Cluster Computing (2024) 27:5517–5545
https://doi.org/10.1007/s10586-023-04256-8

Resource optimization in edge and SDN-based edge computing: a comprehensive study

Ajay Nain[1] · Sophiya Sheikh[1] · Mohammad Shahid[2] · Rohit Malik[1]

Received: 22 August 2023 / Revised: 20 November 2023 / Accepted: 21 December 2023 / Published online: 8 February 2024

Abstract
The proliferation of mobile devices and the increasing use of networked applications have generated enormous data that require real-time processing and low-latency responses. For real-time applications, Edge Computing (EC) gained a lot of attention which helps to overcome the challenges with traditional computing systems. There is a massive growth of data generated at the web edge, but its limited computing resources and boundary dynamics pose significant challenges. For proficient resource utilization and network management, Software-Defined Networks (SDN) integration with EC can provide control and programmability in edge networking management. However, the optimization of resources is a fundamental need and challenge of EC based on SDN. Besides, the utilization of SDN in EC deals with a centralized and distributed infrastructure further enables the processing of data in closer proximity to its origin and helps in the growing demand for efficient and high-performance computing systems. This study presents a systematic review of the current state-of-the-art pertaining to resource optimization in EC and its integration with SDN. The review of challenges, benefits, and various proposed methodologies for managing resources in Edge and SDN-based Edge environments are investigated. This paper offers valuable insights into the current state and future directions of Edge, SDN-based EC systems, which can assist organizations in designing and deploying more efficient resource consumption solutions

Keywords Resource optimization · Resource management · Edge computing · SDN-based edge computing · Dynamic environment

1 Introduction

In real-time applications, the amount of data continues to grow rapidly in both volume and complexity. Therefore, it is necessary to adopt high-performance computing systems to manage this substantial workload effectively. Due to high latency in traditional network systems, there is a need for devices to process data and execute tasks closer to where the data is generated or used, rather than sending all the data to a remote data center or the cloud [1]. Commonly all computational and network resources located between data centers and data sources are referred to as "Edge" resources [2]. The EC platform enhances data processing by bringing computing resources in close proximity to the data sources. This results in accelerated processing speed, minimized network latency, and an enhanced user experience.

EC comprises a decentralized computing architecture in which computing resources are positioned at the network edge, near the data sources and end users. Numerous

Sophiya Sheikh, Mohammad Shahid, and Rohit Malik have contributed equally to this work.

✉ Sophiya Sheikh
sophiyasheikh@gmail.com

Ajay Nain
mr.ajaynain@gmail.com

Mohammad Shahid
mdshahid.cs@gmail.com

Rohit Malik
rohitxmalik7@gmail.com

[1] Department of Computer Application, Lovely Professional University, Phagwara, Punjab 144411, India

[2] Department of Commerce, Aligarh Muslim University, Aligarh, Uttar Pradesh 202002, India

benefits can be derived from this closeness, such as lower data transmission costs, quicker responses, more privacy, and the capacity to handle data-intensive applications with stringent latency requirements. The edge infrastructure is composed of various components, such as edge servers, gateways, access points, and Internet of Things (IoT) devices. This combination is essential for a distributed and heterogeneous computing environment.

Besides, efficient resource allocation is required to enable optimal utilization due to the limited availability of storage and computing resources at the edge. However, there are a number of reasons why resource optimization in EC systems is difficult. The task of resource optimization is made more challenging by the dynamic nature of edge environments, which are characterized by fluctuating workloads and resource needs. Also, the process of resource optimization involves the dynamic allocation of processing power, network bandwidth, storage, and other resources to satisfy the varying needs of a variety of applications while taking into consideration restrictions such as energy consumption and cost. This process is carried out to maximize efficiency and reduce costs. Effective resource management and orchestration become more difficult due to the variety of devices and networks and their different capabilities [5].

Furthermore, the amalgamation of SDN and EC has become a viable option [3] to effectively meet these demands [4]. SDN-based EC incorporates the benefits of both paradigms, creating a decentralized and distributed infrastructure that can carry out data processing near its source of generation. The computing systems integrated with SDN lead to the optimization of the available resources. Also, it is one of the most important factors that determines how effectively the available computing resources are utilized.

In SDN-based EC, the primary goal of resource optimization is to enhance system performance and assure the best use of the resources that are currently available. This may be accomplished by ensuring that the resources are used to their fullest potential [6]. It is possible to achieve increased service quality, lower latency, and better user experiences by effectively allocating their available resources. In addition, optimizing resources plays a vital role in improving energy efficiency. This is accomplished by ensuring that computer resources are used properly, while simultaneously limiting power consumption that is not essential [7].

However, the dynamic environment of SDN-based EC makes resource optimization quite complex. The network is defined by the dynamic nature of its component devices and applications, which are capable of joining or quitting the network at any given moment and demonstrating great unpredictability in their resource needs [8]. In this specific setting, traditional resource allocation techniques for static situations are unsuitable and should not be used. As a direct consequence of this, innovative strategies and procedures are required to be devised.

Resultantly, it is vital to create and carry out algorithms that are both adaptive and scalable to address the issues associated with resource optimization in EC that is based on SDN. This can only be accomplished by using a competent distributed computing model. The algorithms need to be able to make adaptive changes to the allocation of resources in real-time in response to changes in the circumstances of the network and the requirements of the applications. Continuous system monitoring allows for the formulation of choices on resource allocation in reaction to shifting needs. These decisions allow for the maintenance of optimal resource utilization at all times [9].

The use of SDN to optimize resource utilization in EC has major implications from a pragmatic point of view. The concept of EC is becoming more popular in a variety of fields, including smart cities, industrial automation, healthcare, and transportation, among others. For businesses to successfully exploit the full potential of EC based on SDN, they must first optimize the allocation of their resources. This will result in higher productivity, lower expenses, and enhanced service quality [10].

1.1 Background of the study

The proliferation of data stemming from diverse applications and devices, including but not limited to IoT devices, smart appliances, and sensor networks, has resulted in a fundamental transformation of computing architectures in recent times. The increasing need for rapid data processing, minimal latency, and effective management of voluminous data is already posing challenges to the conventional centralized cloud computing models. Consequently, the concept of EC has garnered considerable interest as a potentially effective strategy for tackling these issues [11].

The concept of EC involves the proximity of computing resources to the data source, which facilitates expedited processing and analysis of data at the network edge. The utilization of this methodology diminishes the latency encountered during the transmission of data to a centralized cloud or data center, rendering it appropriate for applications that are sensitive to latency, such as autonomous vehicles, industrial automation, and real-time monitoring systems. In addition, the concept of EC presents the possibility of enhanced scalability, dependability, and robustness through the dispersion of computational responsibilities across a network that is decentralized [12].

The technology of SDN has been identified as a revolutionary approach that separates the control plane from the data plane in network infrastructure. The implementation of SDN allows for the centralization of network management and programmability, resulting in improved flexibility, agility, and control over network resources. Through the integration of SDN and EC, entities can establish adaptive, decentralized, and code-driven network frameworks that effectively handle and oversee data at the periphery.

The effective allocation and management of resources in EC environments based on SDN present notable difficulties. Adaptive and scalable resource optimization techniques are required due to the dynamic nature of the network, where devices and applications can join or leave the network at any given time [13]. In addition, it is important to note that the resource requirements of different applications may differ significantly, necessitating the use of resource allocation algorithms that can adapt dynamically to accommodate evolving demands.

The objective of resource optimization in EC based on SDN is to achieve efficient utilization of various computing resources, such as computational power, network bandwidth, storage, and energy, while taking into account factors such as cost, energy efficiency, and application performance requirements, and adhering to relevant constraints. Through the optimization of resource allocation, organizations can attain various advantages such as enhanced system performance, decreased latency, improved energy efficiency, and cost-effectiveness [14].

The optimization of resources has been the subject of significant research in conventional cloud computing settings. However, the distinctive features of EC based on SDN introduce novel prospects and obstacles. The unique characteristics of the EC environment, which is both decentralized and distributed, coupled with the requirement for instantaneous processing and adaptable resource allocation, mandate the creation of customized resource optimization methodologies that are specifically designed for SDN-based EC frameworks.

1.2 Motivation of the study

The amount and complexity of data created by numerous applications and devices, including IoT devices, smart appliances, autonomous cars, etc., have rapidly increased in recent years. Due to the increase in data, there is a rising need for high-performance, efficient computing systems that can handle the massive workload. Therefore, the combination of EC and SDN has become a potential method of satisfying these needs. SDN-based EC offers a distributed infrastructure that can handle data closer to its source, combining the advantages of both the SDN and EC concepts. Besides, real-time applications can be handled successfully by minimizing latency. However, in such a dynamic and scattered context, managing and allocating resources effectively presents major difficulties.

Resource optimization is a major issue for SDN-based EC. To maintain maximum performance and usage, it is essential to deploy computing resources wisely given the variety of devices and applications that are producing different data streams. In this context, resource optimization entails dynamically distributing computing power, network bandwidth, storage, and other resources to satisfy the needs of diverse applications while considering a variety of restrictions, including energy consumption and cost.

Additionally, the extremely dynamic nature of the environment makes resource optimization in SDN-based EC much more challenging. Applications and devices may enter and exit the network at any moment, and the number of resources they use might change frequently. Traditional resource allocation methods, which were created for static systems, are thus insufficient for this situation. To solve the resource optimization issues unique to SDN-based EC, new methods and strategies must be devised [15].

The goal of this paper is to collate cutting-edge resource optimization methods specifically designed for SDN-based EC systems. The major focus of this paper is to analyze the techniques that help to increase system performance and ensure efficient use of the resources that are already accessible. This research will also concentrate on the study of scalable and adaptable algorithms that can dynamically change resource allocation in response to changing network and application needs.

1.3 Contributions and research gap

This section presents the major contributions of the paper followed by the existing research gap. The contributions are as follows.

1. A detailed literature review and comprehensive analysis related to EC and its integration with SDN have been carried out. Besides, the main challenges faced in deploying and managing IoT-Edge and SDN networks are investigated. Also, various approaches for addressing these challenges are discussed.
2. The fundamental contributions of resource optimization to improve performance in EC and its integration with SDN are further identified.
3. An architecture for developing scalable and reliable IoT-Edge, SDN-based Edge, and SDN systems is implemented and discussed. It provided insights into how these technologies can be integrated and how their performance could be enhanced.
4. This paper offers valuable insights into the current state and future directions of IoT, Edge, SDN, and their

integration, which can assist organizations in designing and deploying more efficient resource solutions.

Besides, we have done a comparison with other related surveys which are shown in Table 1.

The present review of art focuses on some crucial points, new concepts, and solutions that were minimal in existing literature. As [16] performs the analysis of resource optimization in EC but does not evaluate it with SDN. Additionally, it does not include the open issues of both edge and SDN. In [17], the author does not investigate the efficient resource utilization techniques. Besides, in [12] and [18], the author does not contemplate the adequate resource distribution in SDN-based EC. In [19] and [20], authors have discussed resource optimization in an edge environment but do not consider other important aspects of load balancing, resource allocation, and computational offloading in an edge environment. To the best of our knowledge, we observed that there is still room in the literature to focus on these crucial aspects of EC and the behavior of SDN while integrating with EC. This analysis has been covered and presented in Table 1.

1.4 Organization of paper

Section 1 introduces SDN-based EC and resource optimization. The background, motivation, and contribution to the study are also included. Section 2 of the paper focuses on the plan of study, which encompasses the nature of searching, inclusion and exclusion criteria, review judgments, research methodology, and our contribution Section 3 presents the taxonomy of edge and SDN-based EC systems. Section 4 delves into EC, including its integration with the IoT-CC-Edge architecture, advantages, challenges, and optimization techniques with their related work.. Section 5 discusses the SDN of Things using edge and analyses the related literature of the SDN-IoT-Edge Framework, IoT-Edge management, SDN/Network Function Virtualization (NFV) standardization, and resource provisioning management on the edge using SDN Section 6 illustrates open issues, challenges, and current trends for the Integration of SDN-based edge and IoT. Section 7 concludes the paper, summarizing the key points discussed in the previous sections and future scope.

2 Plan of study

According to Prisma 2020, Kitchenham, and Charters' 2007 articles, the systematic literature review (SLR) methodology is used in the current investigation [21, 22]. Our objective is to conduct a comprehensive evaluation of various research papers and examine the contributions of prior researchers to furnish a methodical review that can assist fellow researchers. This section focuses on and indicates the sources of collection of the existing state of the art such as sources of data and different requirements for research.

2.1 Nature of searching

A systematic survey of resource optimization in EC and SDN-based edge environments was conducted through well-known sources. Several search keywords include that are optimization in edge environment, resource

Table 1 Comparison with other related surveys

Article details	Analysis of edge computing	Analysis of SDN	Architecture specification of SDN and edge computing	Resource optimization in Edge computing	Resource optimization in SDN-based edge computing	Challenges & Open issues in edge and SDN
2023 & [16]	✓			✓		
2022 & [17]	✓	✓			✓	
2022 & [12]	✓			✓		
2021 & [18]	✓	✓	✓			✓
2021 & [19]	✓			✓		
2019 & [20]	✓		✓	✓		✓
Our survey	✓	✓	✓	✓	✓	✓

optimization in SDN-based edge environment, overview of SDN-based edge paradigm, resource provisioning in SDN-based edge environment, dynamic resource allocation, and resource allocation in SDN-based IoT-Edge management.

2.2 Sources of data

Numerous reliable sources are selected for reviewing and surveying the entire literature to assemble the different methods and approaches employed in this study. Also, all the journal and conference papers are found using databases like Scopus, WoS, SCI, SCIE, and others. The papers are carefully looked through to find all the applicable fields where the techniques suggested are implemented. Some of the frequent sources of manuscript are given in Table 2.

The statistics of the selected papers according to the percentage of published articles by various publishers is presented in Fig. 1. It is evident from the figure that the major portion of published sources belong to IEEE. Springer is the second major source for related articles. Remaining sources are from Wiley, ACM and Elsevier.

2.3 Duration and validity of study

This research review paper mainly contains papers from 2013 to 2023 and all the papers were selected from well-reputed journals, books, and conferences. The statistics of the included year of publications from 2013 to 2023 are presented in Fig. 2.

The graph plotted reveals that the maximum number of manuscripts were published between the years 2016 to 2019. The period between 2013 to 2019 was a booming period for publication in the concerned fields. However, it gradually decreased after 2019.

2.4 Criteria for inclusion and exclusion

Our sole basis for considering articles (Inclusion) is whether they meet the selection requirements or not. While the exclusion criteria serve as the basis for excluding research papers from the study. Also, a flowchart for the inclusion and exclusion criteria of the selection of papers is given in Fig. 3.

