Cluster Computing
https://doi.org/10.1007/s10586-024-04383-w

Modeling the Green Cloud Continuum: integrating energy considerations into Cloud–Edge models

Yashwant Singh Patel[1,3] · Paul Townend[1] · Anil Singh[1,3] · Per-Olov Östberg[2]

Received: 18 October 2023 / Revised: 11 February 2024 / Accepted: 26 February 2024

Abstract
The energy consumption of Cloud–Edge systems is becoming a critical concern economically, environmentally, and societally; some studies suggest data centers and networks will collectively consume 18% of global electrical power by 2030. New methods are needed to mitigate this consumption, e.g. energy-aware workload scheduling, improved usage of renewable energy sources, etc. These schemes need to understand the interaction between energy considerations and Cloud–Edge components. Model-based approaches are an effective way to do this; however, current theoretical Cloud–Edge models are limited, and few consider energy factors. This paper analyses all relevant models proposed between 2016 and 2023, discovers key omissions, and identifies the major energy considerations that need to be addressed for Green Cloud–Edge systems (including interaction with energy providers). We investigate how these can be integrated into existing and aggregated models, and conclude with the high-level architecture of our proposed solution to integrate energy and Cloud–Edge models together.

Keywords Models · Green · Cloud–Edge · Renewable energy · Resource management · Continuum

1 Introduction

Cloud-centric infrastructures have become dominant in modern distributed systems, and have more recently been augmented by the emergence of tiers such as Edge, Fog, and Mist computing, offering valuable benefits to applications such as ultra-low latency, better scalability, enhanced privacy, etc. These tiers allow applications to operate concurrently and seamlessly across geographically distributed federations of resources, including at the network edge, on intermediate fog nodes, and in distant cloud data centres. These infrastructural federations, often referred to as the *Cloud–Edge Continuum*, are seen as the critical computing fabric for modern digital society—indeed, the Europe Commission views the Continuum as a key strategic technology to drive the region's digital transformation [1].

The size of Continuum systems is expanding at an enormous rate, as is the volume of data that they need to handle; some studies estimate that over 50 billion IoT devices will be deployed in the Continuum by 2025, with orders of magnitude more endpoints brought about by 5G/6G systems [2]. Importantly—this results in Continuum systems *consuming huge amounts of energy*; whilst a single hyper-scale Cloud data center may consume over 100MW of power (equivalent to over 80,000 European homes) [3], data centers as a whole are predicted to consume as much as 8% of global electrical supply by 2030 [4]. In total, data centers and networks are projected to collectively consume

Yashwant Singh Patel, Paul Townend, Anil Singh, and Per-Olov Östberg have contributed equally to this work.

✉ Yashwant Singh Patel
yashwant.patel@umu.se; yashwant.patel@thapar.edu

Paul Townend
paul.townend@umu.se

Anil Singh
anil.singh@umu.se; anil.singh@thapar.edu

Per-Olov Östberg
p-o@cs.umu.se

[1] Department Computing Science, Umeå University, Umeå 90187, Sweden

[2] Biti Innovations & Department Computing Science, Umeå University, Umeå 90187, Sweden

[3] Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala 147004, Punjab, India

Published online: 09 April 2024

approximately 18% of the global electrical power by 2030 [5].

This places serious strain on both local and national power grids [6]. Additionally, it carries the potential for significant environmental impact—especially as nearly 80% of world's energy is still generated by *brown* (non-renewable) energy sources such as fossil fuels, which leave a very high *carbon footprint* [7]. Carbon footprint is considered to be the total amount of greenhouse gases, primarily carbon dioxide (CO_2), emitted into the atmosphere during the operations associated with running and maintaining the data center infrastructure. This includes emissions from several sources such as electricity used to power IT equipment, cooling systems, lighting, and other operations. In Continuum systems, carbon footprint can be estimated by considering the electricity consumption and its carbon intensity, which varies significantly based on temporal and spatial factors [8]. For example, on January 31, 2024, the daytime carbon intensity in Germany surpassed that of Sweden by over twenty times. Futhermore, in Germany, the carbon intensity at midnight was 33% lower than that of noon on the same day [9]. These fluctuations are caused by the availability of different power sources across different locations and time periods. The impact of Cloud–Edge systems is thus becoming a major issue as society's energy demands continue to grow, and is an increasingly critical problem for power grids which must balance the needs of Continuum systems against other energy consumers [6]. It is therefore imperative to develop approaches for mitigating, optimizing, and, whenever possible, reduce energy consumption in the Continuum systems.

A promising strategy is the intelligent placement of software tasks within the Cloud–Edge Continuum. Through close monitoring of large federated systems, software tasks can be assigned to the most energy-efficient resources available, considering multiple factors including service demand, availability of resources, QoS (*Quality of Service*) constraints, pricing, etc. For example, tasks can be intentionally assigned to nodes currently powered by sufficient *green* (renewable) energy sources like solar (generated using the solar panels), wind (generated using wind turbines), biofuels (produced from plants, biowaste, agricultural waste and woods), geothermal (utilizing heat from the Earth's core) and hydro (using the kinetic energy of flowing water) etc. or using mix of energy sources depending on the factors such as time of day, season and climate etc. As a further example, Microsoft is collaborating with a Swedish energy company (Vattenfall) to establish a large-scale 24/7 green energy matching system at their new data center. This system ensures that each megawatt hour (MWh) of energy consumed at the data center aligns with an equivalent MWh of green energy produced during the same hour of consumption [10]. Conversely, tasks can be relocated from nodes in geographical regions with high energy demand to improve local power grid availability for other users and businesses, thereby achieving a more balanced load on regional and national power grids. These scheduling decisions can also incorporate energy pricing alongside user and software service-level objectives (SLOs). For example, tasks demanding ultra-low latency may be placed to nearby edge devices or a local data center for processing, even if they rely on non-renewable energy sources, while less latency-sensitive tasks may be scheduled in more sustainable locations.

An example of this is shown in Fig. 1, where the Continuum comprises of four distinct regions (R_1, R_2, R_3, and R_4) and energy providers (EP_1, EP_2, EP_3, and EP_4) which are geographically distributed in different areas. Each region contains Cloud and Edge data centers denoted by *DC* and *E* respectively, users, an energy provider, and IoT devices represented by blue dots. Initially, certain user tasks T_1, T_2, T_3, and T_4 are assigned to region R_1. However, as the availability of green energy in R_1 reduces to 10%, a decision is made to relocate tasks T_2 and T_3 to another region R_2, where the green energy availability is close to 80%. This strategic shift allows us to optimize the utilization of renewable energy and also balance resource loadings across federated regions of the Continuum.

Energy-aware task migration may initially appear to be a straightforward process, but in production environments it can become extremely complex; effective placement requires intelligent decision-making while taking into account multiple factors including energy providers, energy policies, energy pricing, resource availability, SLO arbitration, etc. This is further exacerbated by the dynamic nature of Cloud–Edge environments, which are highly dynamic, mobile and complex, and above all seen as critical infrastructure that should not suffer from serious disruption.

It is therefore vital that new algorithms, mechanisms and methods to improve energy utilisation in the Cloud Continuum are *grounded on formal scientific models that identify and support the huge range of providers, heterogeneous components, interactions, stochastic properties, (potentially contradictory) service-level agreements, pricings, and contractual requirements* present in both energy and Cloud–Edge systems. The use of formal models not only encourages researchers to take into account all necessary components in a highly complex system, but also facilitates validation through mathematical proofs and simulation.

In the literature, conceptual models have been presented utilising techniques such as mathematical models (e.g., mixed integer programming, heuristics, game theory, etc.),

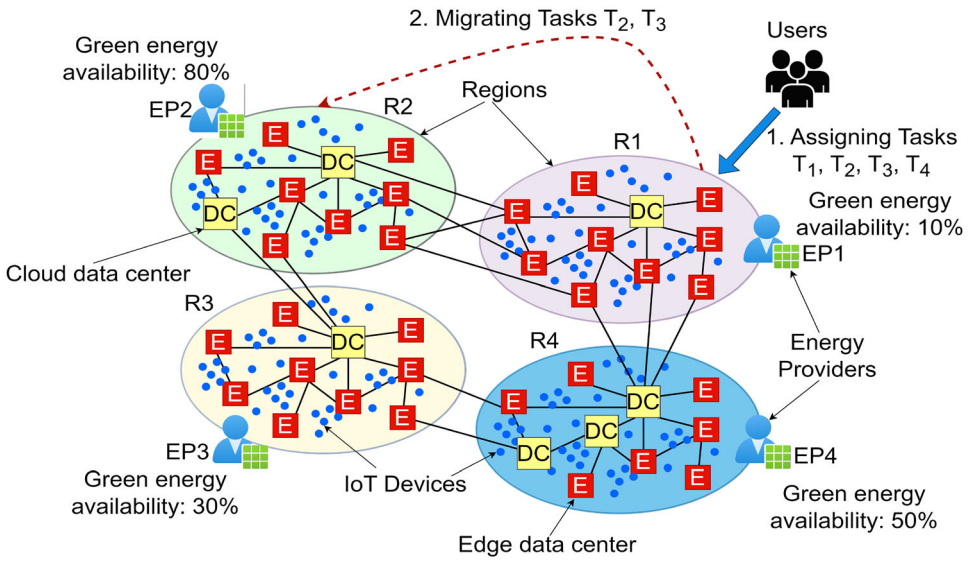


Fig. 1 Migrating tasks in a Continuum system

artificial intelligence models (e.g., machine learning, deep learning, reinforcement learning, etc.), and system and control theory-based models (e.g., Lyapunov-based optimization, Markov decision, fuzzy theory, etc.).

However, *few formal models of federated Cloud–Edge systems exist—and none adequately represent and integrate energy considerations (e.g. multiple providers, renewable energy sources, pricing, and the need to balance consumption over large areas with other non-Cloud consumers, etc.)*. This lack of models is a particular concern when developing autonomous management systems; manual approaches are no longer feasible [11], but existing management mechanisms do not consider energy constraints, policies, and optima across large federations.

In earlier work [12], we discuss a number of formal modeling techniques for Cloud–Edge systems, and present several initial challenges to designing energy-aware Cloud–Edge systems. This paper conducts a much more detailed systematic analysis of current scientific models for Cloud–Edge systems, and—crucially—*identifies the research gaps that need to be addressed to integrate energy considerations into such models*. We describe the key energy-considerations for long-term viability in the rapidly evolving landscape of Cloud–Edge driven systems, and conclude by proposing a high-level architecture and research approach for improving the energy-efficiency and sustainability of federated Cloud–Edge Continuum systems, alongside plans for future work. The key contributions of this work are as follows:

- Presentation of the most relevant scientific models for Cloud, Cloud–Edge, and federated Continuum systems.
- Identification of significant gaps in existing literature in the context of energy-aware Cloud–Edge system models.
- Identification of the key research challenges to integrating energy considerations into models of multi-provider Cloud–Edge infrastructures.
- Introduction of a high-level architecture and research approach for modeling energy-aware Cloud–Edge systems.

The rest of this article is organized as follows. Section 2 presents an overview of modeling Cloud, Cloud–Edge, Fog computing, Mist computing and federated Continuum systems. Then, Sect. 3 provides the current research status of relevant energy-aware Cloud–Edge models. Section 4 discusses research challenges in the development of formal models. Section 5 focuses towards the key energy-considerations for modeling the green Cloud–Edge Continuum and Sect. 6 presents a high-level model to resolve key omissions. Section 7 concludes with future research opportunities in the context of energy-aware Cloud–Edge Continuum modeling and simulation.

2 Background: modeling cloud systems

The architectural landscape of Cloud and Edge systems has evolved rapidly over time, transitioning from “traditional” non-federated Cloud systems to Cloud–Edge architectures, and eventually advancing to federated Continuum systems. It is beneficial to first explore the conceptual nature of these approaches before investigating the modeling behind them.

2.1 Traditional cloud systems

In a non-federated ”traditional” cloud system, a single cloud service provider typically manages one or more geographically dispersed data center sites. A typical geo-

distributed cloud data center environment [13] integrating a single cloud service provider, multiple end-users, and several energy sources is shown in Fig. 2.

Here, the cloud service provider manages several geographically distributed data center sites (DCs). To offer services and resources to cloud consumers, each DC is linked to a backbone network and makes use of a variety of energy sources such as the commercial grid (brown energy) and green energy sources, networking and power equipment, and other devices. DCs can also control how much energy they use; reducing this lowers their energy costs and carbon footprint. For instance, a data center may use either traditional resources, such as the electricity grid, single or combination of green energy sources, such as solar panels and wind turbines. Additionally, data centers may also have installed diesel generators to address power outages and anomalies. Another important component within this environment is the cloud user, who submits service requests in the form of several parameters such as instance type, storage, reservation time, start-time, end-time, etc. [14, 15]. The core of cloud computing lies in the principle of on-demand resource provisioning. This involves leveraging virtualization for on-demand application deployment and employing resource provisioning to effectively manage software and hardware in data centers [16].

In traditional cloud systems, it is crucial to minimize energy usage, carbon emissions and capital costs while ensuring safe and reliable data center operations. To do this, diverse metrics need to be collected—including metrics for IT equipment, cooling systems, temperature control, site selection, building structure, power supply and distribution systems, etc. [17].

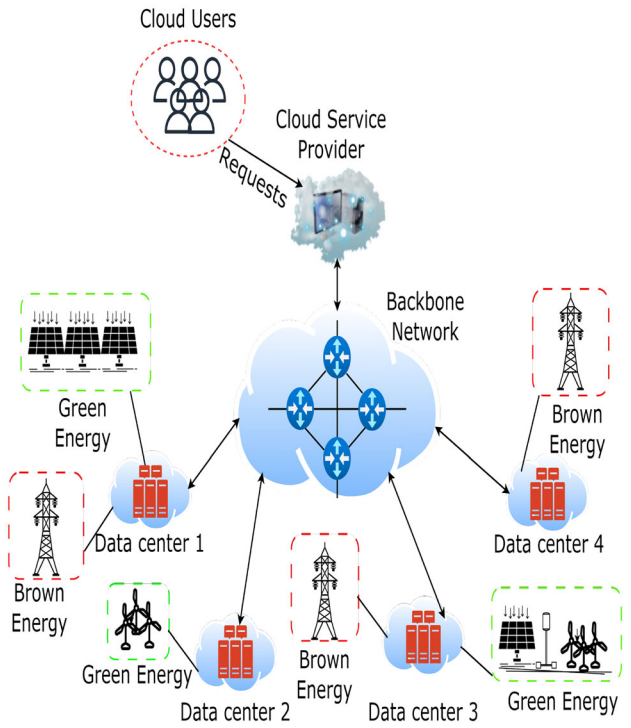


Fig. 2 Non-federated cloud systems

2.2 Energy metrics for cloud systems

A wide variety of energy metrics are available to provide insights into potential inefficiencies, focusing on the key performance indicators of Cloud systems. These metrics enable operators and architects to measure the performance and impact of changes injected in subsystems. Reddy et al. [18] present a detailed study of available metrics for data centers, covering aspects from the power grid to service delivery. From this study, we identify the following key areas alongside examples of useful metrics:

- Energy-efficiency metrics: These metrics are applied for quantitatively evaluating the energy efficiency of a data center and its components; some assess how effectively a data center transfers power from the source to the IT equipment, whilst other metrics identify IT load versus overhead. For example, Power Usage Effectiveness (PUE - the ratio of the total energy consumption of the data center to the energy consumed by the computing hardware within a data center) and Data Center Performance Per Energy (DPPE).
- Greenness metrics: These metrics quantifies the carbon footprint of IT equipment and data centers. They also help to assess the green energy usage, the amount of energy transferred for reuse, and the efficiency of water usage in data centers. For example, Carbon Usage Effectiveness (CUE) and Energy Reuse Factor (ERF).
- Cooling metrics: These metrics measures the efficiency of HVAC systems and their effectiveness in serving cooling demands. For example, Data Center Cooling System Efficiency (DCCSE) and HVAC System Effectiveness (HSE).
- Thermal and Air management metrics: These metrics ensure the temperature issues, effective air flow and aisle pressure management. For example, Relative Humidity and Air flow efficiency.
- Performance metrics: These metrics evaluate the productivity of data centers, their effectiveness in delivering services, and their agility in adapting changes. For example, Data Center Productivity (DCP) and CPU usage.
- Storage metrics: These metrics monitor storage operations and performance and help gain better insight into how effectively storage capacity is utilized. For example, Low-cost Storage Percentage (LSP) and Overall Storage Efficiency (OSE).

