementing the offloading strategy. For example, the study by [18] proposes an algorithm that classifies computing nodes into security levels. The higher the level, the more secure the compute node, and node classification takes into account both physical and software security levels. The strategy proposed in [5] predicts that the resources enabled for the cluster of cloud and edge resources, as well as the users of these resources, must go through an authentication step. After authentication, they will be able to participate in the offloading process. However, the study does not detail this authentication process.

In [7], the authors consider using containers to ensure the security and privacy of applications. Although the strategy uses a mechanism that scans ports and web servers in search of computational nodes, no other security mechanism against malicious nodes is discussed. The offloading model proposed in [23] only considers the security features of the NiFi middleware – the platform used in work. In addition, nodes that wish to be part of the edge cluster must go through an authentication process controlled by a web server in order to be able to interact with other nodes reliably. In the work of [21], no restrictions or security mechanisms are addressed. However, the authors mention that it is still a challenge to deal with the privacy and security leakage of applications in the offloading process, as well as protection tasks so that unauthorized people or applications have access, guaranteeing the sharing of a task in a safe way, and reliable across local/cloud/edge devices, and how to store job data securely.

Most studies predict the use of heterogeneous resources from cloud, edge, and end devices in a distributed way. From different domains, for example, heterogeneous compute nodes from different locations and unreliable data links [29], edge nodes scattered and which must be within reach coverage of end devices [19], nodes edge devices controlled by more than one service provider [6] [22], and end-user devices such as laptops, tablets, and routers [4]. The offloading model proposed by [20] categorizes applications into 4 classes, where tasks are transferred to nodes according to their geographic locations. Despite not mentioning security strategy or constraint in the offloading process, these works describe which security requirements are essential and should be considered in future work. Furthermore, the models proposed in [2], [24], [27], [28], [26] and [25] do not consider any security requirements.

As we can see, a minority of studies consider security requirements in their offloading strategies. However, some works mention the importance of considering security constraints in their future work. It leads us to believe that most strategies consider the premise that the resources present in the cluster are safe and reliable and belong to a controlled environment. However, in offloading models that consider heterogeneous resources and belong to different entities that do not have prior trust with each other, it is still a challenge to implement adequate security restrictions.

VI. Opportunities and Open Challenges

As one can see, several offloading solutions and strategies are proposed in the literature. Mathematical models, heuristic algorithms, Game Theory, and Artificial Intelligence are examples of solutions offered for offloading tasks to the Edge. Reducing latency, costs, and energy consumption are the main objectives of the strategies studied.

One of the objectives of this study was to identify which types of applications were tested in offloading solutions. However, many articles do not describe or specify the application used in the experiments or compare the performance differences only in scenarios with or without offload. Given that, we understand that there is still place for research that addresses experiments of different types of applications in the proposed offloading strategies to clarify better which strategies are most suitable for specific applications. More recent studies suggest offloading strategies to improve the performance of DSP applications due to data streams being generated by all types of sources, in different formats and volumes, continuously and endlessly, and having to be processed in real-time, taking into account that most solutions change or add functionality to the structure of existing commercial frameworks.

Data and application security do not appear as an objective of any strategies. Regarding the restrictions and computational resources considered in modeling the proposals studied, the security of the applications appears in only one of the studies. Some studies only consider security features of third-party solutions, but that is not part of the offloading strategy itself. However, some works consider data and application security when moving processing to another device or fundamental architecture. This leads us to believe that most works still consider the computing resources available in the cloud or edge to be safe and controlled. More recent studies suggest that edge environments are increasingly distributed, especially computing resources belonging to different owners. Thus, it is still challenging to model strategies that seek a better trade-off between security and performance.

With the increasing escalation of data theft and leakage, studies of strategies focusing on data security, especially in the big data and DSP scenario, still have research spaces that can be better explored in the Edge Computing environment. In this context, some studies point to the use of blockchain to ensure the security and privacy of transactions in the context of Edge Computing. For example, the use of smart contracts can ensure that only devices or entities that comply with the requirements set out in the template can securely process tasks [30], [31], [32] and [33]. Still in the context of blockchain, reputation mechanisms and consensus mechanisms can be used to minimize, or solve, trust problems between edge nodes [34], [35], [36], [37] and [38]. The use of these mechanisms is based on the premise that nodes that meet the requirements of the model increase their reputation over time, while nodes with low reputations are excluded from the model, with the verification and agreement of the entire chain. This would be very advantageous in offloading strategies that use nodes from different service providers and that do not have prior trust with each other.