Table 2 Frequent sources of manuscript

Elsevier	https://www.elsevier.com
Springer	http://springer.com
IEEE	http://ieeexplore.ieee.org
ACM Digital Library	https://dl.acm.org
Wiley Online	https://onlinelibrary.wiley.com

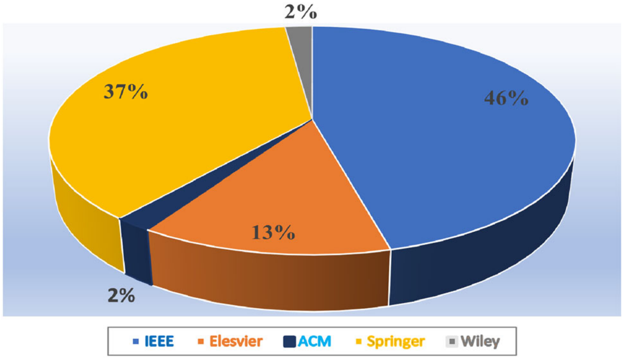

Fig. 1 Percentage of manuscripts from frequent sources

Inclusion criteria

- Articles composed in the English language.
- The aforementioned search strings are pertinent to the subject matter at hand.
- A research paper that can be verified through reliable sources.
- The research articles that are published in reputed Journal/Conferences/Books.
- The publication dates of the materials under consideration range from 2012 to 2023.
- The papers must satisfy the criteria of nature of searching as discussed in section 2.1.

Exclusion criteria

- The discussion lacks focus on the pertinent concerns surrounding the IoT, Edge, and SDN technology.
- An enough attention was not given to the sole optimization issues posed by combining SDN-Edge Technology with the IoT.
- The search criteria were not met as mentioned in Sect. 2.1.
- The article has not undergone peer review.
- The publication predates the year 2010.

2.5 Review judgement

This section focuses on the background of the methods which are used to perform qualitative judgment in the review. The inclusion and exclusion of the articles published was based on certain aspects. The selection of the related articles was done only after analyzing the abstract. After article selection, the article was fully reviewed, and inclusion and exclusion were applied based on standards as mentioned in Fig. 3. Besides, the inclusion of articles based on the following conditions:

- Resource optimization in edge environment and integration of cloud-edge with SDN.
- Different architectures in edge and SDN paradigms.

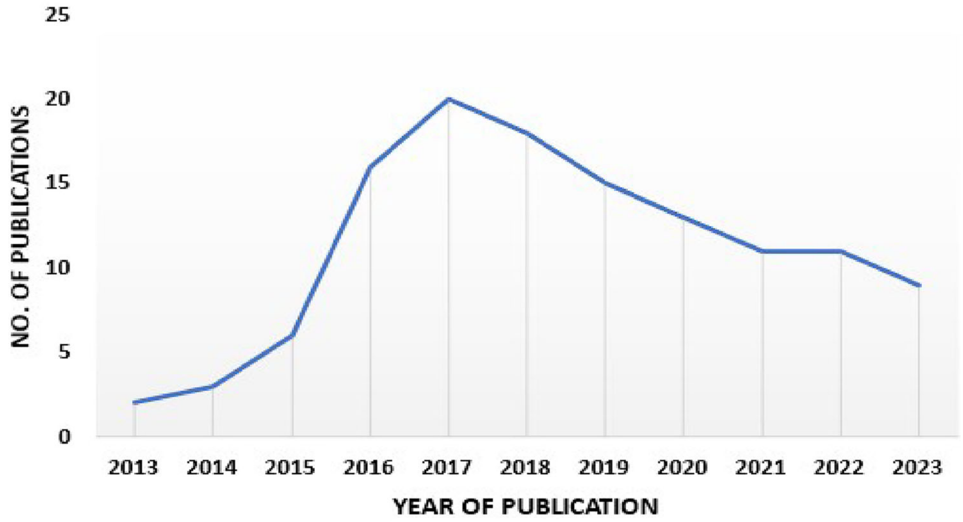

Fig. 2 Number of articles (year wise)

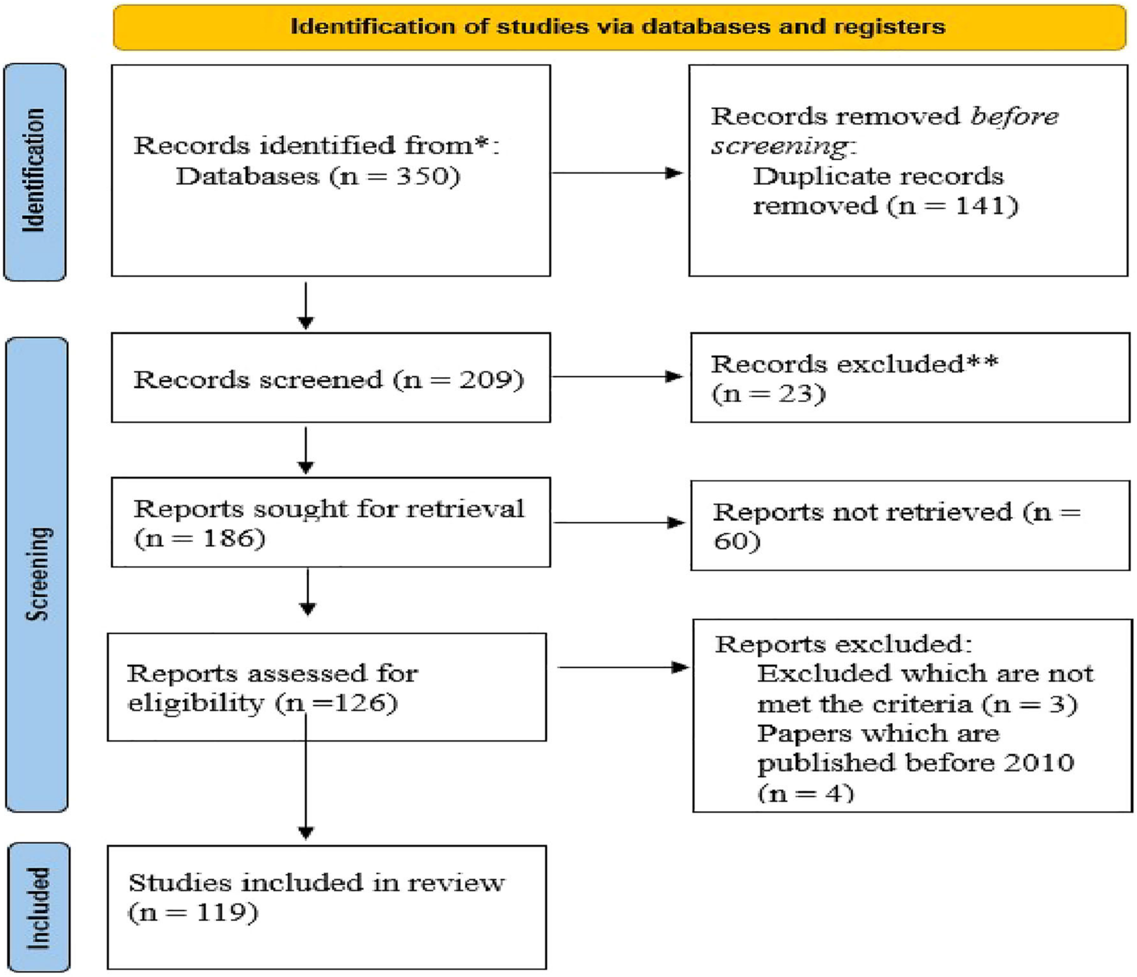
Identification of studies via databases and registers
Identification
Records identified from*:
Databases (n = 350)
Records removed *before screening*:
Duplicate records removed (n = 141)
Screening
Records screened (n = 209)
Records excluded**
(n = 23)
Reports sought for retrieval (n = 186)
Reports not retrieved (n = 60)
Reports assessed for eligibility (n =126)
Reports excluded:
Excluded which are not met the criteria (n = 3)
Papers which are published before 2010 (n = 4)
Included
Studies included in review (n = 119)

Fig. 3 Flow chart of inclusion exclusion criteria

- Also proposed articles were critically analyzed based on the techniques, findings, optimized parameters, shortcomings, etc.

2.6 Adopted methodology

To gain an all-encompassing understanding of research in the field at hand various methods were utilized in our comprehensive review process. Five electronic databases were searched via specific search terms which underwent screening for eligibility based on predefined criteria by two reviewers. Full texts that met our standards were then reviewed in depth while extracting pertinent data from eligible studies via standardized forms—findings that underwent synthesis through both narrative approaches as well as meta-analytic techniques when applicable—and received quality assessments utilizing standardized tools upon inclusion into our analysis pool. Additionally, hand-searching of reference lists and citation searches were conducted for further comprehensiveness. The inherent value of this methodology is the ability to identify and analyze the most relevant and rigorous studies ultimately providing an extensive overview of research in the field. As a final step, the findings are summarized in a clear and concise manner, providing an overview of the state of research in the field. Graphical interpretation of the steps followed are shown in Fig. 4.

3 Taxonomy of the study

EC is a revolutionary field that unites various industries, including telecommunications, automotive manufacturers, IoT, cloud service providers, and emerging technologies. The edge environment contains several components and is connected to different devices and networks. Integration of SDN in Edge makes it a more efficient computing technology. In this section, we discuss the architectural specification of edge, SDN, and the objectives of both technologies. The graphical representation of the taxonomy of the paper is depicted in Fig. 5.

3.1 Architectural specification

The architecture of Edge and SDN have their objective and specifications, as shown in Figs. 6 and 7. In the world of EC, it is all about bringing computation closer to where it is needed. Resource location and mobility are two important aspects of EC, which are discussed below.

Resource location It is a set of guidelines that specify how devices, systems, and applications in an EC

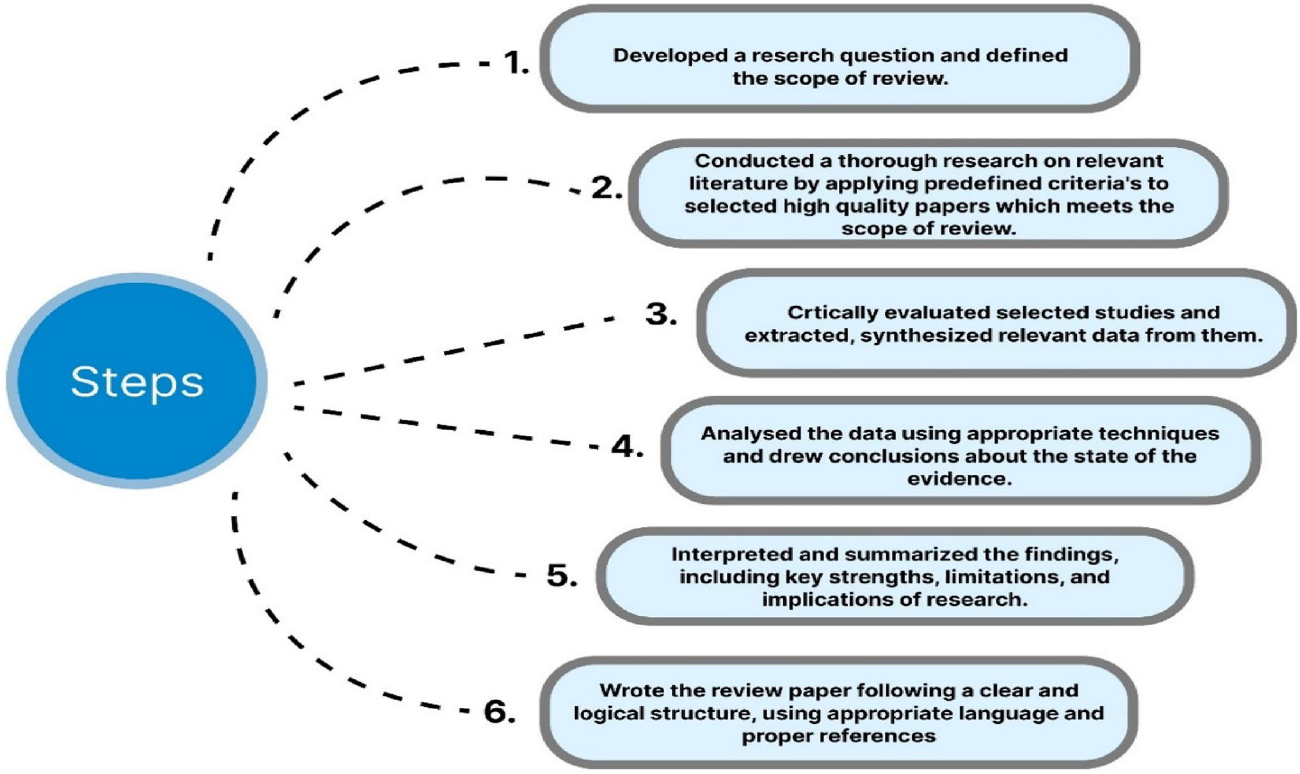
Steps
1.
Developed a reserch question and defined the scope of review.
2.
Conducted a thorough research on relevant literature by applying predefined criteria's to selected high quality papers which meets the scope of review.
3.
Crtically evaluated selected studies and extracted, synthesized relevant data from them.
4.
Analysed the data using appropriate techniques and drew conclusions about the state of the evidence.
5.
Interpreted and summarized the findings, including key strengths, limitations, and implications of research.
6.
Wrote the review paper following a clear and logical structure, using appropriate language and proper references