- Network metrics: These metrics provide insights into the data center network energy efficiency, traffic demands and utilization. For example, Communication Network Energy Efficiency (CNEE), and Network Power Usage Effectiveness (NPUE).
- Financial impact metrics: These metrics quantify financial implications including data center outages, total cost of ownership, and return on investments in management technologies and tools for sustainable data centers. For example, Carbon Credit (CCr) and Operational Expenditure (OpEx).
- Security metrics: These metrics provide continuous monitoring of virtual physical servers and clouds to protect against attacks. Additionally, they include elementary measurements of firewall performance. For example, Vulnerability Exposure (T) and Firewall Complexity (FC).

2.3 Edge and fog systems

The traditional cloud system architecture has numerous drawbacks, including latency issues arising from a data center's distance from end users, and the need for a single data center to handle potentially massive numbers of users and network connections. Certain applications with strict communication latency restrictions, such as Ultra-Reliable Low Latency Communications (URLLC) and Enhanced Mobile Broadband (eMBB) services, which have a unit millisecond delay requirement, are not suited for the traditional cloud approach. To deliver comparable services with lower latency, edge and fog computing models play a crucial role [19, 20].

Figure 3 illustrates a basic form of a non-federated Cloud–Edge system, where processing of client tasks is performed at the data source rather than on a centralized server or in the cloud layer [21]. The centralized cloud layer can be leveraged for long-term storage and processing of tasks that are generally less time-critical. Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) are emerging as innovative techniques to design, build, and operate networks. These two technologies facilitate the agility, network management capabilities and seamless transfer of data between edge nodes and cloud data centers. At the edge layer, edge nodes serve as gateways and perform data capturing services with the capability to process raw data, such as performing real-time tasks like aggregation, filtering, encryption, and encoding of local data streams [22]. This layer serves as the distribution point for cloud resources, where processing of client tasks is performed at the data source rather than on a centralized server or in the cloud layer. In the edge layer, computing resources such as processors, storage, and networking capabilities are located at the edge of the network to move the burden of processing and storing service and device requests closer to the proximity of the original data source.

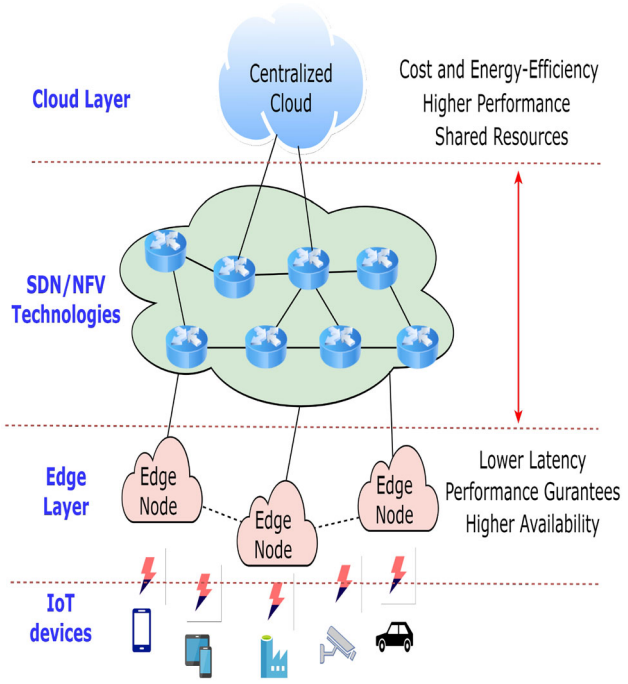


Fig. 3 Non-federated Cloud–Edge systems

As devices have evolved to become computationally powerful and smaller in size, a new paradigm of Cloud computing has been proposed - *Fog Computing*. Fog computing is a decentralized computing framework that brings computation closer to the users, typically located between data generation source and cloud data centers [23] and hence constitutes a layer between traditional cloud and the network edge.

As shown in Fig. 4, the lowest layer represents users, also known as *data generation sources*. The middle layer represents the fog layer, which is in close proximity to users and hence offers less communication delay while communicating within the layer. The fog layer may be used for processing real-time applications, caching, and handling data nearer to the source [24]. The top-most layer represents the cloud layer, consisting of hyperscale data centers used for big data analytics, data warehousing, application hosting, etc. Clouds within this layer are usually far from users, typically resulting in higher network latency [25]. The Fog computing paradigm provides computation, storage and data processing to the users, and is intended to offer a wide range of services; this includes supporting applications with real-time constraints, saving network bandwidth through pre-processing of data at the fog layer, and segregating applications or tasks based on

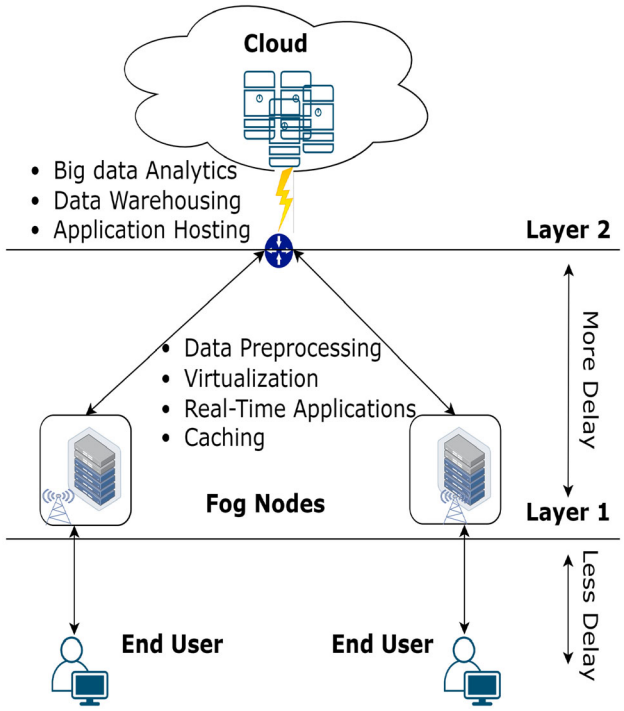


Fig. 4 Fog computing

their requirements and distributing them among cloud and fog nodes to provide users with a better QoS level. Fog computing thus brings forth low latency (critical for real-time applications), mobility support, heterogeneity, edge location awareness, etc. The fog layer also assists in augmenting the edge and cloud layers; the fog nodes situated in-between edge nodes and clouds can coordinate with both of them to help provide better user experience.

2.4 Mist computing

The Fog and Edge computing paradigms bring computation closer to users and assist in the reuse of edge resources. A further evolution (and decentralisation) of these concepts has been proposed as *Mist Computing* [26]. As stated by NIST [27], Mist computing is situated at the extreme edge of the network and can be considered as a light-weight category of Fog Computing. Mist computing aims to utilise computation and storage at the edge device level [28]. The Mist layer consists of low powered computing nodes equipped with sensors and actuators; these nodes can be exploited for computation and storage to maintain QoS constraints in time-centric applications.

As shown in Fig. 5, the Mist layer is at the bottom of computing technology architecture. This layer consists of IoT nodes and is responsible for time-centric data processing. Response time is very critical in real-time applications and Mist layer performs rule-based pre-processing

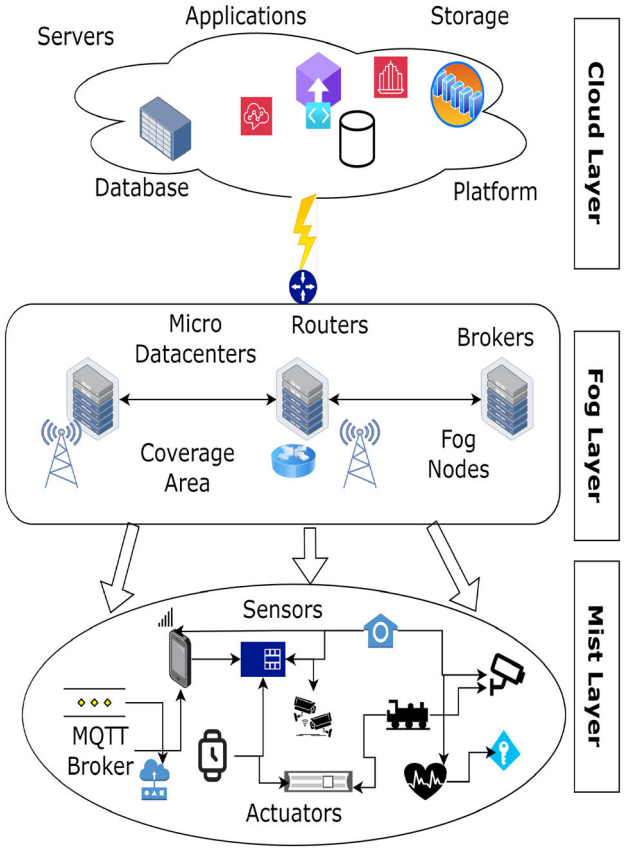


Fig. 5 Mist computing

on data generated by sensors so that data processing can be accelerated, further helping in faster response time. Rule-based pre-processing includes data fusion, aggregation and filtering. Mist layer also aid in reducing the network traffic by eradicating most of the data at the source. The Mist layer mostly consists of sensors, actuators and very light computing resources. Various methods have been developed to enhance messaging at this level; for example, the MQTT (Message Queuing Telemetry Transport) [29] broker in the mist layer is an extremely simple and lightweight messaging protocol which can be very effective in low bandwidth and unreliable networks. The fog layer has more powerful computing resources (micro data centers) than the mist layer, and works as an intermediary between mist and cloud layers.

2.5 Continuum systems

Non-federated paradigms (Cloud–Edge, Fog & Mist) are an effective method to manage device latency; however, this approach is still relatively inefficient if resources are "siloed" - for example, if a fog node is saturated with tasks, there is no obvious mechanism for offloading to other local fog nodes etc. Continuum systems aim to

address this issue, creating a federated and loosely-coupled architecture whereby tasks can be scheduled, monitored, and offloaded as necessary, potentially across different providers [30]. Continuum is a logical progression of the previous paradigms, and brings together a system with central clouds, intermediate fog nodes, and far edge nodes capable of mist computation. It differs from earlier federated approaches (such as [31]), due to a much heavier focus on spatial location and heterogeneous physical resources. As shown earlier in Fig. 1, it extends the conventional large-scale facilities, transforming into heterogeneous, and distributed federations of edge devices and cloud data centers [32], frequently positioned at the network's edge. According to Townend et al. [33], Cloud–Edge Continuum systems possess several distinguishing features that differentiate them from conventional Cloud–Edge architectures. These include: (i) multiple disparate infrastructure providers, (ii) resource constrained devices, (iii) platform heterogeneity, (iv) infrastructural dynamicity, and (v) secure orchestration over public networks. These features significantly increase the *complexity of managing Cloud–Edge Continuum infrastructures and the devices and applications utilizing them*.

2.6 Autonomous resource management and the need for formal models

To handle the massive complexity of Cloud–Edge Continuum infrastructures, we need autonomous resource management, which can ensure the efficiency, performance, and stability of Cloud–Edge Continuum systems. Autonomous resource management plays a crucial role in the effective functioning of Cloud–Edge systems, especially to deal with intrusions (abnormal activity, potential security threat), faults (hardware failures, software bugs, or network issues), performance (application requirements, user demand, and system health), scalability and elasticity (scale resources up or down based on demand), monitoring/audit (tracking changes, investigating incidents, and ensuring compliance with security & regulatory requirements), and other operational challenges.

One particularly important need of automatic resource management is to manage energy in such systems. It not only contributes to cost savings but also facilitates the integration of green energy sources, reduces environmental impact, and enhances overall resilience to power outages and unpredictable energy demands. Autonomous arbitration and multi-objective optimization are well-established fields within computer science, but the concept has not been applied before at the interface between energy systems (with temporal properties) and federated Cloud–Edge resource management systems. The solutions need to be grounded in formal models that can be used to ensure every component is considered. This involves validating approaches through formal proofs and simulations, ensuring the reliability and robustness of the proposed solutions.

In the context of the data center, fog, and edge, several formal models have been proposed for applications that need computation in different layers. These modeling solutions assist in understanding the behavior, performance, stochastic properties, scientific workflows and resource management in such complex systems. Therefore, it is crucial to explore the current modeling solutions before identifying the energy considerations for Green Cloud–Edge Continuum.

3 Current research status of Cloud–Edge Continuum modeling

In recent years, several new approaches have been introduced to model resource distribution across the Cloud–Edge Continuum. This section discusses an overview of the most relevant works available in the literature, and investigates models (workload models, non-federated Cloud–Edge models, federated Cloud–Edge models, and energy-aware Cloud–Edge models) from both technological and architectural perspectives.

Our research performs a systematic review of diverse articles aimed at understanding the current state of the energy-aware Cloud–Edge Continuum. This analysis comprises references from the articles published between the year 2016 and 2023. The framework and methodology adopted in our study draw inspiration from the systematic literature review (SLR) procedure as outlined by Kitchenham [34]. The content within this paper has been aggregated from several sources, including Springer Link, IEEE Xplore, Elsevier, ACM Digital Library, and some additional resources such as Scopus, Google Scholar, and electronic scientific research databases. Specifically, in this section, we report some very relevant works on these types of systems and analyse the research gaps in terms of research focus, Continuum coverage, formal model, energy model, optimization objectives, type of technique, evaluation mode, and type of application.

3.1 Cloud–Edge workload models

In the context of distributed applications, *workload* is interpreted as the overall count of incoming requests dispatched by clients to an application [21]. Influenced by distinct attributes and diverse perspectives, workload can be classified into: (i) random workloads/non-sequential or sequential, (ii) non-transactional or transactional workloads, (iii) data-intensive, memory-sensitive, or computation-intensive workloads [35, 36]. Serving each workload

necessitates the utilisation of a distinct set of resources, in terms of volume and type of resource. Understanding the behavior of workloads is therefore very advantageous when aiming to enhance the reliability and performance of applications and the overall performance, sustainability and efficiency of data centers. For this purpose, significant efforts and advancements have recently been made in workload modeling, especially for modeling real workloads in large-scale systems such as Netflix [37], Facebook [38], Google [39, 40], and Wikimedia [41].

Additionally, there has been a recent surge in demand for forecasting workload behavior [40, 42]. Due to the large number of IoT devices and mobile users, Cloud–Edge applications may encounter unpredictable variations in their workloads; by acquiring advance knowledge of the temporal and spatial distribution of future workloads, systems can proactively adjust resources to promptly address the real-time resource demands of applications and services. From a research perspective, workload modeling and forecasting is therefore crucial for dynamically reallocating available resources to meet SLAs while optimizing energy consumption and reducing costs [14].

Considering the wide spectrum of workload interpretations and objectives, Calzarossa et al. [43] investigate the characteristics of workloads linked to mobile devices, social networks, web, video streaming services and cloud infrastructures. Their work studies distinct workload features and introduces modeling techniques for their characterization, with workload models applied to scenarios including capacity planning, content distribution, provisioning tasks, and performance evaluation. Another recent survey by Duc et al. [21] explores machine learning-based schemes for workload modeling. This survey discusses different methods of workload analysis and prediction, including classical approaches such as Q-learning, reinforcement learning, Markov models, and Bayesian methods as well as recent approaches of complex graph analysis and deep neural networks.

This ongoing work into workload characterization and modeling highlights the importance of integrating workload models into any proposed solution for energy-aware Cloud–Edge systems. For this reason, our proposed solution later in this work integrates workload identification, characterisation, and quantification (based on resource consumption, duration, network, and energy characteristics) into the the 'Application Manager' component (shown in Fig. 13).

3.2 Non-federated Cloud–Edge models

For non-federated Cloud–Edge systems, several models are presented for offloading applications and managing resources between the constrained edge and distant cloud data center. Rahmanian et al. [44] attempt to develop a tool named as 'MicroSplit' for efficient splitting of microservices. Initially, this tool analyses the possible dependencies between the microservices, and applies the Louvain method to split the microservices between the two layers of Cloud–Edge. The authors test its performance in multiple Cloud–Edge settings and improve latency with a reduction in mean response time. To address real-time performance and security issues of tasks, Singh et al. [45] design a scheduling algorithm 'RT-SANE'. Through extensive experiments, they show that the algorithm attains a higher "success ratio" in comparison with existing approaches.

To manage the dynamic allocation of resources and services in the Cloud-to-Edge Continuum, Tusa et al. [46] provide a unified resource management approach comprising both cloud data center and network resources. Their goal is to reduce the 'silo-effect' (isolation of functionalities, resources and services) and provide 'end-to-end slices' (comprising compute, storage and network slices) to perform orchestration of heterogeneous resources and user specific on-demand services ensuring security, isolation and optimized performance. To maintain the trade-off between QoS levels and required computational resources of microservices, Fu et al. [47] design a run-time system called 'Nautilus'. The system is composed of a communication-aware microservice mapper, a load-aware scheduler, and a resource manager. Through experimental results, it is shown that compared to traditional cloud systems, Nautilus minimizes computational resource and network bandwidth usage significantly while ensuring the necessary 99 percentile latency. To deploy latency-critical services in a private Cloud–Edge environment, Ascigil et al. [48] develop uncoordinated resource allocation schemes. Specifically, the authors propose a centralized algorithm to model the QoS requirements of latency-critical services considering user response deadlines.