However, running these algorithms on devices with limited processing power, such as embedded devices, is still challenging. In future work, we intend to extend our research to evaluate better the performance of these algorithms in an infrastructure composed of devices with limited computing resources.

VII. Conclusion

We presented a literature review about application offloading strategies at the Edge. This work presents a current overview and research gaps in edge computing and offers some insights that serve as a basis for future research in the area, which we could not find in other reviews.

We discussed the main characteristics of the offloading strategies. The main objectives observed are to decrease latency and energy consumption. We identified that the strategies consider constraints of several computational resources to model their solutions, mainly processing power, energy consumption, and latency of computational nodes. Big Data applications, real-time image processing, and DSP are the main applications tested in the simulations. The use of heuristics, models, AI, game theory, among others, shows different solutions.

The impact on data and application security needs to be better evaluated in strategies, including security as an objective of offloading and improving the assessment of the impact on application performance. Finally, we identified some possibilities for future work. We placed some open questions that can be better explored by offloading strategies, such as using idle resources on embedded devices to perform tasks at the Edge.

Acknowledgment

This study was financed in part by the CAPES – Finance Code 001, CNPq – 313111/2019-7, FAPESP – 2018/23092-1, 2020/05183-0, 2020/05115-4; FAPERGS – 19/2551-0001266-7, 19/2551-0001224-1, 19/2551-0001689-1, 21/2551-0000688-9.

REFERENCES

[1] S. Sorrell, "The internet of things: Consumer, industrial & public services 2018-2023," *Juniper, Sunnyvale, CA, USA*, 2018.

[2] J. Wu, Z. Cao, Y. Zhang, and X. Zhang, "Edge-cloud collaborative computation offloading model based on improved partical swarm optimization in mec," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, 2019, pp. 959–962.

[3] T. Zheng, J. Wan, J. Zhang, C. Jiang, and G. Jia, "A survey of computation offloading in edge computing," in *2020 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2020, pp. 1–6.

[4] A. Bhattcharya and P. De, "Computation offloading from mobile devices: Can edge devices perform better than the cloud?" in *Proceedings of the Third International Workshop on Adaptive Resource Management and Scheduling for Cloud Computing*, ser. ARMS-CC'16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1–6. [Online]. Available: https://doi.org/10.1145/2962564.2962569

[5] A. Alelaiwi, "An efficient method of computation offloading in an edge cloud platform," *Journal of Parallel and Distributed Computing*, vol. 127, 05 2019.

[6] B. Nour, S. Mastorakis, and A. Mtibaa, "Whispering: Joint service offloading and computation reuse in cloud-edge networks," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[7] R. Dautov and S. Distefano, "Stream processing on clustered edge devices," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020.

[8] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 56–62, 2019.

[9] A. B. De Souza, P. A. L. Rego, T. Carneiro, J. D. C. Rodrigues, P. P. R. Filho, J. N. De Souza, V. Chamola, V. H. C. De Albuquerque, and B. Sikdar, "Computation offloading for vehicular environments: A survey," *IEEE Access*, vol. 8, pp. 198 214–198 243, 2020.

[10] A. K. Jha, M. Patel, and T. Pawar, "Fog offloading: Review, research opportunity and challenges," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019, pp. 1224–1227.

[11] S. Talal, W. S. M. Yousef, and B. Al-Fuhaidi, "Computation offloading algorithms in vehicular edge computing environment: A survey," in *2021 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, 2021, pp. 1–6.

[12] X. Shan, H. Zhi, P. Li, and Z. Han, "A survey on computation offloading for mobile edge computing information," in *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*, 2018, pp. 248–251.

[13] M. Zamzam, T. El-Shabrawy, and M. Ashour, "Game theory for computation offloading and resource allocation in edge computing: A survey," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2020, pp. 47–53.

[14] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," *IEEE Access*, vol. 8, pp. 186 080–186 101, 2020.

[15] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131 543–131 558, 2019.

[16] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," ser. EASE'08. Swindon, GBR: BCS Learning amp; Development Ltd., 2008, p. 68–77.