Fig. 4 Schematic diagram of adopted methodology

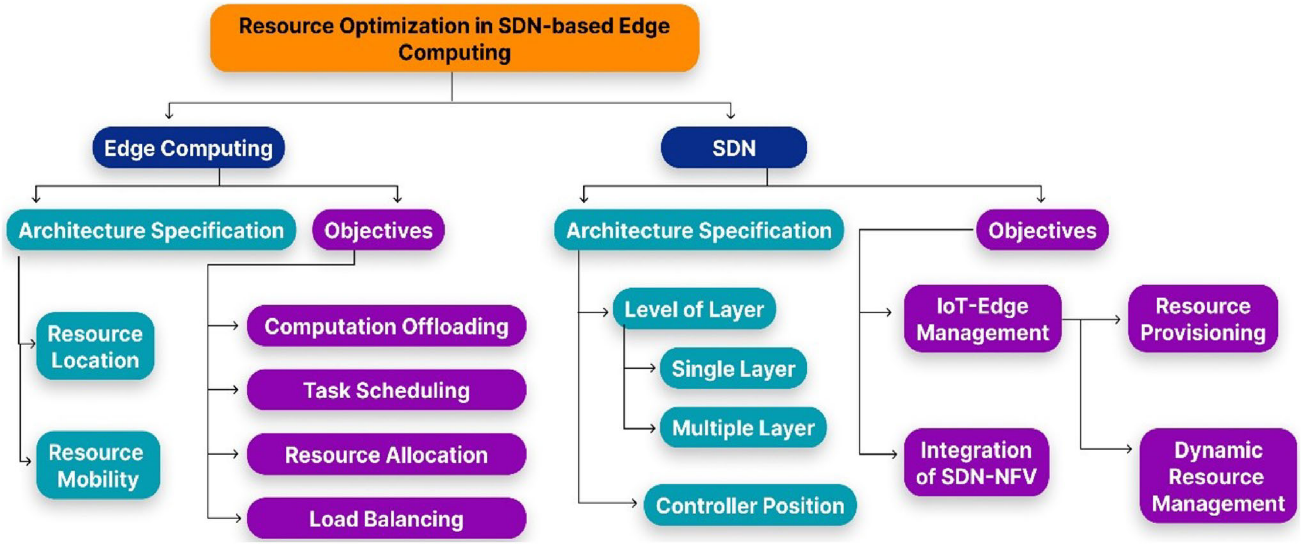

Fig. 5 Taxonomy study

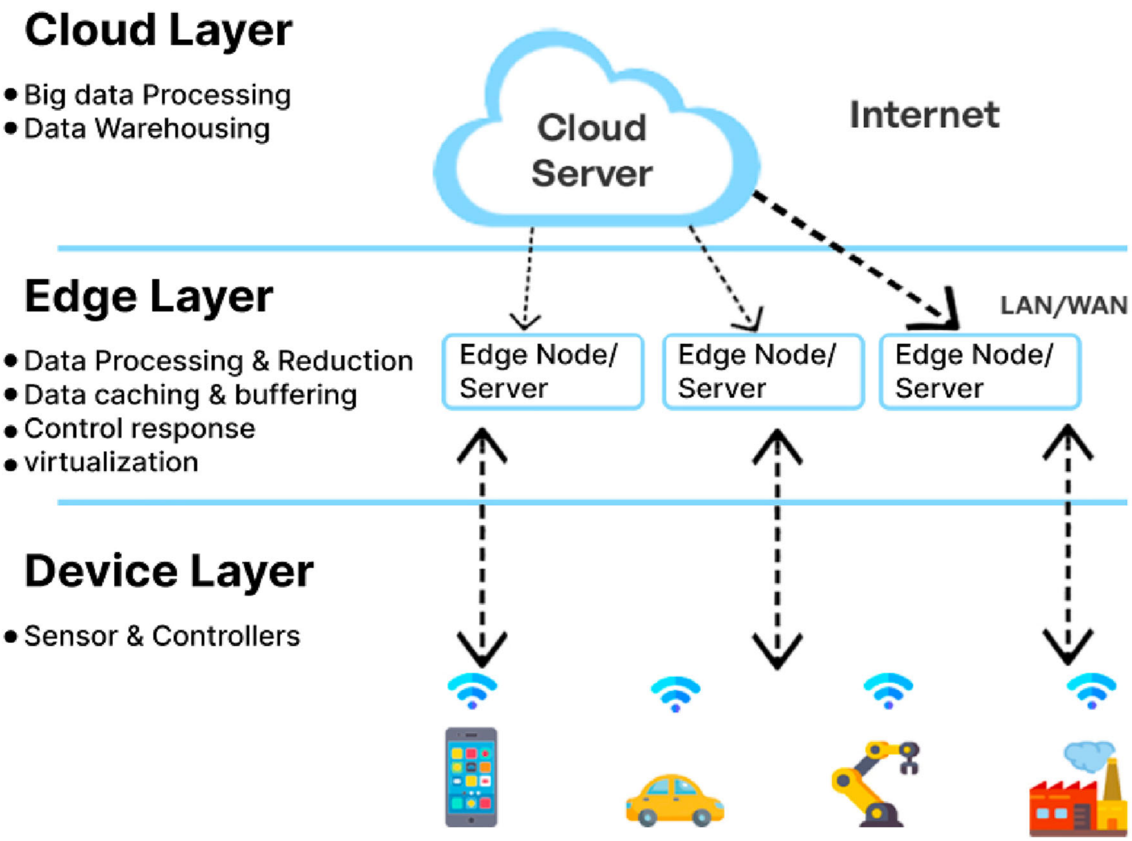

Fig. 6 Edge computing architecture

environment can locate and use computing resources. To meet the performance and functionality needs of EC systems, these standards describe the protocols, processes, and algorithms that facilitate efficient resource identification and allocation.

Resource mobility In a distributed EC environment, resource mobility means that computer resources like data, services, or processing power can move, change, or adapt. Mobility ensures that resources can be dynamically distributed/allocated to maximize their responsiveness and

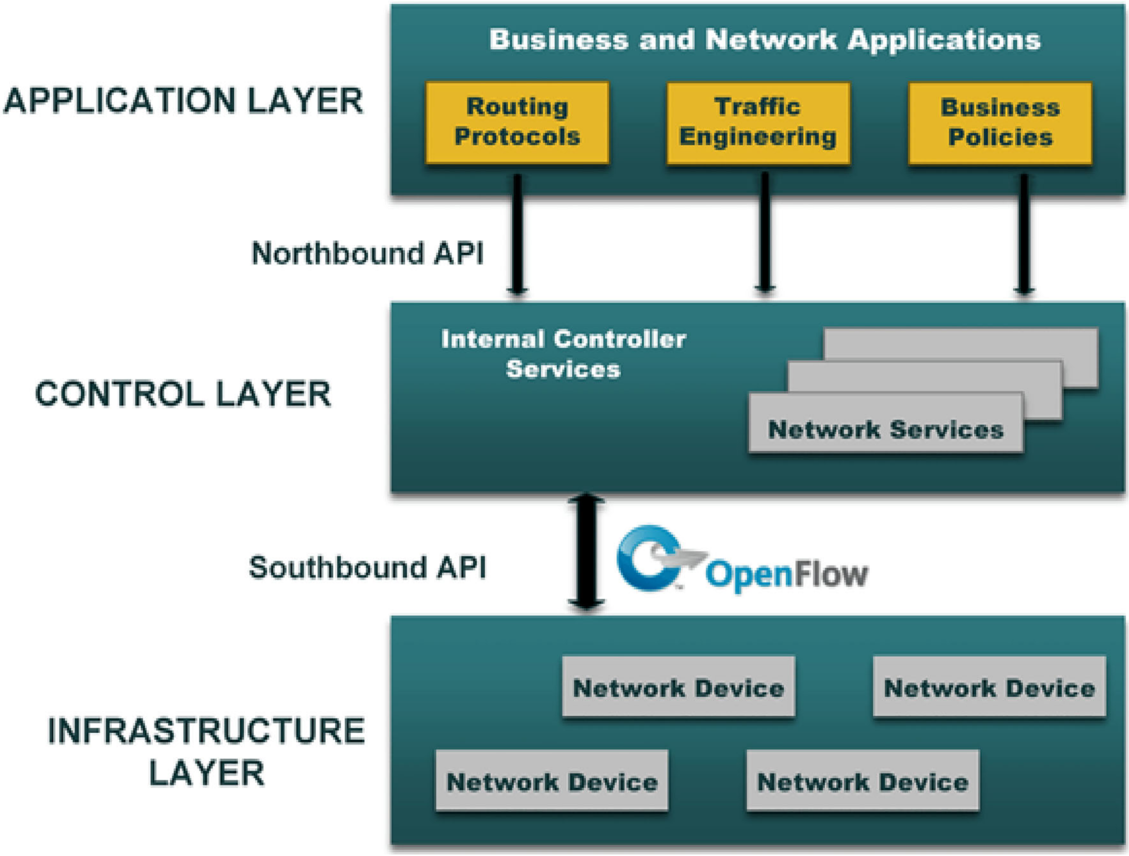

Fig. 7 SDN architecture

availability with the changing needs and expectations of users/applications. Resource mobility specifications include the rules, steps, and techniques that can effectively oversee and coordinate the transfer of these resources.

The Aachitectural specifications of SDN are discussed below:

Single-layer specifications These offer a detailed method for handling particular areas of EC, such as resource distribution, security guidelines, or quality of service. To effectively meet objectives and deal with unique issues or requirements within the edge environment, single-layer specs can be helpful.

Multi-layer specifications Multi-layer specifications are essential in complex and networked EC. They contribute to making the edge architecture function as an entire system. Determining how SDN works with application requirements, controls resources throughout the infrastructure, and maintains segregation and security regulations. Multi-layer requirements guarantee that the EC environment operates as an integrated unit with an extensive overview.

Controller position It is a central element of the SDN architecture responsible for decision-making, resource management, and data flow control. By giving network devices a single point of control, the controller serves as the “brain” of the SDN.

3.2 Objectives for edge and SDN

Objectives of edge adoption in the present study Optimizing resource utilization provides essential solutions like computational offloading, load balancing, and resource allocation. By moving certain processing processes from centralized data centers to edge devices, task offloading dramatically lowers latency and improves the performance of real-time applications. Load balancing is essential to balance the load on edge devices, preserve system stability, and ensure that computing jobs are dispersed equally across the resources. Adequate allocation of CPU, memory, and storage to specific activities or applications ensures resource contention-free operation. These principles are critical to attaining high performance and low latency in an edge environment.

Objectives of SDN adoption in the present study SDN plays a critical role in IoT-Edge management. SDN offers a centralized control layer that enables edge-device network resource orchestration and optimization. EC can effectively distribute network resources to IoT devices and applications by attaining SDN principles. It ensures that data transmission accelerates and that the appropriate resources are distributed as and how it is required. The world of IoT applications is made more dynamic, responsive, and resource-efficient by the integration of SDN into edge management and resource provisioning for IoT devices.

4 Edge computing

Over the last few years, there has been a significant increase in global interconnectivity, resulting in a corresponding surge in the generation and transmission of data across networks. EC [14, 15] was introduced as a solution to bring computing technologies in more proximity to data generation devices as an alternative to depending on centralized data centers or clouds. EC aims to minimize latency, increase efficiency, and improve the reliability of computer systems. The key point for EC is its ability to process and analyze data closer to its sources of generation. EC devices are deployed near the physical devices or to the network edge. Edge servers are positioned to reduce the volume of data that needs to be sent to distant cloud data centers. This, in turn, enables dynamic decision-making and faster response times, making it a perfect solution for run-time applications, i.e., IoT, autonomous vehicles, industrial automation, smart cities, etc [23, 24]. EC can increase productivity in industrial automation by processing data in place, in real time. EC can support decision-making in autonomous vehicles by processing information from sensors quickly. Additionally, by enabling local decision-making in IoT devices, EC lessens the need for constant connectivity with the internet and speeds up response times. Besides, due to the tremendous use of these technologies, the data created and produced from devices and sensors at the network edge continues to grow, and EC is becoming increasingly important.

Imagine a smart city where IoT devices are installed across the city to monitor traffic, detect air pollution levels, and manage waste collection. With EC, these devices can communicate with each other and process data locally. For example, traffic sensors can communicate with streetlights to optimize [25] traffic flow, reducing congestion and carbon emissions. Waste collection trucks can be routed more efficiently based on real-time data on waste levels in different parts of the city. This approach reduces costs, improves efficiency, and enhances the quality of life for residents.

To carry out calculations and offer storage services within the scope of a radio access network, processing/computing servers are located at the periphery of the network in the EC architecture. Enhancing the end user's Quality of Experience (QoE) is the primary goal of edge architecture, achieved by minimizing response time, latency, and throughput. It enables instant decision-making for a variety of time-sensitive services and real-time applications.

Since EC possesses various applications and advantages over the CC (Cloud Computing), it also poses some challenges to the network. To analyze the various advantages and challenges of EC, it is important to study the architecture of EC in the context of what it does and how it can be implemented. Unlike CC, EC embeds computational edge nodes near the network, also known as edge servers, and acts as mini-cloud data centers. Generally, the EC architecture [26] is divided into three aspects the front end specifies the data generation layer or IoT devices, the near-end defines edge nodes, and the far end comprises CC [23], as shown in the Fig. 6 described below.

Front end/physical layer/device layer The end devices (e.g., IoTs, sensors) are strategically positioned at the forefront of the EC architecture. The frontend environment has the potential to enhance user interaction and improve responsiveness for end users. EC can provide real-time services for some applications, leveraging the abundant computing capacity offered by a variety of nearby end devices. Since these end devices possess limited capabilities, most objectives cannot be adequately fulfilled inside the end device environment [27]. Therefore, in these instances, the end devices are required to transmit on-demand resources to the servers.

Near end/middle layer/edge layer The gateways that are implemented in the middle layer which are in proximity to the device layer will facilitate most of the traffic streams within the networks. Edge/cloudlet servers may have various resource requirements for tasks such as real-time data processing and reduction, data caching and buffering, compute offloading, and response control. In EC, a significant portion of data computation and storage is transferred to edge servers that are present near the end devices. As a result, the end devices can enhance their performance in data computing and storage, with a slight decrease in latency.

Far end/cloud layer When the cloud servers are placed away from the data-creating devices, the result is noticeable delays in data transmission across the networks. However, cloud servers located in remote environments can provide increased computing power and data storage capabilities. Cloud servers offer a wide range of solutions in massively parallel data processing, big data management & mining, and machine learning [28, 29].

4.1 Advantages

The architecture of EC having the advantages over a traditional network are discussed below:

Low latency EC is a technology that greatly reduces the amount of time it takes for data to travel from where it is generated to where it is processed. This helps to minimize any delays [30] that may occur during this process. This has many benefits in different applications, particularly in tasks that need immediate responses, as it ensures faster decision-making due to being deployed at the very edge.

Bandwidth optimization Edge devices can process data locally, which means they can handle various data processing tasks without sending them to a central server or cloud. It helps to minimize the time of data transmission and improves overall efficiency. Conserving network bandwidth [31] not only makes the system more efficient and cost-effective, but it also helps to reduce traffic jams in the network.

Real-time analytics Edge devices can analyze incoming data in real time [32], allowing for instant insights to be generated. This technology is utilized in situations where making run-time decisions based on analyzing data is crucial, like when it comes to predicting maintenance needs in industrial settings.

Scalability EC has the flexibility to adjust its capacity [33] to handle different workloads and the number of devices involved. Ensuring scalability is crucial for applications that adopt changes on demand for the number of connected devices. It also helps in efficiently allocating resources.

4.2 Challenges

On the basis of edge architecture and from the existing literature survey we find that EC faces several challenges which are discussed below.

Task migration It is the process of migrating tasks from edge devices to remote cloud devices. When the tasks come for execution on the edge server, the selection of the appropriate edge server and remote cloud server for processing is challenging. A better offloading [34] scheme helps to distribute load adequately among all edge servers to make the system highly efficient.

Privacy and security This challenge is significant due to the distributed and often resource-constrained nature of edge environments. The edge servers are deployed nearby to the data-generating devices, so data privacy [35] and prevention of data from leakage and unauthorized access is a big challenge. Besides, edge devices are generally deployed in open access areas, so the security of network devices during the transmission of data from edge devices to another server is also a big concern. Because of its complex structure, EC is susceptible to malicious attacks related to resource allocation [36]. Attackers with access to edge servers have the potential to steal or alter the information.

Resource management The devices and networks that are connected to provide executed data are heterogeneous in nature and all follow various protocols. The management of these resources [37] is a crucial challenge of an edge environment. We must manage networks and devices properly so that we can have proficient device and network utilization.

Tasks allocation between the cloud and edge The assignment of tasks between the cloud and the edge is a challenge because of the nature of the edge environment [38]. Some tasks are processed locally, and some are transmitted to the cloud so there is a need for efficient task scheduling algorithms which help to make the allocation more reliable and beneficial.

Energy efficiency The edge paradigm is distributed in nature. In real-time scenarios, it executes a number of tasks in a very short time so there is a high consumption of energy. Therefore, new energy protocols are required that help to improve the energy efficiency of the system [39]. Several studies have started in EC, focusing on energy harvesting and wireless charging capabilities [40, 41] in EC. Scheduling resources becomes increasingly challenging when additional energy supplements are incorporated, as it requires the consideration of energy consumption during task transmission, task processing, and the utilization of harvested energy.