Pop et al. [49] present a fog computing platform-enabled reference framework for Industrial IoT applications, offering both service and resource management. This is based on deterministic networking and virtualization to promise interoperability along with security. Etemadi et al. [50] design a centralized approach to resource orchestration in a simulated environment which enables deep learning to perform resource auto-scaling at run-time. Ullah et al. [51] design a mechanism named 'MiCADO' for the orchestration of applications in Cloud–Edge environments. They implement a real solution with case studies in the areas of video processing and healthcare.

3.3 Federated Cloud–Edge models

In the direction of federated Cloud–Edge models, Kar et al. [52] present a survey of offloading techniques in federated

(Continuum) systems. Their study also provides an analysis of recent research into applying traditional optimization and machine learning approaches to federated Cloud–Edge systems. Soumplis et al. [30] identify critical resource allocation challenges in the integration of edge, fog, and cloud systems, presenting a heuristic and ILP-based technique for workload placement in the Continuum. Through simulation, they postulate that the resulting mechanisms effectively meet administrator-set objectives, utilising the processing power of the resources at various resource layers (edge, fog, and cloud), and reducing latency at the expense of higher cost. Silva et al. [53] review the applicability of incorporating context awareness to enhance IoT data sharing across Edge and Cloud. The article provides a general overview of the needs of various IoT contexts and updates solutions that take context-awareness indicators into account to deliver operational gains, such as reducing latency and energy usage. To establish directions for future study, the authors demonstrate that although context awareness is important in IoT contexts, its integration to enable more dynamic IoT environments is still limited.

With an emphasis on container-based orchestration and fog-enabled architectures, Svorobej et al. [54] evaluate different orchestration methods throughout the cloud-to-thing Continuum. Kampars et al. [55] investigate application layer protocols that can be applied for communication between the cloud, edge, and IoT levels. To create and manage the mobile-Cloud–Edge computing Continuum, Baresi et al. [56] suggest the A3-E prototype architecture, which supplements functionality offered by FaaS platforms. Their results indicate that A3-E is capable of deploying microservices and significantly reducing latency and battery consumption. Son et al. [57] suggest dynamic resource provisioning strategies for latency-aware Virtual Network Function placement in distributed Cloud–Edge systems. Their work assigns latency-sensitive services between cloud and edge to ensure desired QoS levels.

A number of federated frameworks have also been developed in the industry, such as Zadara, BEACON, and Kubefed etc. Zadara's federated program [58] enables service providers to manage edge computing and administer distributed clouds and supply computing resources close to users with minimal propagation latency. BEACON [59] manages the automatic deployment of applications and services across federated cloud infrastructures. Through a centralized API, Kubefed [60] enables the management of multiple Kubernetes clusters. The objective is to make multi-geo application deployment easier. An extremely popular open-source framework for managing, deploying, and scaling containers (Kubernetes) can also be used to build clouds, edges, and fog.

3.4 Energy-aware Cloud–Edge Continuum

To produce an energy-efficient data forwarding scheme for Cloud–Edge Continuum, Saraswat et al. [61] design a deadline-driven ubiquitous system. At each layer, they estimate fractions of the task to be computed for minimizing energy consumption. Overall performance is analysed using variable factors such as data size, deadline, delay, accuracy, network topologies, and energy consumption etc. To enable sustainable edge computing with distributed renewable energy resources, Li et al. [62] design a prototype model which supports coordination between edge and energy supply systems. It integrates a microgrid (e.g. a solar-wind hybrid energy system) and edge devices to ensure full utilization of renewable energy while maintaining QoS levels for time-sensitive IoT applications. To address the challenge of minimizing carbon footprints in edge networks, Yu et al. [8] model a joint task offloading and energy sharing problem. They map this minimization problem to a "minimum-cost" flow optimization problem in a graph, where nodes represent local power grids, renewable energy sources, edge servers, tasks, and batteries, and edges denote flows of energy with associated carbon footprint costs. By tracing the optimal (i.e. "minimum-cost") flow in the graph, they obtain the optimal solution. To evaluate the efficacy of proposed approach, they use a 24-hour carbon intensity dataset and compare performance based on server and battery capacities. Jeong et al. [63] develop an energy-efficient scheduling technique for federated edge clouds. The scheduling approach allocates services with actual traffic requirements to satisfy QoS levels, with the aim that it can maximize co-location of services placed on one server whilst reducing the total energy consumption of services.

To address the problem of multi-task offloading, Sharma et al. [64] suggest a hybrid approach integrating first-order meta-learning and deep Q-learning strategies. The authors use simulation to measure improvements in applications' energy consumption and training time under different settings of Cloud–Edge environments. For green mobile edge cloud environments, Chen et al. [65] develop a multi-user, multi-task computation offloading problem and apply the Lyaponuv optimization technique to decide on an energy harvesting policy. Their objective is to maximize revenue from successfully offloading tasks from mobile devices. In this context, "revenue" pertains to the efficient routing of harvested energy from the mobile edge cloud (wireless devices) to a mobile device via an energy link.

Hasan et al. [66] introduce the Aura architecture design, a highly mobile and localized ad-hoc cloud model to utilise IoT devices for work offloading techniques and upgrading apps. Through performance studies of Aura-powered IoT

devices, they show the model's efficacy in terms of job completion times, memory usage, predicted CPU clock cycle requirements, energy consumption, and cost. Gou et al. [67] suggest an architecture for collaborative computation offloading over FiWi (Fibre Wireless) networks. To reduce the total energy consumption of all the mobile devices while meeting the computation execution time limit, they address the issue of Cloud–Edge collaborative computation offloading.

To minimize carbon footprint, energy consumption, and performance interference in Cloud–Edge ecosystems, Kaur et al. [68] design a scalable controller for multi-constraint Kubernetes platforms. For efficient scheduling of containers, they formulate an integer linear programming (ILP) problem based on multi-objective optimization. Their objective is to minimize carbon footprint emissions by maximizing the utilization of green energy sources, such as wind and solar energy. To reduce overall energy consumption, the scheduler aims to consolidate incoming workloads onto a minimal number of Cloud–Edge nodes. Furthermore, they evaluate the performance of the proposed strategy by applying real-time Google Cluster traces.

For scaling and offloading optimization, Yahya et al. [69] present a two-tier architecture comprising of an access network and a core network. To optimize capacity, they introduce a two-phase optimization approach by adjusting capacity and offloading ratios repeatedly. To address privacy Conflict of interest, Xu et al. [70] present an intelligent offloading technique for smart cities, preserving privacy, enhancing offloading efficiency, and promoting edge utility. To achieve trade-offs between service response time, energy, and maintaining load balance while ensuring privacy during service offloading, the authors adopt an ant colony optimization approach.

For mobile edge computing in 5G heterogeneous networks, an energy-efficient computation offloading technique is suggested in [71]. The authors address an offloading system's energy minimization problem, taking into account the expenses associated with both task computing and file transport. Li et al. [72] present a task offloading policy that considers task deadline times. To determine the optimum offloading strategy and address the scalability issue of the deep Q-network action space, they develop an edge-to-device deep reinforcement learning approach. To improve the deep Q-network algorithm, Zhang et al. [73] present a heuristic offloading technique that minimizes both latency and energy consumption. The prime idea behind the use of a heuristic algorithm is to reduce the convergence time in hybrid edge computing networks.

Ahvar et al. [74] introduce an energy model for estimating the energy consumption of different cloud-related architectures. The authors initially present a taxonomy to classify cloud-related architectures, ranging from fully centralized to completely distributed. Subsequently, they design a PUE metric [75] based scalable energy model to evaluate the energy efficiency of diverse infrastructures. In an effort to minimize the energy consumption of serverless platforms, Rastegar et al. [76] introduce an energy-aware execution scheduler 'EneX' for serverless service providers. The authors explore the features of both offline and online solutions, considering critical factors such as complexity, scalability, and performance. Aslanpour et al. [77] design priority-based and zone-oriented algorithms to model energy-aware resource scheduling for serverless edge computing. Through real-world implementations, they demonstrate that their approach enhances the operational availability of nodes by up to 33% while maintaining QoS. To investigate the utilization of serverless platforms within the Cloud–Edge Continuum, Angelelli et al. [78] propose a multi-objective scheduling policy. This policy aims to optimize data transfers, makespan, and system usage, while considering the heterogeneity of platforms.

3.5 Analysis and discussion

We meticulously scrutinized each article, categorizing them into seven categories including research focus, Continuum coverage, formal model, energy model, optimization objectives, type of applied technique, the evaluation method, and prospective application areas, as illustrated in Table 1. The key observations are discussed as follows:

Research focus: Concerning the research focus, the majority of scrutinized papers emphasized task offloading as a primary focus, as shown in Fig. 6.

Task offloading involves the transfer of resource-intensive tasks to a separate platform to execute them more efficiently. This offloading is necessary to meet various constraints under different situations. Some key constraints include considerations like latency, load balancing, privacy, storage limitations, and adherence to Service Level Agreements (SLAs).

Continuum coverage: Most of the analyzed papers considered the Cloud–Edge Continuum architecture divided into three layers (IoT, Edge, Fog, and Cloud). The majority of works discussed in this article directed their approach towards the edge and cloud layers (shown in Fig. 7). This is the most common approach from the perspective that the Edge Layer functions as an intermediate layer to achieve the defined objectives, such as enhancing QoS metrics like latency, deadline, response time, etc. The fulfillment of requests is not solely reliant on the Edge Layer, but also involves the Cloud Layer.

Formal model: Concerning the techniques employed in the examined papers for resource management in Cloud–Edge Continuum, Fig. 8 illustrates the predominant

Table 1 Analysis of current approaches to Cloud–Edge Computing Continuum modeling

Ref.	Research focus	Continuum coverage				Formal model	Energy model	Objectives	Technique	Evaluation	Application
		IoT	Edge	Fog	Cloud						
2023 [46]	Resource orchestration	×	×	✓	✓	Graph model	Not considered	Minimise the silo-effect	End-to-end slices	Test-bed	IoT services
2023 [64]	Task Offloading	×	✓	×	✓	Reinforcement learning	Not considered	Minimise the Offloading cost	Meta-learning, deep Q-learning	Simulation	IoT applications
2023 [79]	Server management	×	✓	×	×	Game theory	Switch cost, running cost	Maximise the number of end-users served, minimise the overall energy consumption	Winner selection schemes, Nash equilibrium	Trace-driven simulation	5 G networks
2023 [80]	Task scheduling	×	×	✓	✓	Deep learning	Computer hardware, cooling facilities	Minimise energy consumption, Maximise job completion rate	Convolutional neural network	Test-bed	IoT applications
2023 [81]	Emergency demand response management	×	✓	×	✓	Game theory	Computing energy	Maximize the total utility of both the bidders and the auctioneer	Auction-based approach	Trace-driven dataset	AI/ML workloads
2023 [82]	Renewable energy scheduling	×	×	×	✓	Deep learning, Reinforcement learning	Green energy	Minimise the SLO violations, total energy monetary cost and total carbon emission	Energy-driven computing resource assignment	Trace-driven simulation	Datacenter energy supply
2023 [83]	Resource management	×	✓	×	×	Discrete event system specification	PUE	Minimise the energy consumption and operational expenses	Modeling, simulation, and optimization framework	Trace-driven dataset	Advanced driver assistance applications
2023 [8]	Task offloading, energy sharing	×	✓	×	×	Mixed integer linear programming, Graph-theory	Battery management system	Minimise the total carbon footprint	Minimum-cost flow	Trace-driven dataset	Energy management
2023 [76]	Scheduling	×	×	×	✓	Linear programming	DVFS, Intel power gadget	Minimise energy consumption	Energy-aware execution scheduler	Simulation, Test-bed	Serverless computing
2023 [78]	Scheduling	×	✓	×	✓	Approximation theory	Not considered	Minimise makespan and total cost	Shmoys and Tardos approximation	Simulation	Serverless computing
2022 [84]	Job scheduling	×	✓	×	×	Control theory	Energy harvesting	Minimise energy consumption, maximise load-balancing	Model predictive control, Douglas-Rachford splitting	Simulation	Time-sensitive jobs
2022 [85]	Energy scheduling, Task offloading	×	✓	×	×	Lyaponuv optimization	Energy harvesting	Minimise execution cost and system cost	Rounding algorithm	Trace-driven simulation	IoT applications
2022 [86]	Task scheduling	×	✓	×	✓	Queuing theory	Time-average carbon emissions	Minimise carbon emissions	Online carbon-intensity based scheduling policy	Trace-driven dataset	Computing networks

Table 1 (continued)

Ref.	Research focus	Continuum coverage				Formal model	Energy model	Objectives	Technique	Evaluation	Application
		IoT	Edge	Fog	Cloud						
2022 [87]	Task prioritization, scheduling, and offloading	✓	×	✓	✓	Fuzzy theory, Population-based	Not considered	Minimise makespan and service latency	Fuzzy logic, Elitism-based multi-population Jaya	Simulation	Latency-sensitive applications
2022 [77]	Resource scheduling	×	✓	×	×	Heuristic	Green energy, battery-powered	Maximise the operational availability	Bin-packing	Trace-driven dataset	Serverless computing
2022 [88]	Resource management	×	×	×	✓	Artificial Intelligence	Energy, thermal and cooling	Maximise the QoS objective score	Gated Graph Convolution Network	Simulation, Test-bed	Industrial
2022 [44]	Computation Offloading	×	✓	×	✓	Graph-based	Not Considered	Minimise communication latency and maximise performance	Louvain community detection	Test-bed	Social network
2022 [30]	Resource allocation	×	✓	✓	✓	Integer linear programming	Not Considered	Minimise processing cost and propagation latency	Heuristic	Simulation	IoT applications
2021 [89]	Resource scheduling	×	×	×	✓	Heuristic, Machine learning	Brown Energy, Renewable Energy, Cooling Energy	Maximise green energy usage	Consolidation, host scaling, brownout, Support vector machine	Test-bed	Web-based application
2021 [63]	Service scheduling	×	✓	×	✓	Heuristic, Reinforcement learning	Static energy, dynamic energy, migration energy	Minimise total energy consumption	Energy first, migration first, Q-learning	Simulation	Face recognition, Online text translation
2021 [69]	Scaling and offloading optimization	×	✓	×	×	Queuing theory	Not Considered	Minimise total capacity cost	Latency aware two-phase iterative optimization	Simulation	5 G Networks
2021 [72]	Task offloading	×	✓	×	×	Reinforcement learning	Dynamic voltage and frequency scaling	Maximize the reward (the weighted sum of utility of processed tasks, penalty of tasks dropping, and energy consumption)	Energy-aware task offloading with deadline constraint	Simulation	Mobile applications
2021 [73]	Computation offloading	×	✓	×	×	Queuing Model, Reinforcement learning	Not Considered	Minimise total time delay and energy consumption	Deep reinforcement learning-based computation offloading and shunting	Simulation	Compute-intensive and time-sensitive applications
2021 [45]	Task scheduling	×	×	✓	×	Heuristic	Not Considered	Maximise the successfully completed jobs	Earliest deadline first	Simulation	Real-time systems
2021 [47]	Microservice deployment	×	✓	×	✓	Graph-based, Reinforcement learning	Not Considered	Minimise the latency and resource usage	Nautilus	Test-bed	Real-system applications

Table 1 (continued)