[17] C. Okoli, "A guide to conducting a standalone systematic literature review," *Communications of the Association for Information Systems*, vol. 37, 11 2015.

[18] X. Huang, Y. Yang, and X. Wu, "A meta-heuristic computation offloading strategy for iot applications in an edge-cloud framework," in *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3386164.3390513

[19] Y. Huang, B. Lin, Y. Zheng, J. Hu, Y. Mo, and X. Chen, "Cost efficient offloading strategy for dnn-based applications in edge-cloud environment," in *2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 2019, pp. 331–337.

[20] T. Wang, Y. Liang, Y. Zhang, X. Zheng, M. Arif, J. Wang, and Q. Jin, "An intelligent dynamic offloading from cloud to edge for smart iot systems with big data," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2598–2607, 2020.

[21] Y. Hao, Y. Jiang, T. Chen, D. Cao, and M. Chen, "itaskoffloading: Intelligent task offloading for a cloud-edge collaborative system," *IEEE Network*, vol. 33, no. 5, pp. 82–88, 2019.

[22] Y. Zhang, B. Di, Z. Zheng, J. Lin, and L. Song, "Joint data offloading and resource allocation for multi-cloud heterogeneous mobile edge computing using multi-agent reinforcement learning," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[23] R. Dautov, S. Distefano, D. Bruneo, F. Longo, G. Merlino, and A. Puliafito, "Data processing in cyber-physical-social systems through edge computing," *IEEE Access*, vol. 6, pp. 29 822–29 835, 2018.

[24] X. Yang, Z. Fei, J. Zheng, N. Zhang, and A. Anpalagan, "Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 018–11 030, 2019.

[25] F. R. de Souza, M. Dias de Assunçao, E. Caron, and A. da Silva Veith, "An optimal model for optimizing the placement and parallelism of data stream processing applications on cloud-edge computing," in *2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 2020, pp. 59–66.

[26] A. Qin, C. Cai, Q. Wang, Y. Ni, and H. Zhu, "Game theoretical multi-user computation offloading for mobile-edge cloud computing," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019, pp. 328–332.

[27] L. Yang, Z. Dai, and K. Li, "An offloading strategy based on cloud and edge computing for industrial internet," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2019, pp. 1666–1673.

[28] Z. Wang, T. Wu, Z. Zhang, and H. Zhou, "A game theory-based computation offloading method in cloud-edge computing networks," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–6.

[29] V. De Maio and I. Brandic, "First hop mobile offloading of dag computations," in *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2018, pp. 83–92.

[30] G. e. J. N. e. F. L. e. Z. Z. J. Wu, YiMing e Lu, "Trusted fog computing for privacy smart contract blockchain," in *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*.

[31] W. Li, M. He, W. Zhu, and J. Zheng, "A study on lightweight and secure edge computing based blockchain," in *2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS)*, 2021, pp. 256–261.

[32] K.-L. Wright, M. Martinez, U. Chadha, and B. Krishnamachari, "Smart-edge: A smart contract for edge computing," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1685–1690.

[33] S. Xu, C. Guo, R. Q. Hu, and Y. Qian, "Blockchain inspired secure computation offloading in a vehicular cloud network," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[34] S. Battula, S. Garg, R. Naha, M. Amin, B. Kang, and E. Aghasian, "A blockchain-based framework for automatic sla management in fog computing environments," *The Journal of Supercomputing*, 05 2022.

[35] X. Huang, X. Liu, Q. Chen, and J. Zhang, "Resource allocation and task offloading in blockchain-enabled fog computing networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 01–05.

[36] J. Zhang, Y. Huang, F. Ye, and Y. Yang, "A novel proof-of-reputation consensus for storage allocation in edge blockchain systems," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 2021, pp. 1–10.

[37] S. Iqbal, A. W. Malik, A. U. Rahman, and R. M. Noor, "Blockchain-based reputation management for task offloading in micro-level vehicular fog network," *IEEE Access*, vol. 8, pp. 52 968–52 980, 2020.

[38] Y. Du, Z. Wang, J. Li, L. Shi, D. N. K. Jayakody, Q. Chen, W. Chen, and Z. Han, "Blockchain-aided edge computing market: Smart contract and consensus mechanisms," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.