4.3 Optimization aspects

As we’ve seen from the challenges being faced by the edge architecture, there is a good need for resource optimization in the related environment. Resource optimization leads to the efficient allocation and utilization of network resources to ensure that critical applications and services receive the sufficient bandwidth needed to operate at peak efficiency [42]. This involves dynamic allocation of resources based on real-time network requirements to achieve the most efficient network investments while minimizing wasteful spending.

Technologies such as SDN, NFV, Network Controllers, and various networking algorithms have been developed to simplify network operations, improve network agility, and reduce costs. It also enables organizations to optimize their network resources. By implementing advanced network security measures, such as micro-segmentation and network slicing, and automating network management tasks, organizations can reduce the risk of human error and free up valuable IT resources for more strategic initiatives.

Therefore, resource optimization is essential for ensuring the productivity and efficiency of computer networks [43]. The various optimization method is given below:

Computation offloading It is the process of offloading computation tasks to computing devices at the edge of the network or in the cloud [46]. This reduces the load on local computing devices and improves performance. EC leverages a wide range of computing devices situated at the periphery of the network, including mobile phones, IoT devices, and network routers, to execute computational operations. On the other hand, cloud computing utilizes remote servers located in data centers to perform computation tasks.

Load balancing Load balancing involves an equal distribution of workloads across numerous computing devices. This ensures that no single device is overburdened and that computing resources are used efficiently. There are many [45] load-balancing algorithms, such as random, and weighted algorithms, that can be used to distribute workloads. Load balancing can be done at various levels, such as the network level, the application level, or the hardware level.

Task scheduling Task scheduling involves the allocation of tasks to computing devices, taking into account their level of urgency and priority. This ensures that critical tasks are completed first and that computing resources are used efficiently. Task scheduling algorithms can be based on different criteria, such as task deadline, priority, or task laxity.

Energy optimization Using power management [44] strategies such as frequency and voltage scaling is one of the most efficient ways to maximize power usage. Depending on the workload, the processor's clock rate is changed by frequency scaling. If the load is light, you can lower the power consumption of the processor by lowering the clock frequency. Voltage scaling lowers the voltage supplied to the CPU, lowering the power consumption of the processor. These processes are often combined to achieve the highest energy efficiency.

Based on these concepts, we performed an intense literature review, divided into different categories which is discussed in the next section.

4.4 Related work

This section aims to provide a review of recent studies on load balancing, resource allocation, task scheduling, and computation offloading. Also, a more comprehensive analysis of several related literature studies is conducted. The technical terms used in EC and Mobile Edge Computing)MEC serve the same purpose and have identical meanings. Furthermore, these technologies consistently maintain their integrity and generality.

MEC is an ingenious technology that can catapult the performance of computing-intensive services like Augmented Reality (AR) applications and IoT by bringing the data processing and computation tantalizingly close to the end-users. This reduces the latency and amps up the QoE for users, breathing new life into the entire gamut of services. To accomplish this, numerous resource allocation plans have been put out in the literature to optimize the use of communication and computational resources [23].

A comprehensive analysis of current research on computation offloading at the edge is conducted in [47]. Various aspects of Computation offloading are discussed, including ways to reduce energy consumption, improve the quality of experience, and ensure quality of service. Furthermore, their review provides valuable information about different approaches to resource allocation.

A thorough overview of task offloading strategies for MEC that have been examined by other researchers was provided by the author [48]. A study of the literature was done on current studies that have been done on task offloading for MEC. Three groups of models have been identified: algorithm paradigm, decision-making, and computational models. The classification and clarification of the architectural model employed in task offloading was another goal of this work. There is also a discussion of the practical use cases, research directions, and design issues associated with this field.

The authors [19] discuss different methods for scheduling resources collaboratively within the framework of EC. The authors also discussed task offloading, resource allocation, and resource providing as three main scheduling research challenges in EC. Also discussed and compared technique for scheduling resources that is both centralized and distributed.

on the basis of these comprehensive studies we are categorizing the Edge optimization in following section

Resource allocation and scheduling Resource Allocation and Scheduling are integral concepts of various decentralized domains including computer networking. It involves the strategic distribution of resources to achieve better resource utilization and get the maximum performance and efficiency of the system. It plays a crucial role in managing tasks, processing data, and workflow.

In the edge environment, there are many Edge Servers available to handle service requests from IoT devices. Every service request can be divided into a set of tasks. To achieve a Service-Level Agreement (SLA), the primary objective of resource scheduling in EC is to identify the optimal approach for allocating various tasks to edge nodes, ensuring efficient completion.

The author [43] has devised an EC-based resource allocation strategy for IoT applications that is nothing short of revolutionary. This cognitive edge framework seamlessly blends dynamic resource allocation with EC resource demand mapping to efficaciously support the enterprise cloud, resulting in a phenomenal outcome. The cloud operating system's bidirectional resource sharing creates a single framework for allocating resources for IoT, leading to a harmonious coexistence of technology and innovation. The test results demonstrate beyond doubt that the proposed technique is far more efficient in handling resource allocation requests, further reinforcing its position as a game-changer in the world of computing.

A unique MEC-based resource allocation strategy for AR applications has been proposed by Ali and Simeone [44]. The plan successfully reduces system costs by combining communication and optimizing IT resources. To ensure optimal energy use in the MEC, a consecutive convex approximation function is used. The results demonstrate that, the method produces superior load reduction capacity as compared to traditional methods.

Another integration of EC and IoT is proposed and describes the cross-layer dynamic network optimization to improve the performance of network setup. The authors [49] used the Lyapunov dynamic stochastic optimization approach for admission control and better resource allocation.

The author of [50] describes the challenges faced by EC regarding resource management and utilization. The author also designed a model for better resource allocation in edge environments by providing a unique key in the system.

According to [51] Wi-Fi access networks have been used to improve MEC services. The results of a resource management strategy based on two-layer time division multiplexing show that MEC over Wi-Fi can reliably use power. It is suggested to use MEC-enabled Fiber-Wireless broadband for low-latency, demanding MEC applications, [52] developed this work. To reduce network latency, a cloud-enabled resource management scheme has been proposed. The results demonstrated that the analytical model was successful in reducing delay and saving energy.

A middleware layer is used by the DRAP solution [53] to connect mobile devices and peripherals. Any mobile device that needs resources can access a pool of resource-rich edge devices. Resource discovery, idle resource computation, and mobile and peripheral functions are all handled by DRAP. This method is very flexible because no device acts as an edge device and because it can automatically reorder as nodes are added or removed.

[42] Proposes to use of a P2P system to overcome the problem of selfish behavior between participating devices. Under the point-based incentive mechanism of this system, devices can share resources to earn points, which they can then use to purchase resources from other devices. The system is built on a pre-trust model and takes into account the social obligations of the community group to ensure that the participating devices are trusted. A middleware framework to improve CPU usage for AR applications is proposed in [54]. The platform provides software services for augmented reality applications and includes functionality to deploy or remove software components at runtime, using only those required for the specific application. While this reduces additional demands on the CPU, the technique only applies to AR applications. some of research article are discussed in Table 3.

Load balancing and distributions Load balancing across edge devices is a critical aspect of EC, which is becoming increasingly important as the use of linked gadgets and IoT devices continues to grow. Instead of processing data in a central cloud, EC processes data at the network's edge, lowering latency and enhancing the network's performance. Edge load balancing is a fundamental factor in ensuring efficient data processing.

One of the key areas of research in edge load balancing has been focused on resource allocation. Researchers have been exploring various methods for resources allocating, such as CPU and memory, to different tasks and applications, to maximize performance and minimize downtime.

CooLoad, a collaborative load-balancing system created by Beraldi [45], is positioned to reduce execution latency at the network edge. Under this collaborative system, data centers exchange cache space based on available space. If the cache at the data center becomes saturated, incoming requests are redirected to another data center that has sufficient storage capacity. This solution has been proven to significantly enhance the efficiency of the information technology department through experimentation.

Load balancing is achieved by assigning VMs to edge devices that have sufficient resources, as proposed in [57]. The primary objective is to reduce the overall travel time while taking into account WAN bandwidth and edge device load. To reduce workload, the method described in [58] integrates mobile devices, edge devices, and remote clouds. A mobile device that needs resources can connect to a nearby edge device or a remote cloud. Low latency and no internet bandwidth are required if service is provided through an edge device. However, if the service is sourced from a remote cloud, it becomes a typical MCC pattern with higher latency due to internet bandwidth usage.

The predefined VM templates are used according to the approach suggested by [59] to meet the user requirements. To keep the infrastructure stable, the edge device selects the best template, customizes the infrastructure, and then rolls back the changes. Using VMs separates infrastructure-level changes from changes to guest operating systems, but load balancing and task sharing between tenant VMs are not covered. Consistent with the concept of mobile EC, Kumar [24] created an intelligent network data management system for a network that can withstand vehicle

Table 3 Related work on resource allocation and scheduling

Author, Year & Reference	Findings/outcomes	Limitations	Research contribution	Results	Optimized parameters
Amjad, 2017 [43]	The primary advantages of EC is the decrease in latency	Edge devices have less processing, storage, and computational capability than centralized data centers	EC based IoT application resource allocation framework	The proposed system manages resource allocation requests more efficiently	Cost, Throughput
Ali and Simeone, 2017 [44]	Edge computing offers improved reliability by reducing dependency on a centralized infrastructure	Managing and growing edge computing infrastructure gets increasingly difficult as edge devices and applications proliferate	Resource allocation for AR applications using energy-efficient MEC	The proposed method achieves better load reduction than the current methods	Energy
Shichao, 2018 [49]	The purpose of utilizing computational resource management is to enhance the overall performance of communication systems	Interference Management, Didn't Show Proper Effective energy harvesting	Mechanism for the cross-layer distribution of dynamic resources	The proposed scheme achieves improvement in performance and throughput	Performance, Delay, Throughput
Jinlai, 2017 [50]	Resource-sharing contracts with edge server providers and employ resource provisioning and latency-aware scheduling	Showing Total dependency on the server in micro data centers	A zenith resource allocation model	A good resource allocation scheme helps to improve performance	Response Time, Performance
Zhenquan, 2019 [55]	Minimize the task execution latency and power consumption's weighted sum cost as achievable.	The results are based on ϵ-greedy Q-learning.	The proposed algorithm for task offloading is an optimal approach that utilizes the ϵ-greedy Q-learning technique.	Low Power Consumption, Low task execution latency	Resource allocation, cost, performance, energy
Rimal, 2017 [51]	Edge devices and servers may use encryption and access restrictions to safeguard data from breaches and illegal access	Specialized frameworks and protocols are needed to handle edge devices, servers, and their interfaces with centralized cloud infrastructure	Use of fiber-wireless (FiWi) access networks for optimizing MEC services	With MEC over WiFi, the proposed strategy has demonstrated adequate power consumption	Delay, Response Time, Energy
Rimal, 2017 [52]	Edge computing may also work in low-connectivity contexts. Applications may share edge computing resources.	Reduced bandwidth costs and efficiency may offset these early expenditures over time.	Arrangements for managing resources within the MEC via a wireless fiber optic network.	Both energy efficiency and quality of experience were improved by the proposed method.	Response Time, Energy &Computation Offloading
Liu, 2020 [56]	Efficient resource allocation with the lowest cost.	Compared with the probabilistic offloading scheme, Heuristic algorithms are not considered.	An iterative optimization algorithm for proficient resource allocation.	Increased resource utilization and Minimized the cost and time consumption.	Cost, Performance

delays. A VM migration strategy is used to reduce data center power consumption. Electric cars placed at the edge of the network handle handling and communication issues while making independent judgments. The test results demonstrate that the proposed solution provides higher throughput and lowest network latency. Some related studies are discussed in Table 4.

Computational offloading and rescheduling Computational offloading is an important concept in resource optimization, it tells how computational tasks are managed for the effective use of resources. It transfers workload from local devices to remote sources and helps to achieve maximum efficiency of the system. This approach is instrumental in resource utilization, enhancing efficiency, and minimizing task execution. It works in various fields from EC and MEC to Cloud Computing. In this section, we discuss various computation offloading schemes in edge environments.

Table 4 Related work on load balancing and distributions

Author, Year & Reference	Findings/outcomes	Limitations	Research contribution	Results	Optimized parameters
Kumar, 2016 [24]	EC lets programs run without a cloud connection	Skills and resources may be needed to maintain a distributed edge network's reliability, security, and efficiency	Data management strategy for MEC Smart Grid with media latency tolerance	The proposed solution increases throughput while minimizing network latency	Energy, Delay, Response time, Throughput
Beraldi, 2017 [45]	EC keeps sensitive data closer to its source and reduces transmission exposure, improving privacy and security	Distributed processing makes data consistency and integrity across devices harder	Collaborative load balancing arrangement at MEC	The performance of IT services is greatly improved by the proposed approach	Time Delay, Average Load
Yaozhong, 2017 [60]	Load balancing Within EC based on QoS parameter	Priority of task are not taken into consideration	Distributing new tasks that come into the EC network at regular intervals	Effective Load Balancing And improved overall performance and output	Makespan and Performance
Tham, 2017 [61]	The objective is to reduce the total processing time of the application	Energy-constrained are not taken into consideration	Presented a load balancing system that is both distributed and capable of achieving quick convergence	Reliable Load Distribution and enhanced productivity and performance all around	Performance, Throughput, Fault tolerance, Reliability.
Babou, 2020 [62]	Distribute the requests hierarchically on the HEC clusters and reduce the latency.	The automation of resource management.	Hierarchical Load Balancing technique	Effective load Balancing with Improved processing time of request and reduced delay	Latency and Performance
Kadhim, 2021 [63]	Load balancing approach to improve utilization of resource	Energy constraints are not taken into consideration	A Load Balancing scheme between Edge servers	Maximizing the CPU utilization, Priority scheduling, Minimizing the required bandwidth	Bandwidth consumption, Latency, Response time, Resource utilization
Zhang, 2020 [64]	Secure load balancing and computation offloading algorithm	As mentioned in the article no security parameters are considered	Load balancing algorithm for distributing mobile data used among small base station	Reduce overall communication cost	Performance, Cost

A game theory strategy has been proposed by Chen [46] to investigate the issue of offloading as it relates to mobile cloud computing. The system creates a distributed multi-user compute offload set among mobile device users to achieve optimal computing in the network. Game engineering successfully achieves the property of Nash equilibrium and reduces the computational load for many users.

For pressure equalization for computational tasks, Laredo [6] presented an important self-organization technique based on the automated sand cell model by Bak-Tang Wiesenfeld. Experimentation has revealed improved utilization of resources without any degradation in the QoS.

A low-latency MEC design in mobile media networks was created by the author Li [66]. The system uses an MEC server and an optimized traffic scheduling mechanism. As a result, a modified handover technique improved mobility management. For task scheduling in MEC systems, A Markov decision method was suggested by Liu [30] that reduces the average latency and power consumption. Comparing the results with reference techniques, there is very little delay.