Ref.	Research focus	Continuum coverage				Formal model	Energy model	Objectives	Technique	Evaluation	Application
		IoT	Edge	Fog	Cloud						
2021 [49]	Resource management	×	×	✓	×	Architecture analysis design language	Not Considered	Not mentioned	Fog computing platform	Test-bed	Industrial IoT
2021 [50]	Resource auto-scaling	✓	×	✓	✓	Deep Learning	Not Considered	Minimise total cost and QoS violation	Recurrent neural network	Simulation	IoT applications
2021 [51]	Application-level orchestration	×	✓	×	✓	Framework	Not Considered	Not mentioned	MiCADO-Edge	Test-bed	Video processing, healthcare
2021 [90]	Application placement	×	✓	×	✓	Fuzzy theory, Heuristic	PUE, Intra and inter cloud network	Minimise energy cost and carbon emission	A* algorithm, Fuzzy sets	Simulation	IoT applications
2021 [91]	Energy measurements	×	✓	×	×	Empirical model	Power meter, Test and learn strategy	Minimise energy consumption	WattEdge	Test-bed	Smart agriculture
2020 [68]	Job scheduling	×	✓	×	✓	Integer linear programming	Green energy	Maximise the green energy usage, minimise carbon footprint emission and performance interference	Kubernetes-based energy and interference driven scheduler	Trace-driven dataset	Industrial IoT
2020 [92]	Vertical, horizontal, and reverse Offloading	×	✓	×	✓	Queuing theory, Heuristic	Not considered	Minimise the total cost	M/M/k, Simulated annealing	Simulation	Time-sensitive jobs
2020 [61]	Data forwarding	×	✓	✓	✓	Mixed-integer programming, Queuing theory	Static energy, dynamic energy	Minimise the energy consumption	Newton method, D/M/1 and M/M/1	Test-bed	Ubiquitous computing
2020 [70]	Service offloading	×	✓	×	×	Queuing theory, Heuristic	Propagation power, Executing power, Total energy power	Minimise the energy, load balance, and service response time	M/M/Z/∞/∞, Ant colony	Simulation	Smart city services
2019 [56]	Microservices placement	×	✓	×	✓	Control theory	Not Considered	Minimise the latency and battery consumption	A3-E ((A)wareness, (A)cquisition, (A)llocation, and (E)ngagement)	Test-bed	Augmented Reality
2019 [57]	VNF provisioning	×	✓	×	✓	Heuristic	Not Considered	Minimise the latency	Latency-aware VNF provisioning	Trace-driven simulation	Time-sensitive applications
2019 [93]	VM migration, Energy scheduling, Task allocation	×	✓	×	×	Heuristic	Green energy	Minimise the energy cost	Relaxation-based algorithm	Simulation	Energy Internet

Table 1 (continued)

Ref.	Research focus	Continuum coverage				Formal model	Energy model	Objectives	Technique	Evaluation	Application
		IoT	Edge	Fog	Cloud						
2018 [62]	Workload scheduling	✓	✓	×	✓	Heuristic	Hybrid power supply, Power capping	Maximise the renewable energy usage and system performance	Demand response program, Energy efficiency program	Lab-scale prototype	IoT applications
2018 [65]	Computation offloading	×	✓	×	✓	Lyaponuv optimization	Energy harvesting, energy links	Maximise the revenue from successful task offloading from mobile devices	Greedy maximal scheduling	Simulation	Mobile applications
2018 [66]	Computation offloading	✓	×	✓	×	Contract theory	Average energy consumption	Minimise the task completion time and energy consumption	Tiered incentive model	Test-bed	Mobile applications
2018 [67]	Computation offloading	✓	✓	×	✓	Approximation theory & Game theory	Local energy and Offloading energy	Minimise the energy consumption	Greedy strategy	Simulation	FiWi networks
2018 [94]	Streaming big data analytics	×	×	×	✓	Reinforcement learning	Discrete-time power model	Minimise the energy cost and migration cost	Neural network, Random pool sampling	Trace-driven dataset	Big data
2017 [48]	Service placement	×	✓	×	✓	Mixed integer linear program	Not considered	Maximise the mean satisfaction rate and minimise the percent idle time	Centralized algorithm	Trace-driven simulation	Real-time application
2017 [13]	VM placement	×	×	×	✓	Heuristic	Brown energy, green energy	Minimise the energy cost and carbon footprint cost	Bin-packing	Simulation	HPC applications
2016 [71]	Task offloading	✓	✓	×	×	Mixed integer programming	Local energy and Offloading energy	Minimise the total energy consumption	Device classification, priority assignment, channel allocation	Simulation	5G networks

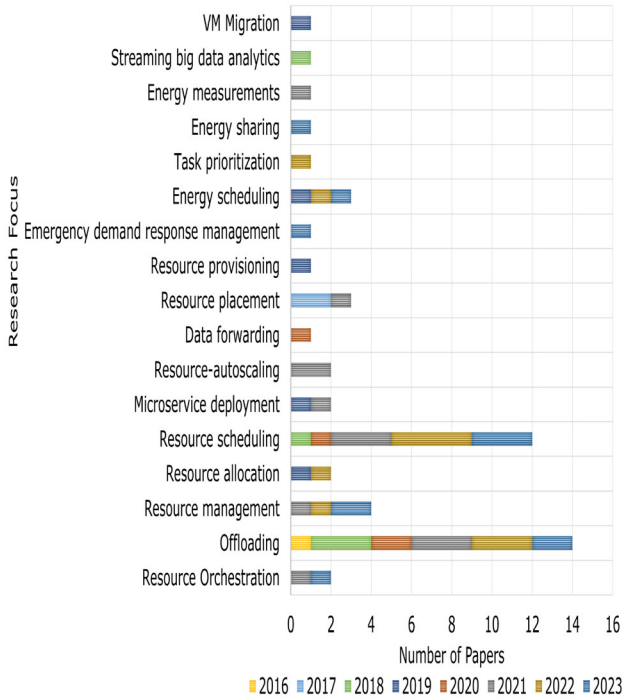


Fig. 6 Analysis based on research focus

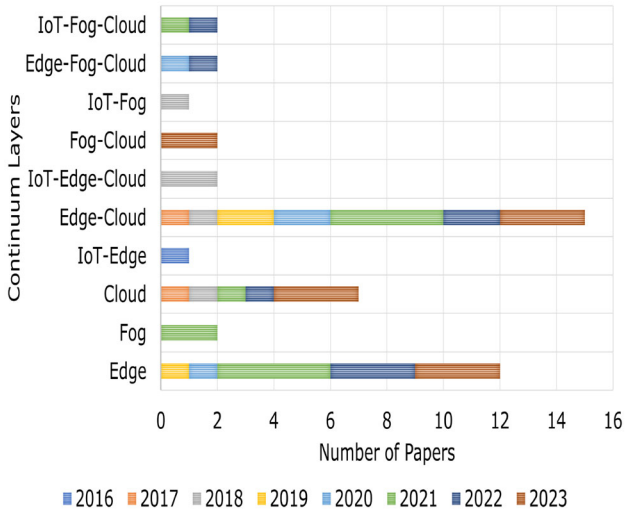


Fig. 7 Analysis based on coverage

modeling techniques. These strategies are categorized into Integer Linear Programming (ILP), empirical model, discrete event-based, population-based, fuzzy theory, game theory, approximation theory, contract theory, Lyapunov optimization, control theory, deep learning, architecture/framework, queuing theory, machine learning, heuristics, artificial intelligence, reinforcement learning, and graph theory. Notably, heuristics emerged as the most frequently utilized modeling technique in the analyzed papers.

The key advantage of heuristics is that they are straightforward algorithms, with less execution time. In the

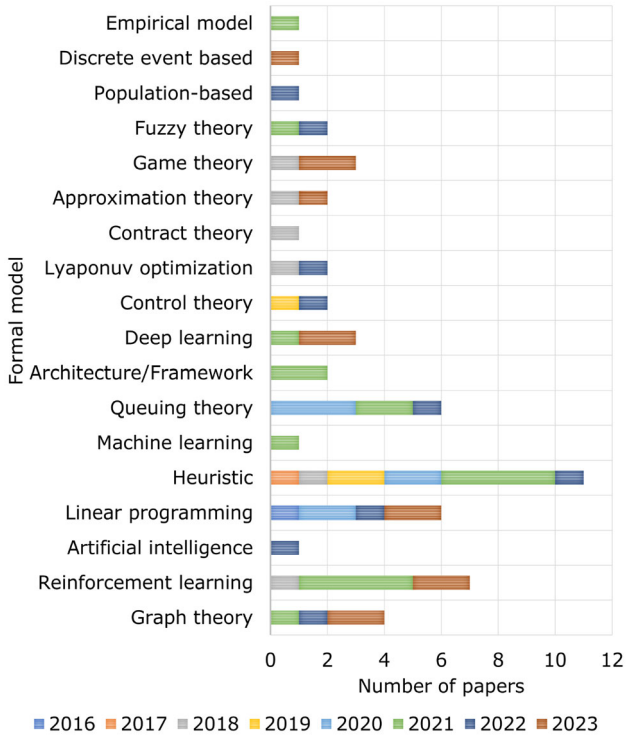


Fig. 8 Analysis based on formal modeling

heuristic-driven solution, decisions are generated based solely on available information, without considering the future effects. This leads to locally optimal choices at each stage of execution, aiming to achieve a good, though not necessarily optimal, but fast and sub-optimal solutions. Such an approach aligns well with the Cloud–Edge Continuum model, as it gives the ability to accommodate the dynamic nature of the environment, characterized by platform heterogeneity, high geographical distribution, and interoperability.

Energy model: Integrating energy awareness into the Continuum design process significantly reduces the energy consumption. The Cloud–Edge Continuum relies on diverse energy driven models, including brown energy model driven by grid power (electricity) for a reliable and uninterrupted power supply, green energy model driven by renewable energy sources such as solar and wind power, battery power for providing backup and sustained service during grid outages, and hybrid energy models that combine multiple energy sources to enhance reliability and efficiency. These energy models ensure uninterrupted operation, reducing carbon footprint, and optimize energy efficiency within the Cloud–Edge Continuum. The selection of energy sources depends on several factors such as location, environmental impact, cost considerations, and the specific requirements of services. Upon reviewing the selected publications, it is observed that most of them have not explicitly considered a specific energy model. In some

of the papers, various energy measures are applied (shown in Fig. 9).

For example, Li et al. [62] utilized a power-capping strategy to align the power supply of edge systems, enabling the postponement of executions for delay-insensitive applications until the local renewable energy becomes available. Other approaches, such as power metering, energy harvesting, power usage effectiveness (PUE), battery management systems (BMS), dynamic voltage and frequency scaling (DVFS), and energy-saving strategies, are applied in different works.

Optimization objective: From our analysis of the articles, it is evident that the primary optimization objective in the Cloud–Edge Continuum is the reduction of energy consumption, as illustrated in Fig. 10. Indeed, this objective is not only standing alone but also intersects with various other modeling objectives, including cost reduction, latency reduction, carbon footprint reduction, execution time reduction, and even performance maximization.

Evaluation: Simulation tools and models play a crucial role in assessing the efficacy of system to work closer with real-world conditions. A predominant approach among the analyzed papers involves the utilization of either numerical simulators or trace-driven simulations to validate their methods (shown in Fig. 11).

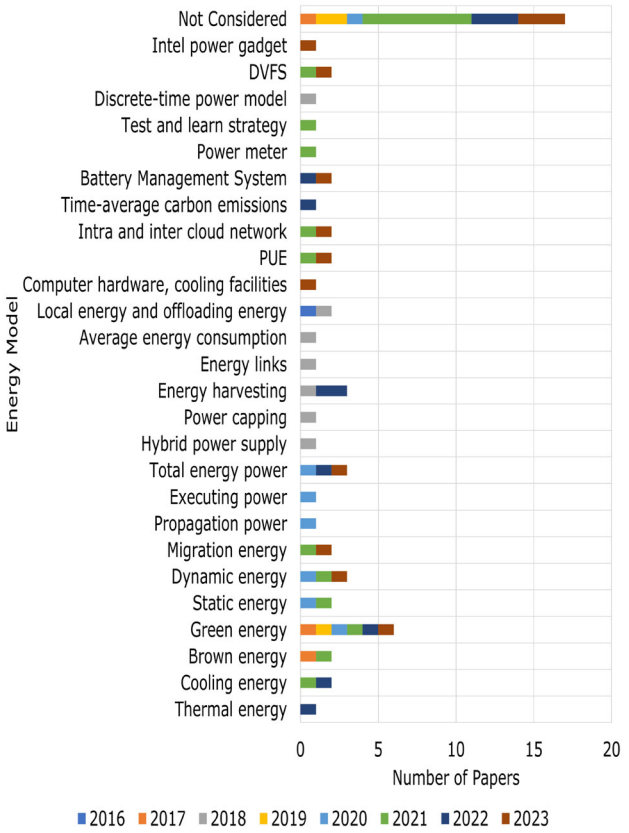


Fig. 9 Analysis based on energy model

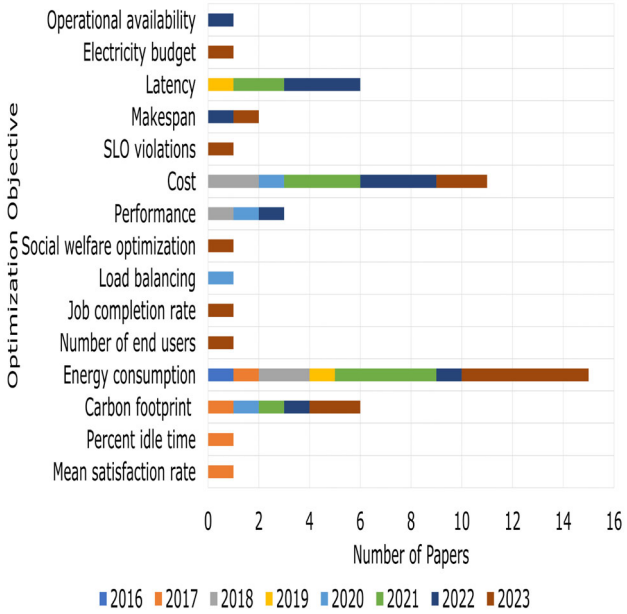


Fig. 10 Analysis based on objectives

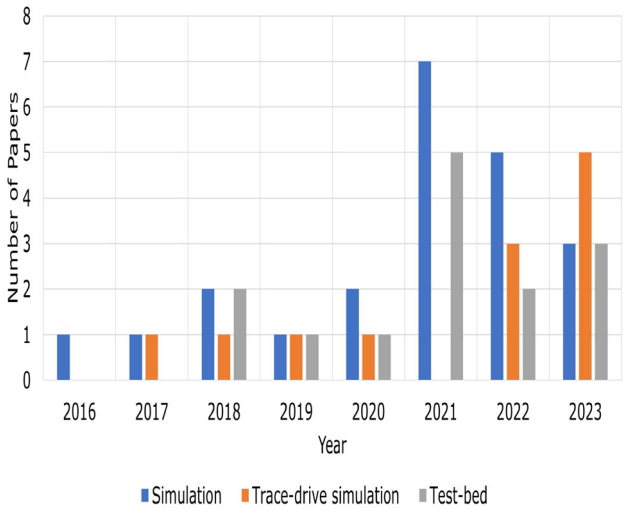


Fig. 11 Analysis based on evaluation

This form of simulation is valuable while studying the behavior of systems whose mathematical models are too complex to provide analytical solutions, as in many nonlinear systems. The majority of works rely on small datasets, synthetic datasets, or datasets that lack representation of real-world scenarios. Only a limited number of works have applied small-scale test-beds for real-time modeling. However, drawing definitive or comprehensive conclusions is challenging without an evaluation in a production environment.

Application: Over the past few years, there has been an increasing attention on systems that support the IoT applications and time-sensitive tasks (shown in Fig. 12).

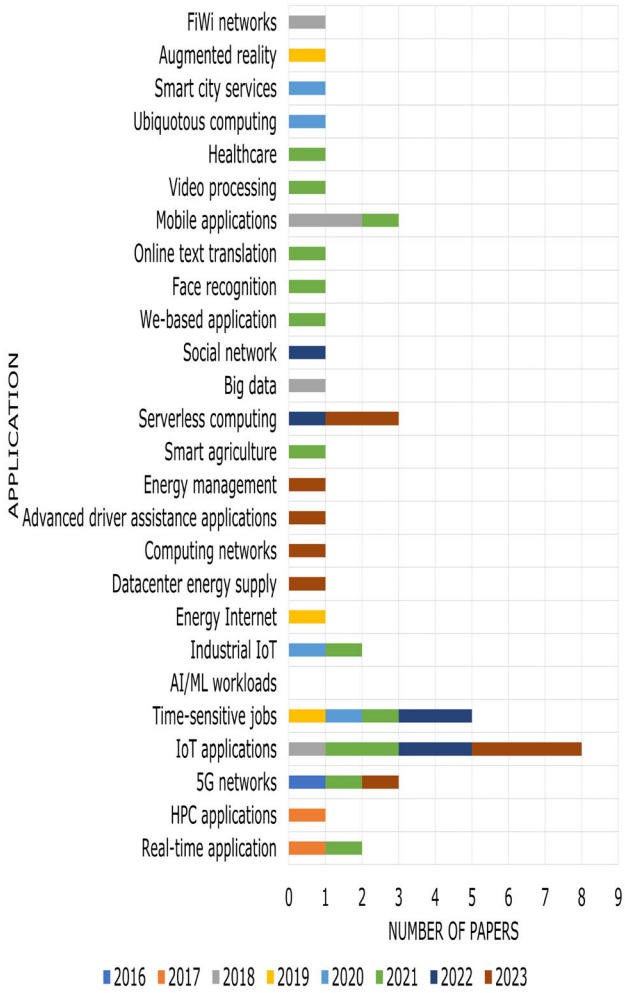


Fig. 12 Analysis based on application

In some of the works, serverless computing and mobile applications are also considered. With the rising demand for applications necessitating low latency, it is expected that new use cases for the Cloud–Edge Continuum will emerge in the coming years.