A computationally offloading approach for MEC systems was created by [67, 68] use of portable energy harvesting (EH) equipment. The model uses a dynamic computational offloading method based on Lyapunov optimization and the outcomes demonstrate that it outperforms benchmark policies.

Various research papers have suggested different advanced computational techniques. [69] proposes a centralized Enterprise Cloud (EC) model, where all mobile and edge devices register with EC, which assigns the appropriate edge device for resource requirements. The advantage is that work can be continued on multiple peripherals, saving money, time, and energy. However, this plan may not be scalable to larger networks because it relies too much on internet capacity. Using a centralized origin server, a similar strategy is provided in [70], where the origin server maintains records of connected peripherals and their services. Requests are forwarded by the origin server to the appropriate edge device, which can perform the task, and distribute it to multiple edge devices. Some related studies are discussed in Table 5.

5 SDN

These days, SDN [93] has become one of the most significant network architectures due to features like scalability, flexibility, and resource monitoring. In addition, the researchers can add innovations for resource optimizations with its integration in IoT [71], cloud computing, and EC.

In SDN, the control plane works like a centralized device known as a controller [72, 73]. There is a lot of benefit to decoupling the control plane and data plane of network devices. The controller is an advanced method for managing and optimizing network resources. It involves separating the control plane, which is responsible for making network decisions, from the data plane, which handles the actual data traffic. That further helps to adequate resource allocation and better load balancing [74].

SDN enables the appropriate distribution and redistribution of network resources according to current demands and traffic patterns, which helps optimize resource utilization. It can optimize the usage of resources by

Table 5 Related work on computational offloading and rescheduling

Author, Year & Reference	Findings/outcomes	Limitations	Research contribution	Results	Optimized parameters
Chen, 2016 [46]	EC processes data locally and responds instantly, enabling real-time decision-making	Installing edge devices and networking infrastructure requires money	Mobile Cloud Offloading Method Using Game Theory	The performance of the proposed unloading system is improved	Average Utilization, Computation Offloading
Laredo, 2017 [65]	Only necessary data is transferred to the cloud or data center by processing and filtering data at the edge	EC processes and stores data closer to its source, lowering cloud-based visibility and management	Important self-organizing method for load balancing in MEC	Better resource consumption is demonstrated by the proposed method without sacrificing QoS	Execution time, Average Load, Load Balancing
Li,2017 [66]	Distributed EC offers horizontal scalability	Configuring, monitoring, and maintaining a distributed edge device network is difficult	MEC design in low-latency mobile vehicle network	Results showed enhanced mobility management with the intercell handover mechanism	Latency, Reliability, Computation Offloading
Liu, 2016 [30]	EC minimizes bandwidth costs by shifting computational activities to edge devices, reducing cloud data transfer	Edge devices and platforms can't communicate without established protocols and interfaces	Task planning using the Markov decision process in MEC systems	Results showed minimum delay compared to baseline approaches	Average Delay, Task Scheduling.
Mao, 2016 [67]	Supporting real-time apps improves response times and user experiences	Cloud or data center resources may be needed for computationally demanding processes or applications	Using energy harvesting is a way to reduce the computational load on MEC	The proposed model outperforms benchmarked policies	Cost, Energy, Completion Time, Computation Offloading
Mao, 2017 [68]	Optimization minimizes network congestion, bandwidth costs, and efficiency	Managing updates, maintenance, failures, and replacements in large-scale edge installations may be difficult	Low complexity suboptimal method for MEC systems	The results show less latency than standard processes	Energy, Delay, Task Offloading, Resource Allocation

dynamically adjusting bandwidth, routing, and network policies through a central controller. SDN is a critical tool in modern network management because it offers flexibility that improves quality of service (QoS), enhances network performance, reduces operational costs, and optimizes resource allocation [75].

SDN architecture [76] comprises three layers, i.e., the Application Layer, Control Layer, and Infrastructure Layer. Figure 6 presents the layers of the SDN Architecture.

Application layer This layer is the open area in SDN where innovative applications can be implemented. Some examples of applications within this layer include Network Automation and Configuration. This layer is responsible for monitoring various systems, troubleshooting any issues, and ensuring QoS.

Control layer The control layer is the intelligent layer that contains the control plane logic. The intermediate layer is responsible for integrating the application and infrastructure layers. It is also used for monitoring business logic to gather and manage different types of network information, statistics, and details about the network's structure. Besides, this layer connects with the application layer through a Northbound interface and connects with the infrastructure layer using a Southbound interface, using the OpenFlow Protocol.

Infrastructure layer The infrastructure layer consists of various network devices that are used to create the data plane and route traffic based on instructions received from the controller. Inside the data center, there are operational network switches.

SDN offers numerous advantages in the network industry, assisting organizations with better solutions. These benefits include cost optimization, simplified troubleshooting, and efficient monitoring of resources. Here are a few advantages of SDN.

SDN benefits SDN brings various benefits to the network industry, like reducing operating costs by promoting virtualization in network environments. It allows for better utilization of resources and ultimately leads to lower operating expenses. Besides, it provides automation to networks to make them more reliable and customizable than ever before. That means SDN-based networks can be further enhanced with automation.

SDN limitations One of the crucial challenges in SDN is security. Since SDN is centralized, all policy-related tasks are performed from the control layer or controller. If someone gains unauthorized access to the controller, it can lead to the entire network being compromised. The DDoS attack is also widely performed in SDN [77]. The current education systems and certification syllabuses of network vendors like Cisco and Juniper do not provide enough information about SDN. This lack of knowledge and training on SDN is causing a shortage of skilled workforce.

5.1 SDN of things

SDN of Things is a novel strategy that effectively manages and controls IoT devices by using SDN and EC [78]. SDN offers centralized network management by isolating the network control layer from the data plane. EC, meanwhile, moves data storage and computation closer to IoT devices. EC is deployed to provide computing power and reduce latency. IoT devices are connected to EC assets that are managed by an SDN controller in an SDN of Thing architecture. This controller provides a centralized view of the Internet of Things and allocates resources dynamically in response to the changing needs of IoT devices. SDN is used to manage data flows between IoT gadgets, EC nodes, and the cloud, enabling dynamic network setup and optimization. IoT devices, SDN, and EC are all part of a network architecture which is known as SDN-IoT.

SDN of Things using EC is an innovative approach to managing and controlling IoT devices that enables organizations to leverage the benefits of both SDN and EC. By providing a more efficient and secure network infrastructure, SDN of Things can help organizations realize the full potential of IoT applications. SDN of Things offers quicker data processing, increased dependability, and decreased latency, which is crucial for IoT applications like continuous surveillance, automation, and control. SDN of Things enables businesses to use network resources more effectively and enhance the network infrastructure at a low cost [79].

One of the key benefits of SDN of Things is the ability to apply advanced network policies and security controls over IoT devices [80]. By centralizing network management, SDN of Things enables organizations to enforce consistent policies across the network which reduces the risk of security breaches and improves the overall network security.

SDN of Things also presents some challenges, such as the need for interoperability, scalability, and the management of distributed resources. Ensuring that IoT devices can communicate with the SDN controller and that data is transferred securely is critical for successful SDN of Things deployments.

5.2 SDN-IoT edge

This section discusses how traditional network protocols can be problematic for network administration, especially given the rapidly evolving nature of computer networks. The SDN can address these challenges by allowing software programs with open APIs to centrally govern networks [4]. However, they note that SDN alone may not be sufficient for fully leveraging the data from IoT applications and network operations. To propose a solution, we put forward an architecture based on SDN-IoT [81] that incorporates virtualization and IoT-specific capabilities in the control plane, enabling tailored control logic for heterogeneous IoT systems. They also explore the potential of integrating edge cloud technologies with SDN to enhance IoT networks, creating an autonomous and agile system. Additionally, we discuss the challenges of transitioning from centralized to distributed cloud solutions and suggest incorporating a knowledge plane and knowledge-bound API into IoT devices to improve data efficiency [8].

SDN is an innovative network management standard that separates the packet processing data plane from the routing processing control plane, creating a dynamic and coordinated system. By enabling applications to monitor network services and traffic, SDN improve resource utilization and information exchange. The SDN controller can respond to fluctuating demands by collaborating with devices, applications, and routers. To cope with the increasing volume of data, robust internet infrastructure is needed, especially given the rapid growth of IoT devices.

In situations where network service demand changes frequently, service providers may look to use a shared network design to access necessary capabilities. Technological innovations like SDN and NFV will allow system resources to be detached, ensuring strict confidentiality between service providers. Innovative and efficient transmission network solutions are actively being developed to keep up with demand.

With the growing use and demand for technology, old technological concepts are becoming obsolete. This is where SDN and NFV come into play, paving the way for new network designs that meet users' and network services' growing demands. These technological concepts allow for efficient resource utilization, flexible network configuration, and improved network performance [10].

5.3 Related work

SDN is a transformative technology that can revolutionize the performance of network-intensive services such as data center management and network virtualization by centralizing control and management functions. This enhances flexibility and agility while improving the overall quality of service for users, rejuvenating the entire spectrum of services. To achieve this, various resource optimization strategies have been proposed in the literature for effective use of network resources and to enhance system performance. This paper explores the core advancements in SDN-based resource optimization schemes, providing a comprehensive overview of the recent developments.

Rahman [82] presented an architecture that utilizes IoT networks and SDN support layers to enable efficient and intelligent industry management. The proposed architecture was evaluated through performance assessments on simulated environments with appropriate configurations.

To reduce network communication costs, Isyaku [83] discussed a flow monitoring strategy that combines node query requests and responses using heuristics-based optimization techniques to keep track of the network. This approach helps to optimize resource allocation and reduce communication costs.

The author of [84] and her team researched attacks on actual IoT devices to emphasize the significance of security in IoT system design. The study revealed current security vulnerabilities in commercial IoT systems, emphasizing the need for essential security features such as reliability, anonymity, privacy, authorization, authentication, recovery of capabilities, and self-organization. Yurchenko [85] built a lightweight Microservices component for OpenNet VM to control network functions (NF) on demand. The potential for interoperability between NFS is made possible by OpenNet VM's support for a high-performance. For packet management, their recommended NF handler takes full advantage of the service chain.

Maksymyuk [86] introduced an industrial 5G touch Internet solution based on SDN NFV. The system utilizes a layered architecture with cloud support and built-in intelligence to automate physical network functions in dynamically created, monitored, updated, and deleted VNFs. The aim is to enhance IoT heterogeneity calculation in industrial settings.

A virtualized SDN-compatible IoT architecture has been described by Antonakakis [87]. They took advantage of the edge SDN/NFV nodes hosting the VNFs. The platform presented by the authors offers several benefits such as low latency and the rapid deployment of IoT devices facilitated by edge devices, thanks to its ability to provide granular user context. The MANO plan is responsible for managing network and infrastructure operations, while the creation of virtual gateways through NFV allows for improved scalability, portability, and speed of deployment. While the theoretical models presented in this article are promising, there is currently no evidence of their practical implementation or testing.

An IoT traffic management prototype using contextual forwarding/processing was discussed by Du [88]. Contextual data delivered from sensor and application layers in IoT networks reduces scalability, security, reliability, and computational limitations. Thus, software-defined data plane services are possible using programmable switches. The FLARE Platform IoT Gateway software provides contextual packet forwarding/processing, IoT device identification, and connectivity.

The authors [89] provide a new technique that connects each cluster of patients using e-health IoT devices to an SDN edge node based on their similarity and distance from each other to facilitate effective caching using spectral clustering. To improve QoS, the suggested MFO-Edge Caching algorithm is considered for taking the near-optimal data alternatives for caching based on evaluated criteria. The suggested method has been shown to improve the cache hit rate and reduce the average time to retrieve data by 76% in experimental settings. The strategy outperforms competing approaches, demonstrating the value of SDN-Edge caching and clustering in maximizing the efficiency of e-health networks.

The authors [90] presented a safe and energy-conscious design for fog computing, and we devised a load-balancing method to maximize efficiency in an SDN-enabled fog setting. A Deep Belief Network (DBN)-based intrusion detection system was built to reduce reduced delays in communication between tasks in the fog layer. Based on simulation results, it was determined that the suggested approach offered a successful technique of load balancing in a fog environment, with improvements of 15% in average reaction time, 23% in average energy usage, and 10% in communication latency over current methods.

The author [38] introduced a novel framework known as software-defined edge cloudlet (SDEC), which is based on RL optimization framework. This framework aims to solve the problem of task offloading and resource allocation in wireless MEC. The recommended strategies for addressing the complex issue include the utilization of Q-learning and cooperative Q-learning-based reinforcement learning techniques. The simulation results indicate that the suggested approach effectively decreases the summation latency by 31.39% and 62.10% in comparison to alternative benchmark approaches, such as classic Q-learning with a random algorithm and Q-learning with epsilon greedy.

The author [91] proposed a load-balancing scheme for machine-to-machine (M2M) networks that takes into account traffic conditions. This scheme utilizes SDN technology. To handle the high demand caused by sudden bursts of traffic, load balancing techniques play a crucial role in M2M networks. By utilizing the power of SDN to observe and manage the network, the suggested load balancing method can meet various quality of service needs by identifying and redirecting traffic.

The author [92] introduced a technology called Fiber-Wireless (FiWi) that can enhance VECNs (Vehicular EC Networks). FiWi technology has advantages like centralized network management and the ability to support multiple communication techniques. The author developed a scheme to reduce the time it takes for vehicles to process their tasks. The scheme uses SDN and load-balancing to offload tasks. This is done in a FiWi-enhanced VECN environment, where SDN helps manage the network and vehicle information centrally. Some of the recent optimization studies in which SDN integrates with EC and their findings, contributions, and results are discussed in Table 6.

5.4 SDN-based IoT-edge management

IoT-Edge management using SDN [93] involves using a centralized controller to manage and orchestrate the network traffic between IoT devices and EC resources. This approach enables organizations to optimize their IoT and EC infrastructure, improving the performance, scalability, and security of their applications

The management of IoT and edge computing infrastructure involves several key tasks, including:

Device discovery and configuration IoT devices need to be discovered and configured on the network before they can be used. This involves assigning IP addresses, setting up security credentials, and configuring device-specific settings.

Resource allocation and optimization EC resources need to be allocated and optimized based on the needs of the IoT devices. This involves monitoring resource utilization [94], identifying bottlenecks, and dynamically adjusting resource allocation based on changing demands.

Network security and compliance IoT devices and EC resources need to be secured and compliant with organizational policies and regulations. This involves implementing security controls, monitoring network traffic, and enforcing security policies [85].

Network monitoring and troubleshooting The network needs to be monitored and troubleshooted to ensure that it is performing as expected. This involves collecting network data, analyzing network traffic, and identifying and resolving issues as they arise.

SDN can help to address these management tasks by providing a centralized controller that manages and orchestrates the network traffic between IoT devices and edge computing resources. The SDN controller can provide a unified view of the network, enabling organizations to optimize their resources and manage security policies more effectively.

5.4.1 IoT-edge management

IoT-Edge management is the art of overseeing devices and applications stationed at the fringes of the network [95]. This includes closely monitoring the gadgets, collating information, and maintaining the applications running on the devices. Edge management plays a crucial role in the world of IoT, as it allows for immediate decision-making in real-time and minimizes the volume of data that must be sent to the cloud for analysis.