3.6 Key omissions in Continuum modeling

After careful investigation of the current research on Continuum modeling, we observe that in the literature, several formal models for traditional cloud systems have been proposed, e.g. [13, 95] but these do not capture the dynamic nature of Cloud–Edge systems or integrate stochastic properties, energy providers, pricing, and renewable energy sources. Most work assumes a single data center, precluding intrinsic challenges faced with the management of federated systems, such as how to monitor and schedule multiple complex resources across multiple networks in a scalable and decentralised manner with SLO awareness [96], and how to balance accuracy with decision making latency (many recent approaches, such as [97, 98], use machine-learning methods that are too slow to provide the ultra-low latency scheduling required by edge applications). A basic model that integrates nodes with energy providers is presented in [89] but does not consider federated edge systems or cross-site monitoring issues, while [62] only considers micro-grid integration with edge nodes, with no centralised cloud integration. Of work that does consider energy-aware federated systems, little has been achieved; [99] propose an integration between smart grids and Cloud–Edge systems but the proposed architectural model is extremely high-level and does not consider monitoring overhead, task properties, or decentralised control of the system.

The overall analysis of the state-of-the-art on Cloud–Edge Continuum highlights the lack of unified systems, formal models, and methods to seamlessly integrate various energy factors including temporal pricing, renewable energy sources, energy provider requirements, resource restrictions, and balance consumption over large-areas with other non-Cloud consumers. Research in this field typically results in either reference architectures or simulated system environments, with computing, networking, and storage resource management serving as the primary focus. These observations demonstrate the absence of a unified resource orchestration technique capable of integrating the pricing models, types of workloads, multi-objective optimization, monitoring, and controlling strategies, QoS and SLO requirements of end-users, heterogeneous systems and networking technologies, energy policies, energy providers, energy sources, and administration of compute and network resources in the energy-aware federated Cloud–Edge Continuum. *There is a clear need to bridge this gap and exploit the modeling of Continuum key components, their relevant stochastic properties and interactions, and their integration with key energy factors*.

4 Research challenges

Based on the omissions described in the previous section, we identify seven key research questions that must be addressed to adequately integrate energy considerations into a formal model for the Cloud–Edge Continuum.

4.1 How to model the system?

In the literature, there is a lack of formal models for federated Cloud–Edge systems in general; no existing model incorporates energy providers, pricing, and sustainability. The creation of formal energy-aware models for federated Cloud–Edge systems is a challenging task due to a lack of empirical data to calculate stochastic properties, a lack of

analysis to model geographical energy distribution factors such as supply and demand of green and brown energy sources, a limited understanding of temporal energy pricing, and limited modeling of energy provider policies & restrictions. To address this, empirical data must be assessed across a range of disciplines, and appropriate model types identified for each sub-system.

4.2 How to combine models?

Once models for each sub-system in an energy-aware Cloud–Edge infrastructure have been created, there are still significant challenges with regard to integrating these models. These challenges include how to best integrate different model types (e.g. a graph-based model integrating with a model based on queuing theory), how to determine appropriate granularities when simulating the models, how to mathematically reason across the combined model, etc. These challenges are not unique to Cloud–Edge systems, but various solutions in the literature need to be properly assessed to determine which is appropriate for the scale and number of interactions required.

4.3 How to model different regions or sub-sets of the system?

Optimization at local level e.g., for a specific sub-system (single data center, application, device, etc.) is relatively straightforward to achieve. However, optimising or balancing resources across geographically federated regions and providers is an extremely challenging task due to the heterogeneity of the respective control systems, different API models, multiple ownerships, conflicting priority levels, user fairness constraints, monitoring and scheduling complexities of multiple resources across multiple networks with SLO awareness.

4.4 How to develop a self-stabilizing model?

In the Continuum, failure of a node (from server to data center level) will impact performance and result in task interruption. An application's sub-tasks may run on various edge nodes; all sub-tasks executing on a specific resource will be interrupted if it fails, and any sub-tasks that depend on those interrupted sub-tasks will likewise be interrupted (a partial manifestation of the "long tail" problem seen in e.g. [100]). There is therefore a challenge to create a failure-resilient scheduling model that can recognize dependencies between tasks and reschedule sub-tasks impacted by failure events to limit interruptions. A further challenge is to develop a self-stabilizing architecture which can recover from transient faults automatically without any manual intervention, as it is predicted that the failure probability of edge servers will be far higher than that of cloud servers [101].

4.5 How to maintain energy performance trade-offs?

There are several studies that have investigated to enhance the performance of individual cloud or edge systems. Most of the existing studies are focused on resource management in a non-federated Cloud–Edge system but do not consider federations of resources (e.g. Cloud–Edge). Additionally, there are no best practices or guidelines to optimize or monitor the overall performance of the federated-Cloud Edge Continuum. In a federated Cloud–Edge system, nodes and regions have different SLOs and pricing-as do energy providers. It is a critical task to optimize between individual and regional SLOs while ensuring performance. Therefore, we need to balance local and global optima at different levels within the stack (e.g. edge, fog, cloud, regional etc.) How to arbitrate and optimize conflicting service levels and energy requirements in a holistic manner across these levels is not yet fully understood.

4.6 How to model green Cloud–Edge systems?

Many new challenges arise when considering the impact of Cloud–Edge resources on power grids, especially when other users and demands on those power grids are taken into account. Different power grids may have different capacities and sources of renewable energy at any moment in time; for example, a power grid in region *A* may at a specific point in time incorporate 20% of its available power from renewable sources and have 30% free capacity. Later in the same day or week, those numbers may change to 10% and 15% respectively. It may therefore be extremely valuable to schedule tasks in a Continuum between different grids to improve utilisation of renewable sources and available capacities (and hence lower costs) whilst maintaining service levels for users and applications. *modeling these factors and ultimately integrating these models into energy-aware resource management systems is a significant and vitally important challenge that needs to be addressed*. As observed in [62], approximately 80% of today's energy is still produced from brown energy sources; mechanisms to increase the use of green energy sources in the Continuum will go a great way towards reducing its carbon footprint (and hence impact on the environment).

4.7 How to develop validation models?

The model-based simulation of any Cloud–Edge system can utilise some existing simulators (such as

EdgeCloudSim [102], ENIGMA simulator [103]). However, to iteratively test different aspects of an entire federated Cloud–Edge system such as decentralized monitoring, arbitration, and optimization is a challenging task due to limited scalability scenarios, Continuum mobility behaviours, topology configurations, network behavior at different levels of granularity, and energy considerations. In addition, designing a software-defined networking-based test-bed to monitor and track the energy consumption of an entire federated Cloud–Edge infrastructure adds another level of complexity.

5 Energy-considerations for modeling the Green Cloud–Edge Continuum

Efficiently integrating energy considerations into Cloud–Edge models is paramount for sustainable and cost-effective computing, ensuring a reduced environmental footprint and optimal resource utilization. It aligns with the broader goals of cost reduction, and long-term viability in the rapidly evolving landscape of cloud and edge computing. Intrinsically, acquiring such objectives depends on identifying the potential energy consumption factors and their effect on these platforms. Therefore, in this section, we discuss several studies involving different energy-considerations (shown in Table 2) for a resilient and energy-aware Cloud–Edge models.

- Energy Sources: The Cloud–Edge Continuum relies on various energy sources, such as Grid Power (Electricity) for reliable and continuous power supply, Renewable energy sources such as solar and wind power, Battery Power for providing backup power and ensuring uninterrupted service during grid outages, and Hybrid energy systems combine multiple sources to enhance reliability and efficiency. These energy ensures uninterrupted operation, reducing carbon footprint, and optimizing energy efficiency within the Cloud–Edge Continuum. The selection of energy sources depends on various factors such as location, environmental impact, cost considerations, and the specific requirements of services. Upon reviewing the selected publications, it is observed that most of the publications have considered dual energy sources. Khosravi et al. [13] considered off-site brown, and on-site renewable energy sources for VM placement in distributed cloud model. Nan et al. [104] considered dual energy sources for energy-aware computation offloading in Cloud of Things systems, where solar power represented the primary energy supply and grid power is used for the backup supply. Li et al. [62], and Xu et al. [70] assumed hybrid power supply model which draws power from both power grid and renewable energy sources to ensure the service reliability in Cloud–Edge systems.
- Energy Constraints: It is necessary to adopt the energy constraints in the Continuum, otherwise it can lead to service disruptions, performance increased operational costs, and environmental concerns. To maintain reliability and sustainability in computing and networking operations, authors have adopted energy constraints depending on generated energy, available energy, storage device operating range, pricing, and total energy budget etc. [13, 62, 104].
- Energy Gentrification: Energy gentrification [6] is considered as an analytical framework through which we can examine negotiations and potential conflicts that may arise when grid owners need to determine priority in allowing grid access to different stakeholders. It also provide the decision making and guidelines to develop energy policies. Only few of the works have considered the gentrification perspective. Such as Libertson et al. [6] identified different scenarios in their case study such as the prioritization of global versus local capital, the resource competition, and the trade-offs between private interests and common goods etc.
- Energy Storage: Continuum systems might not fully utilize all of the available renewable energy during their low workload periods. In such cases, any excess renewable energy can be either stored in energy storage devices or harnessed through net-metering [118]. In some of the works different battery models are adopted for example, Silva et al. [105] adopted UAV (unmanned aerial vehicle) battery model to manage the variable workload for fog infrastructure. To implement the concept of energy sharing, Yu et al. [8] considered a battery management system (BMS) to provide energy to other edge servers.
- Energy Price Metrics: Energy price metrics provide valuable insights into the cost-effectiveness of energy usage. It helps the provider to make informed decisions about when and how to allocate energy resources, balance grid power with renewable sources, and optimize energy-intensive operations. In the literature, several price metrics are used - including cents per kilowatt-hour energy usage (cents/kWh) [13, 104], and determining energy price by different time periods such as peak hour (high activity times, e.g., 2pm-8pm), off-peak hour (when activity levels are lowest, e.g., 10pm-7am) and shoulder hour (moderate levels of demand, e.g., 7am-2pm and 8pm-10pm) [104, 119] are applied for efficient budget planning, and to produce economically viable solutions for energy-aware Continuum environments.
- Energy Models: It is essential to implement hybrid power supply that can harness energy from both the

Table 2 Energy-considerations for modeling the Green Cloud–Edge Continuum

Energy-consideration	Description	Technique	Selection factors
Energy sources	Provide reliable and continuous power supply	Brown energy [13], Green energy [13], Hybrid energy [62, 70, 104]	Location, climate impact, cost, and service requirements
Energy Constraints	To avoid service disruptions, operational costs, and environmental concerns	Maximum available energy, Communication to computation ratio (CCR) [62]	Generated energy, available energy, storage device operating range, pricing, and total energy budget etc
Energy Gentrification	Examine negotiations and potential conflicts between stakeholders	Energy gentrification [6]	Prioritization of global vs. local capital, the resource competition, and the trade-offs between private interests and common goods
Energy Storage	To store excess renewable energy	UAV battery model [105], battery management system (BMS) [8]	Low workload periods
Energy Factors	Factors affecting energy consumption of Cloud–Edge devices	WattEdge [91]	Idle state, CPU, memory, storage, resource bundle, short-range connectivity, network bandwidth utilization, communication protocols, Energy storage
Brownout Approaches	Self-adaptive approach to enable/disable services	CLOUDFARM [106], EDB [107], HYBP [108], SaaScalar [109]	Application design, Workload scheduling, Resource usage monitoring, Brownout controller design, Performance metrics [110]
Energy Load Forecasting	For efficient distribution of power, planning process, and auxiliary operations [111]	Energy cloud management (ECM) [112]	Industry equipment, lighting, water heaters, motors, air-conditioning, peak hour, and off-peak hour load [111, 113]
Energy Models	To harness energy from both the power grid and renewable energy sources	Thermal model [88], cooling model [88], power-capping [62], power metering [91], energy harvesting [84], PUE [83], DVFS [76]	Energy cost, carbon cost, delay sensitive & insensitive applications
Demand Response Prediction	To balance the gap between energy supply and energy demand	Data analytical demand response management [114]	Power supply, power demand
Electricity Price Prediction	Estimate price distribution in future time periods	Short-term, Medium-term & Long-term forecasting [111]	Seasonality, Day-to-day market operations
Energy Outage	Temporary loss of electrical power	Smart metering [115]	Severe weather conditions, equipment failure, maintenance activities [111]
Energy Price Metrics	Provide valuable insights into the cost-effectiveness of energy usage	Cents per kilowatt-hour energy usage (cents/kWh), peak hour, shoulder hour and off-peak hour [13, 104]	Peak hour, Off-peak hour
Green Energy Instability	Due to insufficient renewable energy supply	Instability-resilient green energy allocation system [82]	Time of a day, season, climate [82]
Energy Metrics	To measure the performance and impacts of changes injected in subsystems	Metrics for Greenness, Energy efficiency, Thermal and Air management, Cooling, Performance, Network, Storage, Security, and Financial impact [17, 18]	IT equipment, cooling systems, temperature control, site selection, building structure, and power supply and distribution systems [17, 18]
Anomaly Detection	Unusual energy consumption patterns	Ensemble learning framework [116], Decision tree and SVM-based data analytics [117]	Operation behaviour, usage anomaly, theft detection, load anomaly [111]

power grid and renewable energy sources. So that, we can minimize the adverse impacts of instability on renewable energy supply and ensures the smooth operation [62]. For example, Khosravi et al. [13] adopted renewable energy and brown energy based model to minimize the energy and carbon cost in data centers. In some other works thermal, and cooling based models [65, 88] are also used.

- Other Energy Measures: There are many other energy measures used in different papers. For example, Li et al. [62], utilize a power-capping strategy to match the power supply of edge systems. Through this strategy,

we can postpone the executions of delay insensitive applications until the local renewable energy supply becomes available. Lajevardi et al. [120] suggest the Power Density Efficiency (PDE) metric to provide further insights into energy-efficiency and the effectiveness of thermal management. Considering both energy efficiency and performance requirements, it is crucial to balance the rate at which power is dissipated (power density) with the total amount of energy consumed over a period of time - indeed, some computation models may help to reduce energy density while increasing total energy consumption. Similarly, other approaches include analytical methods [13, 64, 67, 70, 71], energy metrics [17, 18], power metering [91], energy harvesting [84], power usage effectiveness (PUE) [83], and dynamic voltage and frequency scaling (DVFS) [76] etc.

6 High-level model to resolve key omissions

Although there are some preliminary studies on federated Cloud–Edge systems but they are still in their early stages. Thus, it opens several opportunities for future research in energy-aware Cloud–Edge Continuum architectures.

To reason over federated Cloud–Edge systems, key components, and their relevant features and interactions need to be identified and modeled; no Cloud–Edge model has yet been created that integrates multiple components such as energy providers, renewable energy sources, energy pricing, energy provider policies, and restrictions. The major challenges are to identify key hardware, network, and energy components within a Cloud–Edge system and categorize these into a layered stack. Interactions between components and layers are required to be analysed and formally modeled.

To model such a system, the formal model can incorporate three aspects: (i) Creation of a formal layered model: For the development of such models, we need to identify and categorize the different energy, network, hardware, and software components prevalent in Cloud–Edge systems into a series of interacting layers. Interactions between and across these layers are needed to be explored and defined.

(ii) Identify and build models of typical Cloud–Edge workloads: This task is concerned with identifying common types of workload submitted to Cloud–Edge infrastructures and quantifying their resource consumption, duration, network, and energy characteristics.

(iii) A predictive energy consumption model for data centers and workloads: Utilizing the outcomes of (i) and (ii) as the basis for developing a method to quickly estimate predicted energy consumption within Cloud–Edge nodes. This could be used as part of the decision mechanism when balancing and optimizing software placement.

6.1 Perspective model

To address the core research challenges and establish a comprehensive framework, we aim to develop integrated models that encompass various components such as data centers, edge devices, fog nodes, energy providers, software workloads, and the requirements and objectives of users and stakeholders. We propose a perspective model for energy-aware Cloud–Edge computing Continuum as shown in Fig. 13- that identifies the end-users, application manager, Cloud–Edge Continuum infrastructure, network offloading manager, energy provider policies & metrics, and controller components for an energy-aware design, and interconnection between them. The operational aspects of these components are elaborated as follows:

(i) End Users: End users submit their service requests to the Cloud–Edge system through the end-users layer. Within this layer, users have the ability to specify certain QoS restrictions for their requests. These may include parameters such as maximum tolerable delay, available bandwidth for data transfer, deadline, budget, as well as specific privacy and security requirements. By providing these QoS restrictions, users can communicate their desired service levels and constraints to the Cloud–Edge system, allowing it to prioritize and allocate resources accordingly. Subsequently, the end users' submitted requests are directed to the application manager.