Table 6 Related work on optimization in SDN

Year & Reference	Findings/outcomes	Limitations	Research contribution	Results	Optimized parameters
2023, [89]	The performance and reliability of IoT healthcare systems are enhanced by the centralized management and dynamic caching technique. This makes it possible for medical professionals to give patients with better treatment.	If there are any sudden changes in network configurations and settings, IoT devices might have problems working together.	The MFO-Edge Caching algorithm is regarded as an approach for selecting near-optimal data alternatives for caching, utilizing evaluated criteria.	Improved quality of service and minimized the average data retrieval time.	Cache Rate, Delay Time, Performance QoS, and Average time.
2022, [90]	A deep belief network (DBN) helps to reduce the response time and average energy consumption with efficient load balancing.	Improve integration of IoT while integrating with computing technologies like Edge and Fog to reduce workload. Security is also not considered.	SDN-enabled fog environment used for implementation of load balancing algorithms.	Minimized energy consumption and communication delay.	Energy, Execution time, Response Time, Average cost, Performance, and Average Load
2019, [38]	The RL and QL are also used for solving the delay minimization problems.	Security is a main concern while the agents are working on multi-layered architecture.	A resource allocation and computational offloading scheme is proposed in MEC for SDN-based wireless networks.	Minimized delay, improved efficiency, and utilization of the system	Total Delay and Efficiency
2019, [92]	The integration of SDN with fiWi technology is used to enhance vehicle computing-intensive tasks	Lack of adaptability to real-world scenarios and scalability issues.	Better task offloading and load balancing improve the efficiency of the system.	Reduction in process delays by utilizing edge servers and improved resource utilization	Process Delay, Running Time, and Average Utilization
2018, [91]	The capability of SDN improves the system efficiency by monitoring and controlling the network.	The incorporation of Security parameters in such environments is still a challenge	A load-balancing scheme is utilized for SDN-based M2M networks.	Improved QoS of the system and provided benefits of load balancing.	Response Time, CPU workload, and resource utilization.

However, managing the IoT-Edge comes with its own set of challenges. The devices and protocols employed in IoT are highly diverse, ranging from MQTT to CoAP and HTTP. This heterogeneity makes it incredibly difficult to regulate these devices in a cohesive and streamlined fashion. Fortunately, SDN can come to the rescue by providing a unified control plane for the network [96].

By managing the edge, IoT systems can optimize their operations, detect and correct errors quickly, and improve overall performance. Besides, SDN allows for better management of IoT devices and applications, enabling organizations to better leverage their investments in IoT technologies.

5.4.2 SDN for IoT-Edge management

SDN can be used for IoT edge management [97] in several ways. SDN is capable of managing the network infrastructure, including the switches, routers, and gateways that connect the IoT devices. SDN can be used to provision, configure, and manage these devices, facilitating the deployment and administration of IoT edge networks.

Additionally, SDN can manage the IoT applications running on edge devices. SDN can be used to orchestrate the deployment of these applications, ensuring that they are deployed to the appropriate devices and are running correctly. SDN can also monitor the performance of these applications and to make changes to the deployment as needed [98].

Besides, SDN handles the traffic flow between IoT devices and the cloud. It can be used to optimize the traffic flow, ensuring that data is sent to the cloud in the most efficient manner possible. SDN manages the security of the network, ensuring that only authorized devices are allowed to communicate with the cloud.

IoT edge management is critical for ensuring that IoT devices operate correctly and efficiently. SDN can be used to provide a centralized management framework for IoT edge networks, making it easier to deploy, manage, and optimize these networks. SDN can manage the network infrastructure, the IoT applications running on the edge devices, and the flow of traffic between the devices and the cloud servers. With the increasing proliferation of IoT devices, network-edge devices will be managed more effectively with the increasing influence of SDN.

5.5 SDN and NFV

NFV is another significant concept in the context of SDN. NFV is a technology that works well with SDN. It takes network functionalities and puts them into software, making them easier to manage and use. By adopting this approach, it becomes possible to isolate network functions like routing choices from local devices and execute them on distant servers or in the cloud. SDN and NFV [99] have a symbiotic relationship, meaning they can both benefit from each other, but they don't rely on each other to function. Specifically, the virtualization and deployment of network functions can occur independently of SDN, and vice-versa. NFV has the potential to bring numerous benefits, ranging from reducing costs to increasing system openness.

In [6] author introduced a new architecture that focuses on flexible resource management and achieving a balanced utilization of resources. The architecture combines SDN and MEC to allocate bandwidth and computing/storage resources efficiently. The SDN control modules that are present in the cloud/MEC servers allow multiple wireless access networks to work together efficiently and manage the large amount of data being handled. The computing and storage capabilities used at the MEC allow us to meet different quality of service requirements. For example, it is important to have a quick response time for safe and cooperative driving. In the new architecture, different ways of managing resources are designed as an optimization problem. This is done to improve the utilization of computing power, storage, and bandwidth resources. The MEC controller utilizes SDN and NFV regulate modules to efficiently distribute bandwidth resources across multiple access technologies and base stations. This improves the utilization of bandwidth.

The future of communication networks is expected to rely on virtualized infrastructures, which will provide customized services to meet dynamic traffic demands while supporting heterogeneity and diversity. SDN and NFV [100], technologies offer significant benefits, such as reducing operational costs and enabling better resource utilization. They are gaining momentum and are expected to play a critical role in shaping the future of communication networks [101]. The authors [102] propose a resource consolidation scheme that utilizes SDN control features to implement network resource management concepts.

[103] presents a scalable NFV orchestration system that allows for variable distribution of cloud and broadband resources. Also, an SDN-based security architecture for 5G networks is proposed to solve the underlying security problem. SDN-based MEC architecture effectively addresses the challenge of selecting intelligent MEC servers and offloading data.

5.6 Resource provisioning at edge

EC is a viable paradigm for processing data closer to its source, rather than sending it to centralized data centers or clouds, and is resource provisioning at the edge. Advantages of EC include increased reliability, reduced latency, privacy, and security.

Several factors need to be considered for resource provisioning at the edge, including the location and characteristics of the devices, the network connectivity, available bandwidth, and the requirements of the applications running on the devices. For example, devices with limited processing power or storage capacity may require offloading of some of their computational tasks to nearby edge servers/cloud, while applications that require low latency or real-time processing may need to be run entirely at the edge.

There are several approaches to resource provisioning at the edge, including centralized, distributed, and hybrid management. In centralized management, resources are managed from a central location, such as a cloud or data center, and allocated to devices at the edge as needed. In distributed management, resources are managed locally on each device, with coordination between devices to ensure efficient resource utilization. Hybrid management involves a combination of centralized and distributed management, with some resources managed centrally and others managed locally.

There are several challenges associated with resource provisioning at the edge, such as the dynamic nature of edge networks and the need to balance resource availability and energy efficiency. These challenges have led to the development of several resource provisioning frameworks and techniques, such as edge caching, load balancing, and resource prediction. Some notable research works on resource provisioning at the edge are also discussed.

Chen [104] introduced a framework for resource allocation and task scheduling in edge systems. This framework considered energy efficiency and resource availability of the devices at the edge, ensuring that they are put to optimal use.

In the same vein, Zhang [105] introduced a deep reinforcement learning-based approach that strives to optimize the balance between performance and energy consumption in resource allocation for EC. This approach is a remarkable innovation that could potentially revolutionize how resource allocation is done in EC.

Lastly, Sun [106] also made a significant contribution to the field of EC by introducing a load-balancing approach. Their approach takes into consideration the heterogeneity of devices and applications at the edge, aiming to maximize resource utilization while keeping latency to a minimum. These studies highlight the importance of effective resource provisioning at the edge and provide insights into the challenges and opportunities in this area.

5.7 Resource management on edge using SDN

Resource management on the edge is a realm of cutting-edge research that delves into the optimization of resource allocation and utilization in distributed EC environments. In this paper, we shall plunge into the depths of recent developments in resource management on the edge using SDN [107].

EC, a paradigm shift in computing, is a representation of the technological prowess that enables the analysis and processing of data to be carried out close to the data source rather than being restricted to a centralized data center. Nonetheless, managing resources in distributed EC environments can be a formidable undertaking, owing to the ever-evolving nature of edge devices and fluctuating workloads.

SDN is a revolutionary network framework that separates the control plane and data plane of a network. This innovative approach gives network administrators the ability to dynamically and centrally control network behavior. SDN has the capability to address the obstacles associated with resource management in EC by providing a centralized control plane that effectively supervises the network and its resources [6]. This renders SDN the quintessential candidate for resource management on the edge. The following sections explain individually the dynamic resource management, resource allocation aspects, and resource management aspects of SDN..

5.7.1 Dynamic resource management in EC using SDN

A framework for dynamic resource management in EC using SDN was proposed in a recent research paper [105]. The framework uses SDN to dynamically allocate resources to edge devices based on their workload and availability. The framework uses a reinforcement learning algorithm to optimize resource allocation based on the current workload and network conditions. The proposed framework consists of three components: the SDN controller, edge devices, and a learning agent. The SDN controller is responsible for managing the network and resources, while the edge devices perform the computation and storage tasks. The learning agent uses a Q-learning algorithm to learn the optimal resource allocation policy based on the current network conditions.

The proposed framework was evaluated using a simulation environment with multiple edge devices and varying workloads. The results showed that the framework could improve resource utilization by up to 60% compared to a static allocation policy. The framework could also adapt to changes in the workload and network conditions, demonstrating its effectiveness in dynamic environments.

5.7.2 Resource allocation in multi-access edge computing using SDN

Multi-Access EC is an emerging paradigm that enables mobile network operators to deploy compute and storage resources at the edge of the network, closer to the end-users. A recent research paper [106] proposed an SDN-based resource allocation scheme for Multi-Access EC environments. The scheme uses a game-theoretic approach to allocate resources among multiple edge devices while ensuring fairness and efficiency.

The proposed scheme consists of three components: the SDN controller, edge devices, and a resource allocation module. The resource allocation module uses a Stackelberg game model to allocate resources among the edge devices. The model considers the utility of each edge device and the cost of resource allocation. The proposed scheme was evaluated using a simulation environment with multiple edge devices and varying workload. The results showed that the scheme could achieve higher resource utilization compared to a static allocation policy. The scheme could also ensure fairness among the edge devices, demonstrating its effectiveness in a multi-user environment.

5.7.3 Resource management for latency-sensitive applications in edge computing using SDN

Latency-sensitive applications, such as augmented reality and virtual reality, require low latency to provide a seamless user experience. The scheme uses a ML algorithm to predict the latency of each application and allocates resources accordingly.

The proposed scheme consists of three components: the SDN controller, edge devices, and a machine learning module. The ML module uses a Random Forest algorithm to predict the latency of each application based on the workload and network conditions. The SDN controller uses the latency prediction to allocate resources to the edge devices. The proposed scheme was evaluated using a simulation environment with multiple edge devices and varying workload. Some of the proposed algorithms in SDN based EC with their Optimization parameters and platforms are discussed in Table 7 and Comparison between optimized parameters is shown in the Fig. 8.

6 Open issues, challenges, and current trends for the integration of SDN-based edge and IoT

There are a number of unresolved problems, issues, and emerging trends in the integration of SDN-based edge paradigms and other integrated technologies like Cloud Computing(CC) and IoT [108]. Some essential factors to consider in this area are discussed below:

Scalability and heterogeneity IoT-Edge-CC ecosystems are made up of an enormous number of heterogeneous IoT devices, each with a different set of capabilities, protocols, and connection needs [109]. Scalability and maintaining a variety of device kinds are issues when integrating SDN with Edge-IoT. A major issue still lies in ensuring seamless interoperability and effective resource management across such a wide and varied ecosystem [114].

Security and privacy Security and privacy are of utmost importance in the IoT-Edge due to the proliferation of linked devices. To secure sensitive [115] data, SDN-based integration must take into account concerns like authentication, access control, data integrity, and encryption. The vast volumes of personal data that IoT devices create also raise privacy issues, necessitating careful design and deployment of privacy-preserving methods.

*Network latency and dependability*The applications often have severe requirements for network latency and dependability [116]. The dynamic nature of IoT devices, changing network circumstances, and the need for in-the-moment data processing must all be taken into account when integrating SDN with IoT-CC. In a dynamic CC context, achieving low-latency communication, dependable connection, and effective routing techniques is still difficult

Resource management and orchestration It’s crucial to effectively manage and coordinate resources in a Edge-CC ecosystem. SDN may provide centralized management and programmability, but controlling traffic flows, distributing network [117] and computing resources to IoT devices, and optimizing resource use are continuous issues. Critical issues that must be addressed include load balancing, QoS assurances, and dynamic resource allocation.

Interoperability issues arise because Edge, CC and the IoT use a variety of communication technologies, standards, and protocols. The heterogeneity of devices must be addressed in order to provide smooth communication across various platforms, networks, and applications. To provide standard protocols, interfaces, and data models for effective integration, standardization activities are necessary.

The following are current developments in CC and IoT integration using SDN.

EC and fog networking When combining SDN with CC, EC and fog networking paradigms are often used to overcome latency and reliability issues. Therefore, it is possible to minimize data transmission, speed up reaction times, and

Table 7 Summary of the studies proposed algorithms in SDN-based edge computing

Author, Year & Reference	Proposed algorithm	Key concept	Optimized parameter	Platform
Skondras,2019 [108]	FLSA-CC QoS	Task Scheduling, Resource Allocation	Throughput, Waiting Time	Lte-Sim Simulator, iCan cloud, Omnet ++
Zhang, 2020 [109]	Online efficient energy approach	Task offloading, Task Scheduling	Energy, Running time	Realtime
Guo, 2016 [110]	SDN-based TCP Aggregation/ scheduling smart grid framework	Aggregation and Scheduling	Throughput, Performance, Fairness	Smartgrid environment Scenario
Feng, 2014 [111]	Joint allocation model in SDN, Popular flow scheduling policy	Resource allocation and scheduling	Delay time, bandwidth, throughput	Mininet
A.Sharma,2021 [112]	Priority based context information Pr-CAI algorithm	Task Scheduling and Resource scheduling	Delay time, Performance	Omnet ++, Sumo
Chen, 2022 [113]	Design and effective flow scheduling system	Task scheduling, Resource allocation	Cost, throughput, performance, complexity	Realtime

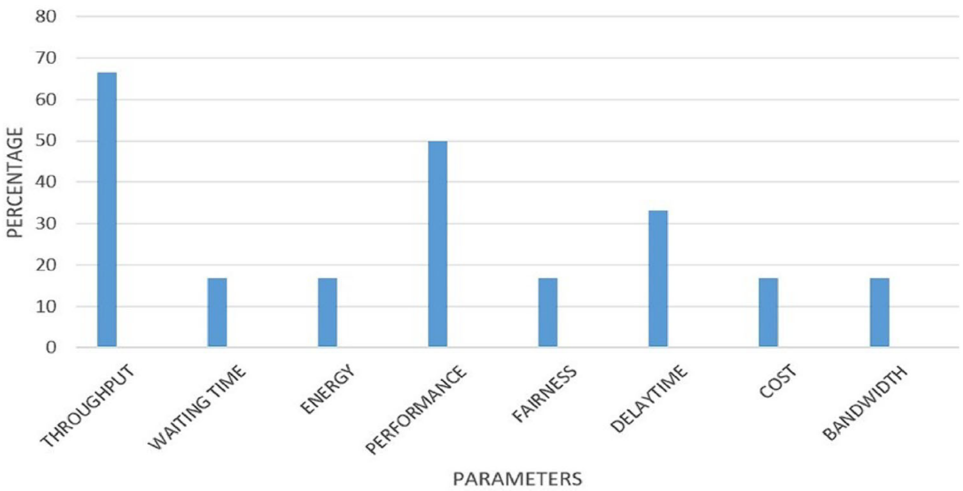

Fig. 8 Optimized parameters

enhance performance by moving processing power closer to the network's edge [118].