(ii) Application Manager: The application manager is responsible for selecting suitable platforms to fulfill end users' requests across the heterogeneous and distributed resources of the Cloud–Edge Continuum. There are several common types of workloads that are typically submitted to the Continuum infrastructures. These include IoT data processing, collaborative applications, web and application hosting, video streaming and content delivery, data storage and retrieval, big data analytics, offloading workloads, and real-time applications etc. To process these different types of incoming workloads, the application manager characterizes (as discussed in Sect. 3.1) the workloads based on their resource consumption, duration, network, and energy characteristics and utilizes various hosting engines. The hosting engines can be containerization platforms like Docker [121] and Kubernetes [122], serverless computing platforms such as AWS Lambda [123],

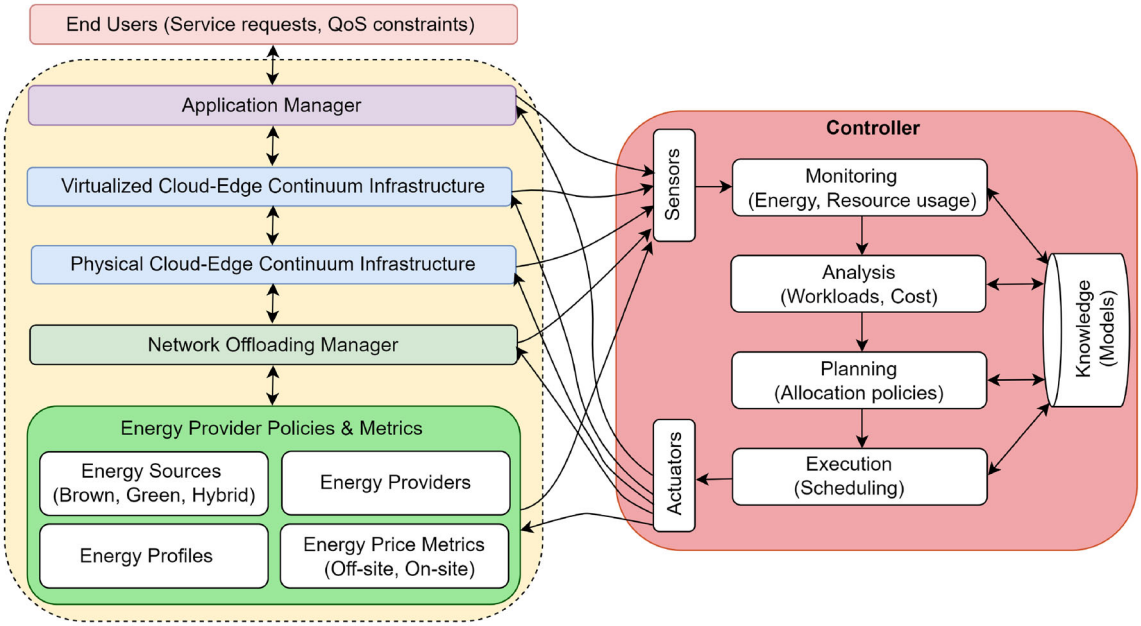


Fig. 13 A high-level model for energy-aware Continuum systems

Microsoft Azure Functions [124], Google Cloud Functions [125], and IBM Cloud Code Engine [126]. The selection of a hosting engine depends on crucial factors such as workload requirements, resource constraints, latency considerations, and scalability needs. Each hosting engine offers specific benefits and features that align with the workload characteristics and goals of the Cloud–Edge system design.

(iii) Continuum Infrastructure: The Continuum infrastructures can be divided into two parts: (i) virtualized Cloud–Edge Continuum infrastructure, utilizing virtual resources, and (ii) physical Cloud–Edge Continuum infrastructure, utilizing physical resources. For example, if an application is containing several microservices then their deployment across various Virtual Machines (VMs) is a feasible option. The virtualized platform efficiently handles virtualized resources managed through platforms like VMware [127]. In the context of single-layer cloud systems, infrastructure management platforms like OpenStack [128] come into play, facilitating the deployment of multiple VMs across various hosts. In the case of multi-layer systems that integrate cloud, fog, and edge resources, the application manager needs to handle the specification of dependencies, execution logic, performance metrics, and lifecycle management of running services [129]. It needs to enable the coordination and orchestration of complex workflows within the Continuum.

Additionally, to consider data storage requirements and the recent emergence of accelerators (e.g. TPU, GPU, ASIC, FPGA, etc.), the multi-layer system adopts different components including: (i) data aware policies to optimize data storage mechanisms and ensure efficient data movement between different layers; (ii) integration of accelerators to accelerate data-intensive tasks; (iii) dynamic provisioning of resources based on computational and data processing aspects; (iv) data caching techniques to store data in near-edge resources to improve latency, speed up task execution and save energy; (v) optimisation of data transmission protocols to select feasible communication protocols and data compression techniques to reduce the overhead associated with data transmission.

(iv) Network Offloading Manager: Since Cloud–Edge Continuum systems aim to enhance user experience by providing better QoS, it is important to perform well regulated filtering at the network edge to filter unuseful data. This is the responsibility of infrastructure networking. Similarly the network offloading manager is also responsible for balancing the traffic across the Cloud–Edge Continuum nodes, multi-tenant networking, securing data from interception through packet sniffing, monitoring delays & packet losses, and packet transfer rates. One critical concern is that of maintaining data sovereignty - ensuring that data is stored and processed in compliance with

regulatory requirements across the system. We propose to treat data sovereignty as a set of constraints (e.g. regional whitelists, sets of excluded nodes/providers, etc.) to be incorporated by the network offloading manager. To incorporate data considerations, the network offloading manager stores data network locations, which are combined with the underlying network topology model to calculate the impact on performance, latency, energy etc. of transmitting data across network links.

(v) Energy Provider Policies and Metrics: The energy provider policies and metrics module is introduced to integrate energy considerations in the Cloud–Edge paradigm. It consists of sub-modules such as: energy sources (brown, green, hybrid), energy providers (responsible for following different regional considerations such as grid control policies and power regulations by governing authorities), energy profiles, and energy price metrics for both off-site and on-site utility grid providers. The functionality of these sub-components is explained as follows:

(a) Energy Sources: The foremost purpose of this sub-module is to analyze all available energy sources available in a particular region and maximize the use of renewable energy sources while, at the same time, assuring reliable and efficient Cloud–Edge Continuum systems. For better sustainability, reducing operational energy usage and energy wastage alone is not sufficient. Using green energy as much as possible and minimizing power supply to the infrastructure is equally crucial. To attain this goal, the implementation of a demand response program [130] has become indispensable. It provides a balance between demand and supply in the Cloud–Edge Continuum by efficiently coordinating with available energy sources. This strategy employs direct and indirect load control strategy to optimize power usage and maintain a balance between the demand and supply of electricity [62]. In the Cloud–Edge Continuum, energy dependency is largely on grid electricity that comes from sources such as coal, natural gas, nuclear plant, hydroelectric, wind or solar plants. These sources are location dependent. To increase the use of renewable energy, energy sources such as hydroelectric, solar or wind plants need to be used more so that environmental footprint can be reduced. On-site energy generation and grid electricity can also be combined to provide a hybrid approach to ensure reliability and less environmental footprint.

(b) Energy Providers: Energy providers are responsible for delivering a consistent and reliable supply of energy for uninterrupted operations in the Cloud–Edge Continuum, while also following different regional considerations such as grid control policies and power regulations (such as energy gentrification [6] perspectives to prioritize user requests for grid owners).

Grid control policies and power regulations may exhibit variations across countries, regions, and utility companies, as they aim to balance several objectives. These include ensuring grid reliability, promoting the adoption of renewable energy, optimizing energy markets, and safeguarding consumer interests in the energy sector [6]. The energy grid control policies and regulations component stores information about the rules and guidelines established by governing authorities to govern the operation, management, and control of the energy grid.

It encompasses various aspects of grid operations, including policies dictating the types of power generation sources allowed, such as renewable energy sources (solar, wind, hydro) or traditional fossil fuel-based power plants; requirements for grid interconnection and power quality standards; load management measures aim to maintain grid stability and prevent overload conditions; grid resilience policies focus on enhancing the resilience of the grid to withstand disruptions, energy market, and pricing regulations; policies addressing the integration of distributed energy resources (DERs), such as rooftop solar panels or small wind turbines, into the grid; environmental policies to promote cleaner energy production and reduce greenhouse gas emissions; policies focusing on energy consumers' rights etc. These grid control policies and regulations provide a framework for governing energy grid operations and ensuring the reliable and sustainable functioning of the energy-driven Cloud–Edge Computing Continuum.

(c) Energy Profiles: Energy profiles is the detailed analysis and usage pattern of different actors (energy producers & consumers). Energy profiles are instrumental in understanding the energy usage and this leads to efficient energy management in the Cloud–Edge Continuum. The energy profiles can be optimized in distributed fashion, where each entity optimize their energy consumption or production profile according to their preference [131].

Energy profiles also include parameters such as

Power Usage Effectiveness (PUE), that is the ratio of the total energy used by Cloud–Edge Continuum to the energy used by computing infrastructure, Energy load distribution among different components of Cloud–Edge Continuum, energy environmental footprint, energy consumption effectiveness & trends and monitoring & management of energy.

(d) Energy Price Metrics: The energy price metrics in Cloud–Edge Continuum refers to structured data that provides crucial information regarding energy cost at different interval of time so that the energy consumption can be optimized. The energy cost varies depending on the energy source type (on-site energy, off-site energy, green or brown energy). The cost of these energy sources is based on carbon emission intensity & carbon taxes across different locations and energy prices variation throughout the day (on-peak and off-peak). The controller used these different metrics and knowledgeable decision to efficiently manage the energy consumption and cost in Cloud–Edge Continuum.

(vi) Controller: A controller, whether centralized or distributed, based on the MAPE-K (Monitoring, Analysis, Planning, Execution, and Knowledge) model [89, 132, 133], is essential to support resource provision, monitoring, and allocation in the Cloud–Edge Continuum. To develop interactions with the system, sensors (hardware-attached devices responsible for collecting data from various levels) and effectors (actuator devices used to enable or disable services through API calls) are applied. The monitoring module receives information regarding energy usage and resource utilization through these sensors.

The analysis module characterizes workloads based on multiple factors, including time sensitivity, resource intensity (e.g. compute, memory, data, network), location, and performance requirements. It utilizes cost models to calculate energy cost, carbon cost, energy wastage, and considers impact on climate. The planning module utilizes allocation policies to make scheduling decisions and analyzes the potential consequences of implementing changes in the system.

The execution module utilizes actuators to perform resource scheduling on the Continuum infrastructure. It employs proactive scheduling policies for multi-objective optimization. The optimization objectives and trade-off parameters are implemented and stored in the Knowledge pool of the MAPE-K model. Resource scheduling algorithms can be utilized to update the rules within the Knowledge pool. The proactive scheduling algorithms make use of the models stored in the Knowledge pool to forecast the supplied amount of energy, including both green energy and brown energy, as well as the expected resource usage.

6.2 Applicability of our solution

Based on our perspective model, we demonstrate the applicability of our proposed solution for scheduling Cloud–Edge Continuum workloads when considering appropriate energy sources, energy providers, infrastructures, and controllers (centralized and distributed).

We illustrate six different scenarios for delay-sensitive requests through a sequence diagram in Fig. 14. Here, the objective is to maximize the utilization of renewable energy in the federated Cloud–Edge Continuum. We categorise the scenarios based on the average communication cost to computation cost ratio (CCR) [62] for a given service request. By applying this cost metric, we can identify whether a given service request is computation intensive ($CCR < 1$), communication-intensive ($CCR > 1$), or ordinary ($CCR = 1$).

- In Scenario (I), through the End User Layer, a user submits a 'service request' to the Application Manager, which classifies it as 'communication-intensive (delay sensitive)'. The Application Manager forwards the request to Controller 1. Controller 1 analyzes that the 'available green energy is sufficient' in Cloud–Edge Continuum Infrastructure 1 (CE1) and sends a scheduling request to CE1 for 'local edge processing'. Once the request is completed, CE1 sends the result back to the End User Layer through the Application Manager.
- In Scenario (II), we assume that 'available green energy in CE1 is low', possibly due to an insufficient local renewable energy supply. In such cases, Controller 1 sends a 'processing request to Controller 2' located in another geographical region. If Controller 2 has 'sufficient available green energy', it responds to Controller 1, and the 'service request is migrated' to CE2.
- In Scenario (III), the Application Manager identifies the request as 'computation-intensive' and sends it to Controller 1. Controller 1 analyzes that the 'available green energy is low at the edge' and sends an 'offload to remote cloud request' to CE1 and the execution is performed at CE1.
- In Scenario (IV), if the 'available green energy is sufficient at the edge', then Controller 1 can send a 'local edge processing request' to CE1.

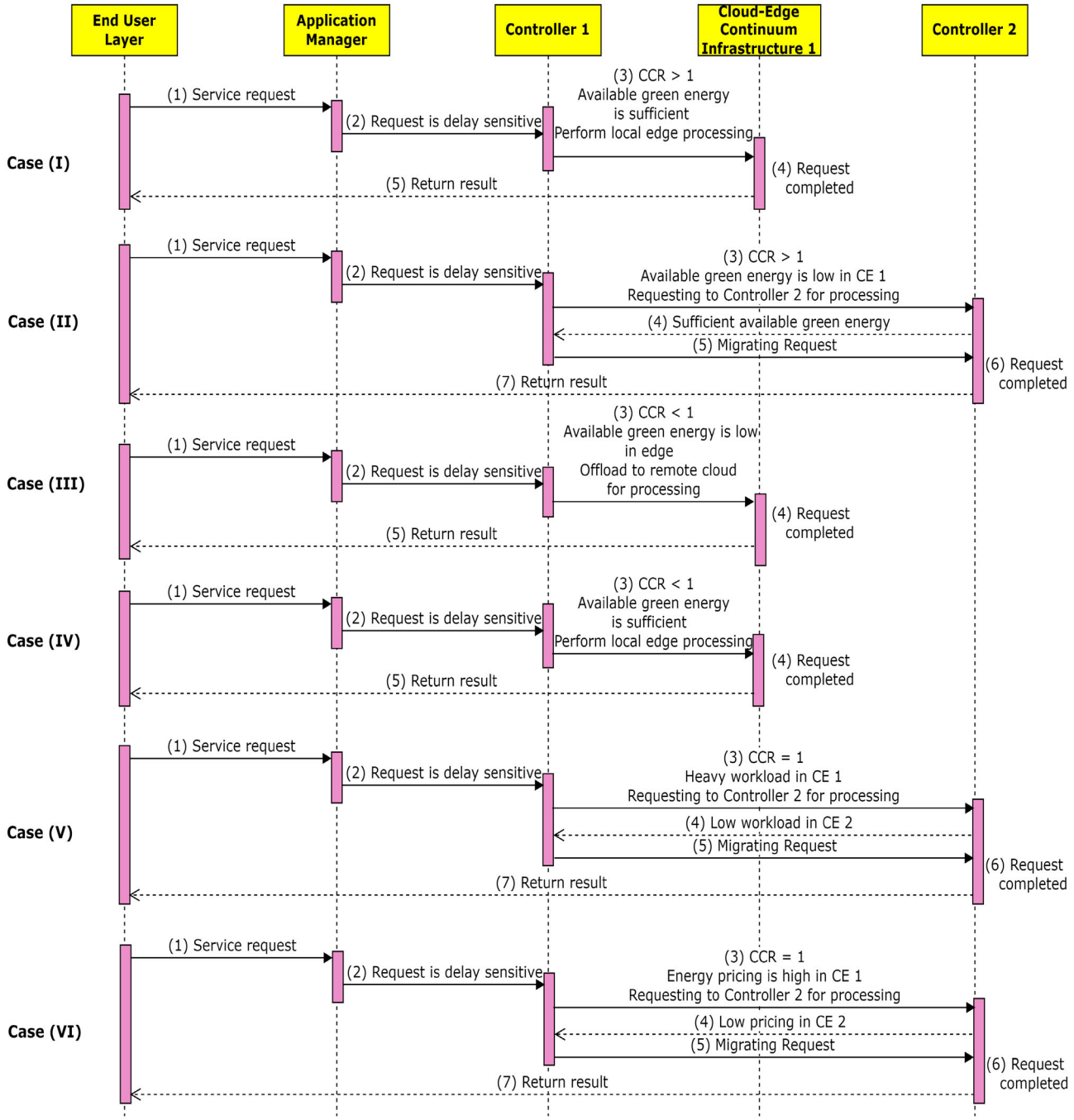


Fig. 14 Sequence diagram illustrating workflow execution for six scenarios in Sect. 6.2

- Scenario (V) shows the case of 'balancing energy consumption' for an ordinary request. Due to heavy workload in CE1, Controller 2 is requested for 'load distribution'. If 'workload is low in CE2', it can accept the 'migrating request' and complete the service execution.
- Scenario (VI) demonstrates how to handle energy pricing variations. As shown in Scenario (VI), Controller 1 identifies that the 'energy pricing is high in CE1'. Thus, it sends the processing request to Controller 2. If the 'energy pricing is low' in the CE2 region, then Controller 2 acknowledges and the 'request is migrated' to Controller 2 for execution.