Machine learning and AI-driven management It is becoming more popular to use machine learning and artificial intelligence to manage and optimize SDN-based edge networks. To properly manage the dynamic nature of IoT settings, these technologies may allow proactive resource management, predictive analytics, and automated decision-making.

Edge intelligence and analytics It's crucial to conduct data analytics and intelligence at the network's edge. SDN-based edge systems may extract useful insights in real-time, minimize data transfer to the cloud, and improve privacy and security by processing data locally on edge devices.

Edge security using blockchain To improve security, trust, and data integrity in CC, edge and IoT contexts, blockchain technology is being investigated. SDN-based edge systems may provide decentralized authentication, secure data exchange, and transparent transactional records by using blockchain-based solutions.

5G and network slicing The introduction of 5G networks and network slicing tools [119] opens up a lot of possibilities for SDN-based Edge and IoT integration. The performance and flexibility of IoT applications are improved by 5G's increased bandwidth, ultra-low latency, and network slicing, which enables the construction of virtualized networks customized to their unique requirements.

The effective and safe integration of SDN-based Edge and IoT ecosystems will be greatly aided by addressing these open concerns and difficulties while using existing trends.

7 Conclusion

SDN is a promising extension of EC due to its specific management attributes and solutions for the needs of emerging technologies. The present study analyzed some of the new state-of-the-art findings in Edge, SDN, and their integration. We have majorly classified the objectives of resource optimization in edge computing by utilizing load balancing, computation offloading, resource allocation, and task scheduling. Additional categorizations are considering the function of SDN in the optimization of edge computing. The considered methodology defined the magnitude of the problem formulation, particularly for the scenario and underlying assumptions. Besides, this paper presents an analysis of the discussion and research problems, followed by the proposal of three recommendations. Firstly, it is necessary to take into account the gains, limitations, and optimization of resources in both paradigms to effectively employ the end devices. Furthermore, the issue of resource orchestration among multiple providers can be effectively addressed by the implementation of a multiparadigm system that combines SDN with NFV. And lastly, it is crucial to incorporate dynamism into the scenario to replicate true situations of real-time computing systems. It has also been observed that the utilization of SDN-based technology can effectively resolve various unresolved matters pertaining to effective resource allocation in MEC.

Some future scopes of study are mentioned below:

- Resource allocation and load balancing strategies to enhance resource utilization in SDN-based EC could be developed.
- Researchers can implement efficient energy-aware task scheduling methods to reduce power consumption and operational costs in SDN-based EC.

- Several security measures can be considered for robust, efficient SDN-based EC systems to provide a secure architecture with enhanced QoS parameters.

Author Contributions Ajay Nain, Sophiya Sheikh and Mohammad Shahid contributed to the study conception and design of the model. The data acquistion, data analysis,manuscript prepration, and editing are completed by all authors.

Funding Not applicable.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests.

Consent to participate All authors gave explicit consent to participate in this work.

Consent for publish All authors gave explicit consent to publish this manuscript.

Ethical approval All authors have seen and agreed with the contents of the manuscript and are looking forward to publishing this paper on this journal.

References

1. Botta, A., De Donato, W., Persico, V., Pescape, A.: Integration of cloud computing and internet of things: a survey. Future Gen. Comput. Syst. **56**, 684–700 (2016)
2. Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., Nikolopoulos, D.S.: Challenges and opportunities in edge computing. In: 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, pp. 20–26 (2016). https://doi.org/10.1109/SmartCloud.2016.18
3. Nain, A., Sheikh, S.: Software-defined network: an emerging solution for IoT-CC-edge paradigm—an impeccable study. In: Recent Advances in Computing Sciences. CRC Press, New York (2023). pp. 201–205
4. Li, J., Cai, J., Khan, F., Rehman, A.U., Balasubramaniam, V., Sun, J., Venu, P.: A secured framework for sdn-based edge computing in IOT-enabled healthcare system. IEEE Access **8**, 135479–135490 (2020)
5. Alomari, A., Subramaniam, S.K., Samian, N., Latip, R., Zukarnain, Z.: Resource management in SDN-based cloud and SDN-based fog computing: taxonomy study. Symmetry **13**(5), 734 (2021)
6. Peng, H., Ye, Q., Shen, X.S.: SDN-based resource management for autonomous vehicular networks: a multi-access edge computing approach. IEEE Wirel. Commun. **26**(4), 156–162 (2019)
7. Xu, X., Cao, H., Geng, Q., Liu, X., Dai, F., Wang, C.: Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment. Concurr. Comput. **34**(14), e5674 (2022)
8. Hao, H., Zhang, J., Gu, Q.: Optimal IoT service offloading with uncertainty in SDN based mobile edge computing. Mob. Netw. Appl. **1**, 1–10 (2021)
9. Anoushee, M., Fartash, M., Akbari Torkestani, J.: An intelligent resource management method in SDN based fog computing using reinforcement learning. Computing, 1–30 (2023)
10. Kumhar, M., Bhatia, J.B.: Edge Computing in SDN-enabled IoT-based healthcare frameworks: challenges and future research directions. Int. J. Reliable Qual. E-Healthcare **11**(4), 1–15 (2022)
11. Jiang, C., Wan, J., Abbas, H.: An edge computing node deployment method based on improved k-means clustering algorithm for smart manufacturing. IEEE Syst. J. **15**(2), 2230–2240 (2020)
12. Haibeh, L.A., Yagoub, M.C., Jarray, A.: A survey on mobile edge computing infrastructure: Design, resource management, and optimization approaches. IEEE Access **10**, 27591–27610 (2022)
13. Bu, C., Wang, J.: Computing tasks assignment optimization among edge computing servers via SDN. Peer-to-Peer Netw. Appl. **14**, 1190–1206 (2021)
14. Li, C., Zhang, Y., Luo, Y.: Deep reinforcement learning-based resource allocation and seamless handover in multi-access edge computing based on SDN. Knowl. Inf. Syst. **63**, 2479–2511 (2021)
15. Taleb, T., Dutta, S., Ksentini, A., Iqbal, M., Flinck, H.: Mobile edge computing potential in making cities smarter. IEEE Commun. Mag. **55**(3), 38–43 (2017)
16. Raeisi-Varzaneh, M., et al.: Resource scheduling in edge computing: architecture, taxonomy, open issues and future research directions. IEEE Access **11**, 25329–25350 (2023)
17. Du, Jun, et al.: SDN-based resource allocation in edge and cloud computing systems: an evolutionary Stackelberg differential game approach. IEEE/ACM Trans. Netw. **30**(4), 1613–1628 (2022)
18. Jazaeri, S.S., et al.: Edge computing in SDN-IoT networks: a systematic review of issues, challenges and solutions. Clust. Comput. **1**, 1–42 (2021)
19. Luo, Q., Hu, S., Li, C., Li, G., Shi, W.: Resource scheduling in edge computing: a survey. IEEE Commun. Surv. Tutor. **23**(4), 2131–2165 (2021). https://doi.org/10.1109/COMST.2021.3106401
20. Hong, C.-H., Varghese, B.: Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. ACM Comput. Surv. **52**(5), 1–37 (2019)
21. Keele, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
22. Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., et al.: The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ **372**, 71 (2021). https://doi.org/10.1136/bmj.n7
23. Filali, A., Abouaomar, A., Cherkaoui, S., Kobbane, A., Guizani, M.: Multi-access edge computing: a survey. IEEE Access **8**, 197017–197046 (2020). https://doi.org/10.1109/ACCESS.2020.3034136
24. Kumar, N., Zeadally, S., Rodrigues, J.J.P.C.: Vehicular delay-tolerantnetworks for smart grid data management using mobile edge computing. IEEE Commun. Mag. **54**(10), 660 (2016)
25. Dama, S., Sathya, V., Kuchi, K., Pasca, T.V.: A feasible cellularInternet of Things: enabling edge computing and the IoT in dense futuristiccellular networks. IEEE Consum. Electron. Mag. **6**(1), 66–72 (2017)
26. Yu, Wei, et al.: A survey on the edge computing for the Internet of Things. IEEE Access **6**, 6900–6919 (2017)
27. Yu, W., et al.: A survey on the edge computing for the Internet of Things. IEEE Access **6**, 6900–6919 (2018). https://doi.org/10.1109/ACCESS.2017.2778504
28. Yu, W., Xu, G., Chen, Z., Moulema, P.: A cloud computing based architecture for cyber security situation awareness, In:

Proceedngs of IEEE Conference in Communication Network Security (CNS), pp. 488–492 (Oct 2013)

29. Chen, Z., et al.: A cloud computing based network monitoring and threat detection system for critical infrastructures. Big Data Res. **3**, 10–23 (2016)
30. Liu, J., Mao, Y., Zhang, J., Letaief, K.B.: Delay-optimal computation task scheduling for mobile-edge computing systems, In: Proceedings of IEEE International Symposium in Information Theory (ISIT), pp. 1451–1455 (Jun 2016)
31. Sajjad, H.P., Danniswara, K., Al-Shishtawy, A., Vlassov, V.: SpanEdge: towards unifying stream processing over central and near the-edge data centers, In: Proceedings of IEEE/ACM Symposium in Edge Computation (SEC), pp. 168–178 (Oct 2016)
32. Georgakopoulos, D., Jayaraman, P.P., Fazia, M., Villari, M., Ranjan, R.: Internet of Things and edge cloud computing roadmap for manufacturing. IEEE Cloud Comput. **3**(4), 66–73 (2016)
33. Rodrigues, T.G., Suto, K., Nishiyama, H., Kato, N., Temma, K.: Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration. IEEE Trans. Comput. **67**(9), 1287–1300 (2018). https://doi.org/10.1109/TC.2018.2818144
34. Liang, B., Gregory, M.A., Li, S.: Multi-access edge computing fundamentals, services, enablers and challenges: a complete survey. J. Netw. Comput. Appl. **199**, 103308 (2022)
35. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., Zhao, W.: A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J. **4**(5), 1125–1142 (2017)
36. He, X., Jin, R., Dai, H.: Peace: Privacy-preserving and cost-efficient task offloading for mobile-edge computing. IEEE Trans. Wirel. Commun. **19**(3), 1814–1824 (2020)
37. Liu, H., Hua, S., Zhuo, X., Chen, D., Cheng, X.: Cooperative spectrum sharing of multiple primary users and multiple secondary users. Digit. Commun. Netw. **2**(4), 191–195 (2016)
38. Kiran, N., et al.: Joint resource allocation and computation offloading in mobile edge computing for SDN-based wireless networks. J. Commun. Netw. **22**(1), 1–11 (2019)
39. Piovesan, N., Gambin, A.F., Miozzo, M., Rossi, M., Dini, P.: Energy sustainable paradigms and methods for future mobile networks: A survey. Comput. Commun. **119**, 101–117 (2018)
40. Bi, S., Zhang, Y.: Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. IEEE Trans. Wireless Commun. **17**(6), 4177–4190 (2018)
41. Chen, W., Wang, D., Li, K.: Multi-user multi-task computation offloading in green mobile edge cloud computing. IEEE Trans. Serv. Comput. **12**(5), 726–738 (2019)
42. Zhou, B., Dastjerdi, A.V., Calheiros, R.N., Srirama, S.N., Buyya, R.: A context sensitive offloading scheme for mobile cloud computing service. In: Proceedings of IEEE 8th International Conference in Cloud Computing, p. 869876, July 2015
43. Amjad, A., Rabby, F., Sadia, S., Patwary, M., Benkhelifa, E.: Cognitive edge computing based resource allocation framework for Internet of Things. In: Proceedings of 2nd International Conference in Fog Mobile Edge Computing (FMEC), p. 194200, May 2017
44. Al-Shuwaili, Simeone, O.: Energy-efficient resource allocation for mobile edge computing-based augmented reality applications. IEEE Wirel. Commun. Lett., **6**(3), 398401 (2017)
45. Beraldi, R., Mtibaa, A., Alnuweiri, H.: Cooperative load balancingscheme for edge computing resources. In: Proceedings of 2nd International Conference in Fog Mobile Edge Computing (FMEC), p. 94100, May 2017
46. Chen, X., Jiao, L., Li, W., Fu, X.: Efcient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans. Netw. **24**(5), 27952808 (2016)
47. Jiang, C., Cheng, X., Gao, H., Zhou, X., Wan, J.: Toward computation offloading in edge computing: a survey. IEEE Access **7**, 131543–131558 (2019)
48. Islam, A., Debnath, A., Ghose, M., Chakraborty, S.: A survey on task offloading in multi-access edge computing. J. Syst. Archit. **118**, 102225 (2021)
49. Li, S., Zhang, N., Lin, S., Kong, L., Katangur, A., Khan, M., Ni, M., Zhu, G.: Joint admission control and resource allocation in edge computing for Internet of Things. IEEE Netw. **32**(1), 72–79 (2018). https://doi.org/10.1109/MNET.2018.1700163
50. Xu, J., Palanisamy, B., Ludwig, H., Wang, Q.: [IEEE 2017 IEEE International Conference on Edge Computing (EDGE) - Honolulu, HI, USA (2017.6.25-2017.6.30)] 2017 IEEE International Conference on Edge Computing (EDGE) - Zenith: Utility-Aware Resource Allocation for Edge Computing, pp. 47–54. https://doi.org/10.1109/IEEE.EDGE.2017.15 (2017)
51. Rimal, P., Van, D.P., Maier, M.: Mobile edge computing empowered fiber-wireless access networks in the 5G era. IEEE Commun. Mag. **55**(2), 192200 (2017)
52. Rimal, P., Van, D.P., Maier, M.: Cloudlet enhanced fiber-wirelessaccess networks for mobile-edge computing. IEEE Trans. Wirel. Commun. **16**(6), 36013618 (2017)
53. Chen, M., Hao, Y., Li, Y., Lai, C.-F., Wu, D.: On the computation offloading at ad hoc cloudlet: architecture and service modes. IEEE Commun. Mag. **53**(6), 1824 (2015)
54. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case forVM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 1423 (2009)
55. Qin, Z., Qiu, X., Ye, J., Wang, L.: User-edge collaborative resource allocation and offloading strategy in edge computing. Wirel. Commun. Mob. Comput. https://doi.org/10.1155/2020/88671 (2020)
56. Liu, X., Qin, Z., Gao, Y.: Resource allocation for edge computing in IoT networks via reinforcement learning. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). https://doi.org/10.1109/icc.2019.8761385 (2019)
57. Ha, K., Abe, Y., Chen, Z., Hu, W., Amos, B., Pillai, P., Satyanarayanan, M.: Adaptive VM handoff across cloudlets, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Technical Report CMU-CS-15-113, Jun 2015
58. Agarwal, R., Nayak, A.: DRAP: A decentralized public resourced cloudlet for ad-hoc networks. In: Proceedings of IEEE 4th International Conference in Cloud Networking (CloudNet), pp. 309314 (Oct. 2015)
59. Tang, L., Chen, X., He, S.: When social network meets mobile cloud: a social group utility approach for optimizing computation offloading in cloudlet. IEEE Access **4**, 58685879 (2016)
60. Song, Y., Yau, S. S.: An approach to QoS-based task distribution in edge computing networks for IoT applications. In: IEEE 1st International Conference on Edge Computing (2017)
61. Tham, C.-K., Chattopadhyay, R.: [IEEE 2017 IEEE 18th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM) - Macau, China (2017.6.12-2017.6.15)] 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM) - A load balancing scheme for sensing and analytics on a mobile edge computing network, pp. 1–9, 2017. https://doi.org/10.1109/WoWMoM.2017.7974307
62. Babou, C.S.M., Fall, D., Kashihara, S., Taenaka, Y., Bhuyan, M.H., Niang, I., Kadobayashi, Y.: Hierarchical load balancing and clustering technique for home edge computing. IEEE