6.3 Integrations: cloud bursting and edge caching

It is feasible to integrate our perspective model (with its emphasis on integration of energy considerations) with a number of existing cloud management techniques, such as cloud bursting [134] and edge caching [135]. *Cloud bursting* is utilised when local resources are exhausted; computation is burst (transferred) to alternative data centers (typically central Clouds rather than Edge resources). Our approach can be integrated when deciding which resources to move computation to during the "bursting" phase, whereby resources can be selected based in part on energy and sustainability decisions. Conversely, *Edge caching* is a technique that stores content and data in near edge resources (caches) to improve latency and speed up task execution. Integration with our method can again be used to make intelligent placement decisions at the network edge, balancing latency and task execution gains with overall energy efficiency and sustainability.

7 Conclusion

Energy consumption has become a critical issue as society's energy challenges grow, and is a serious concern for power grids which must balance the needs of clouds against other users. However, at present few formal models of federated Cloud–Edge systems exist—and none adequately represent and integrate energy considerations (e.g. multiple providers, renewable energy sources, pricing, and the need to balance consumption over large-areas with other non-Cloud consumers, etc.). This paper analyses how the modeling of Cloud, Cloud–Edge, and federated Continuum systems has been addressed in the literature, with a particular focus on the integration of energy concerns. Importantly, we identify key omissions in these models in terms of Continuum coverage, federation model, energy model, optimization objectives, technique, application, and evaluation. To model the green Cloud–Edge Continuum, we discuss several energy-considerations ensuring long-term viability in the rapidly evolving landscape of Cloud–Edge driven systems. We propose an initial high-level architecture and approach to begin addressing the research gaps, with the ultimate goal to develop a set of integrated models to provide a formal foundation for energy-aware Continuum management systems.

7.1 Future work

This paper lays the foundation for our vision of an energy-aware and sustainable Green Cloud–Edge continuum. To develop this further, there are a number of key steps that must be taken. We discuss three areas of focus for our upcoming future work in this area.

(i) *Developing an integrated set of models to enable formal reasoning over energy-aware Continuum systems*: This phase focuses on the creation of formal models for federated Cloud–Edge systems and workloads that incorporates energy sustainability, pricing, providers, and consumption. For the development of such models, we need to identify and categorize the different energy, network, hardware, and software components prevalent in Cloud–Edge systems into a series of interacting layers.

(ii) *Simulation and validation using a large-scale physical test-bed*: Once a comprehensive layered model of the energy-aware Cloud–Edge continuum exists, a natural progression is to use this to create a model-based simulation framework. This allows us to experiment with temporal and stochastic properties (latencies, renewable energy availabilities, resource utilisations, etc.) - the results of which can then be validated on physical test-beds. A physical test-bed may be implemented in a similar manner to the E2Clab platform [129].

(iii) *Autonomous resource management for Green Cloud–Edge Continuum systems*: Once a validated model-based simulation has been created (and released publicly), it will be possible to use this to develop and iteratively test novel energy-aware autonomous management approaches. We aim to do this with a focus on next-generation Continuum paradigms, particularly Serverless computing [136, 137]. As an example, there exists clear potential to integrate such a model-based simulation with Serverless runtimes developed as part of the SovereignEdge.COGNIT project [33].

Acknowledgements This work is partially funded by the Kempe Foundations (Kempestiftelserna) grant "De facto Center of Excellence in Autonomous Distributed Systems", and supported by the European Commission through the Horizon Europe project SovereignEdge.COGNIT (Grant 101092711) and Horizon 2020 project ASSISTANT (Grant 101000165).

Author contributions Yashwant Singh Patel: Conceptualization, Methodology, Writing, Experiments. Paul Townend: Supervision, Conceptualization, Methodology, Writing & Reviewing & Editing. Anil Singh: Conceptualization, Methodology, Writing, Experiments. Per-Olov Östberg: Supervision, Conceptualization, Methodology, Reviewing.

Funding Open access funding provided by Umea University.

Data availability Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical approval Not applicable

References

1. Montevecchi, F., Stickler, T., Hintemann, R., Hinterholzer, S.: Energy-efficient cloud computing technologies and policies for an eco-friendly cloud market. European Commission (2020)
2. Zhang, T., Gao, L., He, C., Zhang, M., Krishnamachari, B., Avestimehr, A.S.: Federated learning for the internet of things: applications, challenges, and opportunities. IEEE Internet Things Mag. **5**(1), 24–29 (2022)
3. Sun, J., Xu, M., Cespedes, M., Kauffman, M.: Data center power system stability-part I: power supply impedance modeling. CSEE J. Power Energy Syst. **8**(2), 403–419 (2022)
4. Jones, N.: How to stop data centres from gobbling up the world's electricity. Nature **561**(7722), 163–167 (2018)
5. Andrae, A.S., Edler, T.: On global electricity usage of communication technology: trends to 2030. Challenges **6**(1), 117–157 (2015)
6. Libertson, F., Velkova, J., Palm, J.: Data-center infrastructure and energy gentrification: perspectives from Sweden. Sustainability **17**(1), 152–161 (2021)
7. Renewable energy: a world turned upside down, the economist. https://www.economist.com/briefing/ 2017/02/25/a-world-turned-upside-down. Accessed 09 May 2023 (2017)
8. Yu, Z., Zhao, Y., Deng, T., You, L., Yuan, D.: Less carbon footprint in edge computing by joint task offloading and energy sharing. IEEE Netw. Lett. **1**, 1 (2023)
9. Electricity maps. https://app.electricitymaps.com. Accessed 01 Feb 2024 (2024)
10. How Microsoft's new datacenter region in Sweden incorporates the company's sustainability commitments. 2021. https://news.microsoft.com/europe/features/how-microsofts-newdatacenter-region-in-sweden-incorporates-the-companys-sustainabilitycommitments/. Accessed 01 Feb 2024 (2024)
11. Pahl, C., Azimi, S., Barzegar, H.R., El Ioini, N.: A Quality-Driven Machine Learning Governance Architecture for Self-Adaptive Edge Clouds. In: International Conference on Cloud Computing and Services Science, pp. 305–312 (2022)
12. Patel, Y.S., Townend, P., Östberg, P.O.: Formal Models for the Energy-Aware Cloud-Edge Computing Continuum: Analysis and Challenges. In: 2023 IEEE International Conference on Service-Oriented System Engineering (SOSE), pp. 48–59 (2023)
13. Khosravi, A., Andrew, L.L., Buyya, R.: Dynamic vm placement method for minimizing energy and carbon cost in geographically distributed cloud data centers. IEEE Trans. Sustain. Comput. **2**(2), 183–196 (2017)
14. Patel, Y.S., Jaiswal, R., Misra, R.: Deep learning-based multivariate resource utilization prediction for hotspots and coldspots mitigation in green cloud data centers. J. Supercomput. **78**(4), 5806–5855 (2022)
15. Patel, Y.S., Malwi, Z., Nighojkar, A., Misra, R.: Truthful online double auction based dynamic resource provisioning for multi-objective trade-offs in IaaS clouds. Clust. Comput. **24**, 1855–1879 (2021)
16. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., Jue, J.P.: All one needs to know about fog computing and related edge computing paradigms: a complete survey. J. Syst. Architect. **98**, 289–330 (2019)
17. Shao, X., Zhang, Z., Song, P., Feng, Y., Wang, X.: A review of energy efficiency evaluation metrics for data centers. Energy Build. **271**, 112308 (2022)
18. Reddy, V., Setz, B., Rao, G., Gangadharan, G., Aiello, M.: Metrics for sustainable data centers. IEEE Trans. Sustain. Comput. **2**(3), 290–303 (2017)
19. Luan, T. H., Gao, L., Li, Z., Xiang, Y., Wei, G., Sun, L.: Fog computing: focusing on mobile users at the edge. arXiv preprint arXiv:1502.01815 (2015)
20. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
21. Duc, T.L., Leiva, R.G., Casari, P., Östberg, P.O.: Machine learning methods for reliable resource provisioning in edge-cloud computing: a survey. ACM Comput. Surv. **52**(5), 1–39 (2019)
22. Taherizadeh, S., Jones, A.C., Taylor, I., Zhao, Z., Stankovski, V.: Monitoring self-adaptive applications within edge computing frameworks: a state-of-the-art review. J. Syst. Softw. **136**, 19–38 (2018)
23. Fog Computing: The Internet of Things: Extend the Cloud to Where the Things are. Cisco White Paper, 13 (2015)
24. Varghese, B., Wang, N., Nikolopoulos, D.S., Buyya, R.: Feasibility of fog computing, pp. 127–146. Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things (2020)
25. Aburukba, R.O., AliKarrar, M., Landolsi, T., El-Fakih, K.: Scheduling Internet of Things requests to minimize latency in hybrid Fog-Cloud computing. Future Gen. Comput. Syst. **111**, 539–551 (2020)
26. Ketu, S., Mishra, P.K.: Cloud, fog and mist computing in IoT: an indication of emerging opportunities. IETE Tech. Rev. **39**(3), 713–724 (2022)
27. Iorga, M., Feldman, L., Barton, R., Martin, M.J., Goren, N.S., Mahmoudi, C.: Fog computing conceptual model (2018)
28. López Escobar, J.J., Díaz Redondo, R.P., Gil-Castiñeira, F.: In-depth analysis and open challenges of Mist Computing. J. Cloud Comput. **11**(1), 81 (2022)
29. Kawaguchi, R., Bandai, M.: Edge based MQTT broker architecture for geographical IoT applications. In: 2020 IEEE International Conference on Information Networking (ICOIN), pp. 232–235 (January, 2020)
30. Soumplis, P., Kokkinos, P., Kretsis, A., Nicopolitidis, P., Papadimitriou, G., Varvarigos, E.: Resource Allocation Challenges in the Cloud and Edge Continuum. In: Advances in Computing, Informatics, Networking and Cybersecurity: A Book Honoring Professor Mohammad S. Obaidat's Significant Scientific Contributions, pp. 443–464. Cham: Springer (2022)
31. Townend, P., Looker, N., Zhang, D., Xu, J., Li, J., Zhong, L., Huai, J.: Crown-c: A high-assurance service-oriented grid

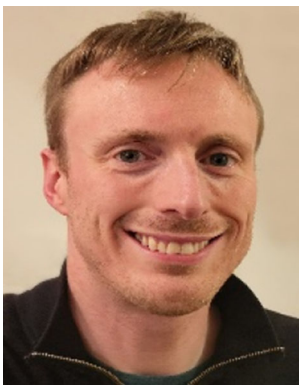
middleware system. In: 10th IEEE High Assurance Systems Engineering Symposium (HASE'07), pp. 35–44. IEEE (2007)
32. Moreschini, S., Pecorelli, F., Li, X., Naz, S., Hästbacka, D., Taibi, D.: Cloud Continuum: the definition. IEEE. Access **10**, 131876–131886 (2022)
33. Townend, P., et al.: COGNIT: Challenges and Vision for a Serverless and Multi-Provider Cognitive Cloud-Edge Continuum. In: 2023 IEEE International Conference on Edge Computing and Communications (EDGE) pp. 12–22. IEEE (2023)
34. Kitchenham, B.: Procedures for performing systematic reviews. Keele University, Keele, UK **33**, 1–26 (2004)
35. Carrera, D., Steinder, M., Whalley, I., Torres, J., Ayguade, E.: Autonomic placement of mixed batch and transactional workloads. IEEE Trans. Parallel Distrib. Syst. **23**(2), 219–231 (2012)
36. Gulati, A., Kumar, C., Ahmad, I.: Storage workload characterization and consolidation in virtualized environments. In: Workshop on Virtualization Performance: Analysis, Characterization, and Tools (VPACT), p. 4 (2009)
37. Summers, J., Brecht, T., Eager, D., Gutarin, A.: Characterizing the workload of a Netflix streaming video server. In: 2016 IEEE International Symposium on Workload Characterization (IISWC), pp. 1–12 (2016)
38. Xu, Y., Frachtenberg, E., Jiang, S., Paleczny, M.: Characterizing Facebook's Memcached Workload. IEEE Internet Comput. **18**(2), 41–49 (2014)
39. Liu, B., Lin, Y., Chen, Y.: Quantitative workload analysis and prediction using Google cluster traces. In: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 935–940 (2016)
40. Liu, C., Liu, C., Shang, Y., Chen, S., Cheng, B., Chen, J.: An adaptive prediction approach based on workload pattern discrimination in the cloud. J. Netw. Comput. Appl. **80**, 35–44 (2017)
41. Rodrigo, N., Calheiros, E.M., Ranjan, R., Buyya, R.: Workload prediction using ARIMA model and its impact on cloud applications' QoS. IEEE Trans. Cloud Comput. **3**(4), 449–458 (2015)
42. Kumar, J., Singh, A.K.: Workload prediction in cloud using artificial neural network and adaptive differential evolution. Future Gen. Comput. Syst. **81**, 41–52 (2018)
43. Calzarossa, M.C., Massari, L., Tessera, D.: Workload characterization: a survey revisited. ACM Comput. Surv. **48**(3), 43 (2016)
44. Rahmanian, A., Ali-Eldin, A., Skubic, B., Elmroth, E.: MicroSplit: Efficient Splitting of Microservices on Edge Clouds. In: 2022 IEEE/ACM 7th Symposium on Edge Computing (SEC), pp. 252–264. IEEE (2022)
45. Singh, A., Auluck, N., Rana, O., Jones, A., Nepal, S.: Scheduling real-time security aware tasks in fog networks. IEEE Trans. Serv. Comput. **14**(6), 1981–1994 (2021)
46. Tusa, F., Clayman, S.: End-to-end slices to orchestrate resources and services in the cloud-to-edge Continuum. Future Gen. Comput. Syst. **141**, 473–488 (2023)
47. Fu, K., Zhang, W., Chen, Q., Zeng, D., Peng, X., Zheng, W., Guo, M.: Qos-aware and resource efficient microservice deployment in Cloud-Edge Continuum. In: 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 932–941. IEEE (2021)
48. Ascigil, O., Phan, T.K., Tasiopoulos, A.G., Sourlas, V., Psaras, I., Pavlou, G.: On uncoordinated service placement in edge-clouds. In: 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 41–48. IEEE (2017)
49. Pop, P., Zarrin, B., Barzegaran, M., Schulte, S., Punnekkat, S., Ruh, J., Steiner, W.: The FORA fog computing platform for industrial IoT. Inf. Syst. **98**, 101727 (2021)
50. Etemadi, M., Ghobaei-Arani, M., Shahidinejad, A.: A cost-efficient auto-scaling mechanism for IoT applications in fog computing environment: a deep learning-based approach. Clust. Comput. **24**(4), 3277–3292 (2021)
51. Ullah, A., Dagdeviren, H., Ariyattu, R.C., DesLauriers, J., Kiss, T., Bowden, J.: Micado-edge: Towards an application-level orchestrator for the cloud-to-edge computing Continuum. J. Grid Comput. **19**, 1–28 (2021)
52. Kar, B., Yahya, W., Lin, Y.D., Ali, A.: Offloading using traditional optimization and machine learning in federated Cloud-Edge-fog systems: asurvey. IEEE Commun. Surv. Tutor. (2023)
53. Da Silva, D.M.A., Sofia, R.C.: A discussion on context-awareness to better support the IoT cloud/edge Continuum. IEEE Access **8**, 193686–193694 (2020)
54. Svorobej, S., Bendechache, M., Griesinger, F., Domaschka, J.: Orchestration from the Cloud to the Edge. The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing, pp. 61–77 (2020)
55. Kampars, J., Tropins, D., Matisons, R.: A review of application layer communication protocols for the IoT edge cloud Continuum. In: 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), pp. 1–6. IEEE (2021)
56. Baresi, L., Mendonça, D.F., Garriga, M., Guinea, S., Quattrocchi, G.: A unified model for the mobile-edge-cloud Continuum. ACM Trans. Internet Technol. **19**(2), 1–21 (2019)
57. Son, J., Buyya, R.: Latency-aware virtualized network function provisioning for distributed edge clouds. J. Syst. Softw. **152**, 24–31 (2019)
58. Zadara: Federated Edge - On-Demand Edge Cloud Services for MSPs. Retrieved April 25, 2023, from https://www.zadara.com/federated-edge/
59. Moreno-Vozmediano, R., et al.: BEACON: A cloud network federation framework. In: Advances in Service-Oriented and Cloud Computing: Workshops of ESOCC 2015, Taormina, Italy, September 15–17, (2015) Revised Selected Papers 4, pp. 325–337. Springer, Cham (2016)
60. Kubefed: Kubernetes Cluster Federation. Retrieved April 25, 2023, from https://github.com/kubernetes-sigs/kubefed
61. Saraswat, S., Gupta, H.P., Dutta, T., Das, S.K.: Energy efficient data forwarding scheme in fog-based ubiquitous system with deadline constraints. IEEE Trans. Netw. Serv. Manag. **17**(1), 213–226 (2020)
62. Li, W., Yang, T., Delicato, F.C., Pires, P.F., Tari, Z., Khan, S.U., Zomaya, A.Y.: On enabling sustainable edge computing with renewable energy resources. IEEE Commun. Mag. **56**(5), 94–101 (2018)
63. Jeong, Y., Maria, E., Park, S.: Towards energy-efficient service scheduling in federated edge clouds. Clust. Comput. **26**(5), 2591–2603 (2023)
64. Sharma, N., Ghosh, A., Misra, R., Das, S.K.: Deep meta Q-learning based multi-task offloading in edge-cloud systems. IEEE Trans. Mob. Comput. (2023)
65. 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 (2018)
66. Hasan, R., Hossain, M., Khan, R.: Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading. Future Gen. Comput. Syst. **86**, 821–835 (2018)
67. Guo, H., Liu, J.: Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks. IEEE Trans. Veh. Technol. **67**(5), 4514–4526 (2018)
68. Kaur, K., Garg, S., Kaddoum, G., Ahmed, S.H., Atiquzzaman: KEIDS: Kubernetes-based energy and interference driven scheduler for industrial IoT in edge-cloud ecosystem. IEEE Internet Things J. **7**(5), 4228–4237 (2019)