Access **8**, 127593–127607 (2020). https://doi.org/10.1109/ACCESS.2020.3007944
63. Kadhim, A.J., Naser, J.I.: Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN. China Commun. **18**(2), 271–289 (2021)
64. Zhang, W.-Z., et al.: Secure and optimized load balancing for multitier IoT and edge-cloud computing systems. IEEE IoT J. **8**(10), 8119–8132 (2020)
65. Laredo, J.L.J., Guinand, F., Olivier, D., Bouvry, P.: Load balancingat the edge of chaos: How self-organized criticality can lead to energy- efficient computing. IEEE Trans. Parallel Distrib. Syst. **28**(2), 517529 (2017)
66. Li, L., Li, Y., Hou, R.: A novel mobile edge computing-based architecture for future cellular vehicular networks, In: Proceedinds of IEEE Wireless Communication in Networking Conference (WCNC), p. 16, March 2017
67. Mao, Y., Zhang, J., Letaief, K.B.: Dynamic computation offloadingfor mobile-edge computing with energy harvesting devices. IEEE J. Sel. Areas Commun. **34**(12), 35903605 (2016)
68. Mao, Y., Zhang, J., Letaief, K.B.: Joint task ofoading scheduling andtransmit power allocation for mobile-edge computing systems, In: Proceedinds in IEEE Wireless Communication in Networking Conference (WCNC), p. 16, March 2017
69. Bohez, S., De Turck, J., Verbelen, T., Simoens, P., Dhoedt, B.: Mobile,collaborative augmented reality using cloudlets, In: Proceedings of International Conference in MOBILe Wireless MiddleWARE, Operating System, Application, pp. 4554, Nov 2013
70. Moon, S., Lim, Y.: Task migration with partitioning for load balancingin collaborative edge computing. Appl. Sci. **12**(3), 1168 (2022)
71. Ali, J., Roh, B.: An effective approach for controller placement in software-defined Internet-of-Things (SD-IoT). Sensors **22**(8), 2992 (2022)
72. Ali, J., et al.: A novel features prioritization mechanism for controllers in software-defined networking. Comput. Mater. **69**, 267–282 (2021)
73. Singh, A., Kaur, N., Kaur, H.: Extensive performance analysis of OpenDayLight (ODL) and Open Network Operating System (ONOS) SDN controllers. Microprocess. Microsyst. **95**, 104715 (2022)
74. Ali, J., et al.: ESCALB: An effective slave controller allocation-based load balancing scheme for multi-domain SDN-enabled-IoT networks. J. King Saud Univ. **35**(6), 101566 (2023)
75. Ali, J., Roh, B., Lee, S.: QoS improvement with an optimum controller selection for software-defined networks. PLoS ONE **14**(5), e0217631 (2019)
76. Singh, A., Kaur, N., Kaur, H.: An extensive vulnerability assessment and countermeasures in open network operating system software defined networking controller. Concurr. Comput. **34**(15), e6978 (2022)
77. Singh, A., Kaur, H., Kaur, N.: A novel DDoS detection and mitigation technique using hybrid machine learning model and redirect illegitimate traffic in SDN network. Clust. Comput. **1**, 1–21 (2023)
78. Bera, S., et al.: Soft-WSN: Software-defined WSN management system for IoT applications. IEEE Syst. J. **12**(3), 2074–2081 (2016)
79. Ren, W., et al.: BLLC: A batch-level update mechanism with low cost for SDN-IoT networks. IEEE Internet Things J. **6**(1), 1210–1222 (2018)
80. Galluccio, L. et al.: SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In: IEEE Conference on Computer Communications (INFOCOM). IEEE (2015)
81. Ali, J., Roh, B.: A novel scheme for controller selection in Software-Defined Internet-of-Things (SD-IoT). Sensors **22**(9), 3591 (2022)
82. Rahman, A., Chakraborty, C., Anwar, A., Karim, M.R., Islam, M.J., Kundu, D., Rahman, Z., Band, S.S.: SDN-IoT empowered intelligent framework for industry 4.0 applications during COVID-19 pandemic. Clust. Comput. (2021). https://doi.org/10.1007/s10586-021-03367-4
83. Isyaku, B., MohdZahid, M.S., BteKamat, M., Abu Bakar, K., Ghaleb, F.A.: Software defined networking flow table management of openflow switches performance and security challenges: a survey. Futur. Internet. **12**, 147 (2020). https://doi.org/10.3390/fi12090147
84. Meneghello, F., Calore, M., Zucchetto, D., Polese, M., Zanella, A.: IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT Devices. IEEE Internet Things J. **6**, 8182–8201 (2019). https://doi.org/10.1109/JIOT.2019.2935189
85. Yurchenko, M., Cody, P., Coplan, A., Kennedy, R., Wood, T., Ramakrishnan, K.K.: OpenNetVM, pp. 1–2 (2018). https://doi.org/10.1145/3185467.3190786
86. Aksymyuk, T., Brych, M., Dumych, S., Satria, D., Jo, M.: An IoT based monitoring framework for software defined 5G mobile networks. In: Proceedings of 11th International Conference in Ubiquitous Information Management and Communication IMCOM 2017, pp. 7–10 (2017). https://doi.org/10.1145/3022227.3022331
87. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Arbor, A., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Arbor, A., Invernizzi, L., Kallitsis, M., Network, M., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y., Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J.,Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y.: Understanding the Mirai Botnet. USENIX Security, pp. 1093–1110 (2017). https://doi.org/10.1016/j.religion.2008.12.001
88. Du, P., Putra, P., Yamamoto, S., Nakao, A.: A context-aware IoT architecture through software-defined data plane. In: Proceedings of 2016 IEEE Reg. 10 Symposium in TENSYMP 2016. pp. 315–320 (2016). https://doi.org/10.1109/TENCONSpring.2016.7519425
89. Jazaeri, S.S., et al.: Composition of caching and classification in edge computing based on quality optimization for SDN-based IoT healthcare solutions. J. Supercomput. **1**, 1–51 (2023)
90. Singh, J., et al.: Energy-efficient and secure load balancing technique for SDN-enabled fog computing. Sustainability **14**(19), 12951 (2022)
91. Chen, Y.-J., et al.: SDN-enabled traffic-aware load balancing for M2M networks. IEEE Internet Things J. **5**(3), 1797–1806 (2018)
92. Zhang, ., et al.: Task offloading in vehicular edge computing networks: a load-balancing solution. IEEE Trans. Vehicular Technol. **69**(2), 2092–2104 (2019)
93. Wang, A., et al.: Software-defined networking enhanced edge computing: a network-centric survey. Proc. IEEE **107**(8), 1500–1519 (2019)
94. Stamou, A., Kakkavas, G., Tsitseklis, K., Karyotis, V., Papavassiliou, S.: Autonomic network managementand cross-layer optimization in software defined radio environments. Future Internet **11**, 37 (2019)
95. Wang, Bo., et al.: A reliable IoT edge computing trust management mechanism for smart cities. IEEE Access **8**, 46373–46399 (2020)
96. Bhayo, J., Shah, S.A., Hameed, S., Ahmed, A., Nasir, J., Draheim, D.: Towards a machine learning-based framework for

DDOS attack detection in softwaredefined IoT (SD-IoT) networks. Eng. Appl. Artif. Intell. **123**, 106432 (2023)

97. Mavromatis, A., et al.: A software-defined IoT device management framework for edge and cloud computing. IEEE Internet Things J. **7**(3), 1718–1735 (2019)
98. Eramo, V., Lavacca, F.G., Catena, T., Polverini, M., Cianfrani, A.: Effectiveness of segment routing technology in reducing the bandwidth and cloud resources provisioning times in network function virtualization architectures. Future Internet **11**, 71 (2019)
99. Akyildiz, I.F., Lin, S.-C., Wang, P.: Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation. Comput. Netw. **93**, 66–79 (2015)
100. Moyano, R.F., Fernandez, D., Bellido, L., Merayo, N., Aguado, J.C., de Miguel, I.: Nfv-based qos provision for software defined optical access and residential networks, In: Quality of Service (IWQoS), 2017 IEEE/ACM 25th International Symposium on IEEE, 2017, pp. 1–5
101. Yao, J., Han, Z., Sohail, M., Wang, L.: A robust security architecture for SDN-based 5G networks. Future Internet **11**, 85 (2019)
102. Mitsis, G., Apostolopoulos, P.A., Tsiropoulou, E.E., Papavassiliou, S.: Intelligent dynamic data offloadingin a competitive mobile edge computing market. Future Internet **11**, 118 (2019)
103. Leivadeas, A., Kesidis, G., Ibnkahla, M., Lambadaris, I.: VNF placement optimization at the edge and cloud. Future Internet **11**, 69 (2019)
104. Chen, Y., Ma, J., Huang, L., Zhang, X.: Energy-aware resource allocation and task scheduling in edge computing systems. IEEE Trans. Sustain. Comput. **6**(1), 83–95 (2021)
105. Zhang, H., Yang, W., Xiang, Y., Xiang, Z.: Dynamic resource allocation for edge computing using software-defined networking. Future Gen. Comput. Syst. **110**, 362–372 (2020)
106. Wang, S., Zhang, L., Wang, Y., Zhu, X.: Resource allocation scheme for multi-access edge computing using software-defined networking. J. Netw. Comput. Appl. **176**, 102938 (2021)
107. Balasubramanian, V., et al.: Intelligent resource management at the edge for ubiquitous IoT: An SDN-based federated learning approach. IEEE Netw. **35**(5), 114–121 (2021)
108. Emmanouil, S., et al.: QoS-aware scheduling in LTE-A networks with SDN control. In: 2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA). IEEE (2016)
109. Zhang, J., Cui, Q., Zhang, X., Ni, W., Lyu, X., Pan, M., Tao, X., et al.: Online optimization of energy-efficient user association and workload offloading for mobile edge computing. IEEE Trans Vehicular Technol. **71**, 2 (2022)
110. Guo, W., Mahendran, V., Radhakrishnan, S. et al.: Achieving throughput fairness in smart grid using sdn-based flow aggregation and scheduling. In: 2016 Ninth IEEE International Workshop on Selected Topics in Mobile and Wireless Computing
111. Feng, T., Bi, J., Wang, K.: Allocation and scheduling of network resource for multiple control applications in sdn. China Commun. **12**(6), 85–95 (2015)
112. Sharma, A., Awasthi, L.K., et al.: Pr-CAI: priority based-context aware information scheduling for SDN-based vehicular network. Comput. Netw. **193**, 108097 (2021)
113. Chen, K., et al.: Improving integrated LTE-WiFi network performance with SDN based flow scheduling. In: 2018 27th International Conference on Computer Communication and Networks (ICCCN). IEEE (2018)
114. Tayyaba, S.K. et al.: Software-defined network (sdn) based internet of things (iot) a road ahead. In: Proceedings of the International Conference on Future Networks and Distributed Systems (2017)
115. Kalkan, K., Zeadally, S.: Securing internet of things with software defined networking. IEEE Commun. Mag. **56**(9), 186–192 (2017)
116. Alvizu, R., et al.: Comprehensive survey on T-SDN: software-defined networking for transport networks. IEEE Commun. Surv. Tutor. **19**(4), 2232–2283 (2017)
117. Bera, S., Misra, S., Vasilakos, A.V.: Software-defined networking for internet of things: a survey. IEEE Internet Things J. **4**(6), 1994–2008 (2017)
118. Carvalho, G., et al.: Edge computing: current trends, research challenges and future directions. Computing **103**, 993–1023 (2021)
119. Foukas, X., et al.: Network slicing in 5G: survey and challenges. IEEE Commun. Mag. **55**(5), 94–100 (2017)

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

Ajay Nain received his bachelor's degree from the University of Delhi in Physical Sciences and received his master's degree in computer science from Maharishi Dayanand University. He qualified NET\JRF and currently pursuing his Ph.D. degree in Computer Applications from Lovely Professional University Phagwara, Punjab, India. The area of research is resource optimization in SDN-based networking systems.

Sophiya Sheikh is an Associate Professor at Lovely Professional University, Phagwara, Punjab. She has completed her Ph.D. from the Department of Computer Science, Central University of Rajasthan, India. She did her MCA in Computer Science and Applications from Rajasthan Technical University, Kota, and her B.Sc. in Information Technology from Maharshi Dayanand Saraswati University, Ajmer, India. She is a potential reviewer in various reputed journals like IEEE System Journal, Cluster Computing, Scientific Reports, Journal of Supercomputing, Journal of Cloud Computing, etc. Her research interests are Grid/Distributed Computing, Cloud Computing, and Optimization in Edge and IoT. She regularly writes articles and research papers in reputed National and International magazines and Journals. She is the editor of books published by T&F, INAE, etc. You can contact her at the School of Computer

Applications, Lovely Professional University, Phagwara, Punjab (India) – 144411.

Mohammad Shahid (Member, IEEE) received the M.Tech. and Ph.D. degrees from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India, and the M.C.A. degree from the Department of Computer Science, Aligarh Muslim University, Aligarh, India. He is currently an Associate Professor of computer science with the Department of Commerce, Aligarh Muslim University. He has published many research papers in international conferences and international journals of repute, including Elsevier, Springer, InderScience, Wiley, Taylor & Francis, and IGI Global. His research interests include grid/cloud computing, workflow scheduling, evolutionary computing, meta-heuristic optimization, and portfolio optimization. He was awarded a Research Startup Grant from the UGC-BSR, India, in 2017.

Rohit Malik received his bachelor’s degree from University of Delhi in Physical Sciences and received his master’s degree in computer science from Maharishi Dayanand University. He qualified NET\JRF and currently pursuing his Ph.D. degree in Computer Applications from Lovely Professional University Phagwara, Punjab, India. The area of research is resource optimization in network systems and the integration of blockchain with IoT.