69. Yahya, W., Oki, E., Lin, Y.D., Lai, Y.C.: Scaling and offloading optimization in pre-CORD and post-CORD multi-access edge computing. IEEE Trans. Netw. Serv. Manag. **18**(4), 4503–4516 (2021)
70. Xu, X., Huang, Q., Yin, X., Abbasi, M., Khosravi, M.R., Qi, L.: Intelligent offloading for collaborative smart city services in edge computing. IEEE Internet Things J. **7**(9), 7919–7927 (2020)
71. Zhang, K., et al.: Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE Access **4**, 5896–5907 (2016)
72. Li, Z., Chang, V., Ge, J., Pan, L., Hu, H., Huang, B.: Energy-aware task offloading with deadline constraint in mobile edge computing. EURASIP J. Wirel. Commun. Netw. **2021**, 1–24 (2021)
73. Zhang, J., Shi, W., Zhang, R., Liu, S.: Deep reinforcement learning for offloading and shunting in hybrid edge computing network. In: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6. IEEE (2021)
74. Ahvar, E., Orgerie, A.C., Lebre, A.: Estimating energy consumption of cloud, fog, and edge computing infrastructures. IEEE Trans. Sustain. Comput. **7**(2), 277–288 (2019)
75. Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., Pineau, J.: Towards the systematic reporting of the energy and carbon footprints of machine learning. J. Mach. Learn. Res. **21**(1), 10039–10081 (2020)
76. Rastegar, S.H., Shafiei, H., Khonsari, A.: EneX: An Energy-Aware Execution Scheduler for Serverless Computing. IEEE Trans. Ind. Inf. (2023)
77. Aslanpour, M S., Toosi, A.N., Cheema, M.A., Gaire, R.: Energy-aware resource scheduling for serverless edge computing. In: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 190–199 (2022)
78. Angelelli, L., Da Silva, A A., Georgiou, Y., Mercier, M., Mounié, G., Trystram, D.: Towards a Multi-objective Scheduling Policy for Serverless-based Edge-Cloud Continuum. In: 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 485–497 (2023)
79. Cui, G., He, Q., Xia, X., Chen, F., Yang, Y.: Energy-efficient edge server management for edge computing: a game-theoretical approach. In: Proceedings of the 51st International Conference on Parallel Processing, pp. 1–11 (2022)
80. Iftikhar, S., Ahmad, M.M.M., Tuli, S., Chowdhury, D., Xu, M., Gill, S.S., Uhlig, S.: HunterPlus: AI based energy-efficient task scheduling for cloud-fog computing environments. Internet of Things **21**, 100667 (2023)
81. Wang, F., Jiao, L., Zhu, K., Lin, X., Li, L.: Toward Sustainable AI: Federated Learning Demand Response in Cloud-Edge Systems via Auctions. In: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, pp. 1–10 (2023)
82. Shen, H., Wang, H., Gao, J., Buyya, R.: An instability-resilient renewable energy allocation system for a cloud datacenter. IEEE Trans. Parallel Distrib. Syst. **34**(3), 1020–1034 (2023)
83. Cárdenas, R., Arroba, P., Risco-Martín, J.L., Moya, J.M.: Modeling and simulation of smart grid-aware edge computing federations. Clust. Comput. **26**(1), 719–743 (2023)
84. Perin, G., Berno, M., Erseghe, T., Rossi, M.: Towards sustainable edge computing through renewable energy resources and online, distributed and predictive scheduling. IEEE Trans. Netw. Serv. Manag. **19**(1), 306–321 (2021)
85. Ma, H., Huang, P., Zhou, Z., Zhang, X., Chen, X.: GreenEdge: joint green energy scheduling and dynamic task offloading in multi-tier edge computing systems. IEEE Trans. Veh. Technol. **71**(4), 4322–4335 (2022)
86. Yang, C.S., Huang-Fu, C. C., Fu, I.K.: Carbon-neutralized task scheduling for green computing networks. In: GLOBECOM 2022–2022 IEEE Global Communications Conference, pp. 4824–4829 (2022)
87. Chakraborty, C., Mishra, K., Majhi, S.K., Bhuyan, H.K.: Intelligent Latency-aware tasks prioritization and offloading strategy in Distributed Fog-Cloud of Things. IEEE Trans. Ind. Inf. **19**(2), 2099–2106 (2022)
88. Tuli, S., et al.: HUNTER: AI based holistic resource management for sustainable cloud computing. J. Syst. Softw. **184**, 111–124 (2022)
89. Xu, M., Toosi, A.N., Buyya, R.: A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing. IEEE Trans. Sustain. Comput. **6**(4), 544–558 (2020)
90. Ahvar, E., Ahvar, S., Mann, Z.Á., Crespi, N., Glitho, R., Garcia-Alfaro, J.: DECA: A dynamic energy cost and carbon emission-efficient application placement method for edge clouds. IEEE Access **9**, 70192–70213 (2021)
91. Aslanpour, M.S., Toosi, A N., Gaire, R., Cheema, M.A.: WattEdge: a holistic approach for empirical energy measurements in edge computing. In: Service-Oriented Computing: 19th International Conference, ICSOC 2021, Virtual Event, November 22-25, 2021, Proceedings 19, pp. 531–547 (2021)
92. Kar, B., Lin, Y.D., Lai, Y.C.: OMNI: Omni-directional dual cost optimization of two-tier federated Cloud-Edge systems. In: ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1–7 (2020)
93. Gu, L., Cai, J., Zeng, D., Zhang, Y., Jin, H., Dai, W.: Energy efficient task allocation and energy scheduling in green energy powered edge computing. Future Gen. Comput. Syst. **95**, 89–99 (2019)
94. Xu, C., Wang, K., Li, P., Xia, R., Guo, S., Guo, M.: Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning. IEEE Trans. Netw. Sci. Eng. **7**(1), 205–215 (2018)
95. Benzadri, Z., Belala, F., Bouanaka, C.: Towards a Formal Model for Cloud Computing. In: Service-Oriented Computing-ICSOC 2013 Workshops LNCS 8377, p. 381 (2014)
96. Nastic, S., Pusztai, T., Morichetta, A., Pujol, V.C., Dustdar, S., Vii, D., Xiong, Y.: Polaris scheduler: Edge sensitive and slo aware workload scheduling in Cloud-Edge-iot clusters. In: 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), pp. 206–216. IEEE (2021)
97. Zhang, X., Wu, T., Chen, M., Wei, T., Zhou, J., Hu, S., Buyya, R.: Energy-aware virtual machine allocation for cloud with resource reservation. J. Syst. Softw. **147**, 147–161 (2019)
98. Gao, J., Wang, H., Shen, H.: Smartly handling renewable energy instability in supporting a cloud datacenter. In: 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 769–778. IEEE (2020)
99. Minh, Q.N., Nguyen, V.H., Quy, V.K., Ngoc, L.A., Chehri, A., Jeon, G.: Edge computing for IoT-enabled smart grid: Future Energy. Energies **15**(17), 6140 (2022)
100. Garraghan, P., Ouyang, X., Townend, P., Xu, J.: Timely long tail identification through agent based monitoring and analytics. In: 2015 IEEE 18th International Symposium on Real-Time Distributed Computing, pp. 19–26. IEEE (2015)
101. Cai, L., Wei, X., Xing, C., Zou, X., Zhang, G., Wang, X.: Failure-resilient DAG task scheduling in edge computing. Comput. Netw. **198**, 108361 (2021)
102. Sonmez, C., Ozgovde, A., Ersoy, C.: EdgeCloudSim: An environment for performance evaluation of Edge Computing systems. In: 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), pp. 39–44. IEEE (2017)
103. Del-Pozo-Puñal, E., García-Carballeira, F., Camarmas-Alonso, D.: A scalable simulator for cloud, fog and edge computing

platforms with mobility support. Future Gen. Comput. Syst. **144**, 117–130 (2023)
104. Nan, Y., Li, W., Bao, W., Delicato, F.C., Pires, P.F., Dou, Y., Zomaya, A.Y.: Adaptive energy-aware computation offloading for cloud of things systems. IEEE Access **5**, 23947–23957 (2017)
105. Silva, D.A., R.A., da Fonseca, N.L., Boutaba, R.: Evaluation of the employment of UAVs as fog nodes. IEEE Wirel. Commun. **28**(5), 20–27 (2021)
106. Nikolov, V., Kächele, S., Hauck, F. J., Rautenbach, D.: Cloudfarm: An elastic cloud platform with flexible and adaptive resource management. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 547-553 (2014)
107. Desmeurs, D., Klein, C., Papadopoulos, A.V., Tordsson, J.: Event-driven application brownout: reconciling high utilization and low tail response times. In: 2015 International Conference on Cloud and Autonomic Computing, pp. 1–12 (2015)
108. Pandey, A., Moreno, G. A., Cámara, J., Garlan, D.: Hybrid planning for decision making in self-adaptive systems. In: 2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), pp. 130–139 (2016)
109. Hasan, M.S., Alvares, F., Ledoux, T., Pazat, J.L.: Investigating energy consumption and performance trade-off for interactive cloud application. IEEE Trans. Sustain. Comput. **2**(2), 113–126 (2017)
110. Xu, M., Buyya, R.: Brownout approach for adaptive management of resources and applications in cloud computing systems: a taxonomy and future directions. ACM Comput. Surv. **52**(1), 1–27 (2019)
111. Gao, J., Wu, J., Liu, J., Aremanda, V.V.: Green Energy Cloud - Taxonomy, Infrastructure, Platform, and Services. In: 2023 IEEE International Conference on Service-Oriented System Engineering (SOSE), Athens, Greece, pp. 182–190 (2023)
112. Kumari, A., Gupta, R., Tanwar, S., Kumar, N.: Blockchain and AI amalgamation for energy cloud management: challenges, solutions, and future directions. J. Parallel Distrib. Comput. **143**, 148–166 (2020)
113. Weron, R.: Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach. Wiley, New York (2007)
114. Jindal, A., Singh, M., Kumar, N.: Consumption-aware data analytical demand response scheme for peak load reduction in smart grid. IEEE Trans. Ind. Electron. **65**(11), 8993–9004 (2018)
115. He, Y., Jenkins, N., Wu, J.: Smart metering for outage management of electric power distribution networks. Energy Proc. **103**, 159–164 (2016)
116. Araya, D.B., Grolinger, K., ElYamany, H.F., Capretz, M.A., Bitsuamlak, G: An ensemble learning framework for anomaly detection in building energy consumption. Energy Build. **144**, 191–206 (2017)
117. Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar, N., Mishra, S.: Decision tree and SVM-based data analytics for theft detection in smart grid. IEEE Trans. Ind. Inf. **12**(3), 1005–1016 (2016)
118. Comello, S., Reichelstein, S.: Cost competitiveness of residential solar PV: The impact of net metering restrictions. Renew. Sustain. Energy Rev. **75**, 46–57 (2017)
119. Ausgrid. Time of Use Pricing. https://www.ausgrid.com.au/Your-energy-use/Meters/Time-of-use-pricing. Accessed 30 Jan 2024 (2024)
120. Lajevardi, B., Haapala, K., Junker, J.: An energy efficiency metric for datacenter assessment. IIE Annual Conference and Expo, pp. 1715–1722 (2014)
121. Docker documentation - docker documentation. https://docs.docker.com/
122. Production-grade container orchestration - kubernetes. https://kubernetes.io/
123. AWS Lambda. https://aws.amazon.com/pm/lambda/
124. Azure Functions. https://azure.microsoft.com/en-in/products/functions
125. Google Cloud Functions. https://cloud.google.com/functions
126. IBM Cloud Code Engine. https://cloud.ibm.com/docs/codeengine
127. Vmware - official site. https://www.vmware.com/
128. Open source software for creating private and public clouds. https://www.openstack.org/
129. Rosendo, D., Silva, P., Simonin, M., Costan, A., Antoniu, G.: E2clab: exploring the computing Continuum through repeatable, replicable and reproducible edge-to-cloud experiments. In: 2020 IEEE International Conference on Cluster Computing (CLUSTER), pp. 176–186 (2020)
130. Palensky, P., Dietrich, D.: Demand side management: demand response, intelligent energy systems, and smart loads. IEEE Trans. Ind. Inf. **7**(3), 381–388 (2011)
131. Stephant, M., Abbes, D., Hassam-Ouari, K., Labrunie, A., Robyns, B.: Distributed optimization of energy profiles to improve photovoltaic self-consumption on a local energy community. Simul. Model. Pract. Theory **108**, 102242 (2021)
132. Nallur, V., Bahsoon, R.: A decentralized self-adaptation mechanism for service-based applications in the cloud. IEEE Trans. Softw. Eng. **39**(5), 591–612 (2012)
133. Chen, T., Bahsoon, R.: Self-adaptive and online qos modeling for cloud-based software services. IEEE Trans. Softw. Eng. **43**(5), 453–475 (2016)
134. Mattess, M., Vecchiola, C., Garg, S.K., Buyya, R.: Cloud bursting: managing peak loads by leasing public cloud services. Cloud Comput. **1**, 343–368 (2011)
135. Liu, D., Chen, B., Yang, C., Molisch, A.F.: Caching at the wireless edge: design aspects, challenges, and future directions. IEEE Commun. Mag. **54**(9), 22–28 (2016)
136. Aslanpour, M.S., et al.: Serverless edge computing: vision and challenges. In: Proceedings of the 2021 Australasian Computer Science Week Multiconference, pp. 1-–10 (2021)
137. Nastic, S., Raith, P., Furutanpey, A., Pusztai, T., Dustdar, S.: A Serverless Computing Fabric for Edge & Cloud. In: 2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI), pp. 1–12 (2022)

Yashwant Singh Patel is currently a Postdoctoral Fellow in the Department of Computing Science, Umeå University, Sweden. He is also an Assistant Professor in the Computer Science and Engineering Department (CSED) at Thapar Institute of Engineering & Technology, Punjab, India. He received his PhD degree in Computer Science and Engineering from the Indian Institute of Technology Patna, India, in May 2021. He has authored more than 40 peer-reviewed papers and articles in high impact journals and internationally reputed conferences. His research interests include Cloud-

Edge Systems, Edge Intelligence, Serverless Computing, Energy-Efficient Computing, and Distributed Algorithms.



Paul Townend is Associate Professor at Umeå University, Sweden, where he is founder and head of the Green Distributed Computing research group. He is interested in energy efficient and sustainable distributed systems, with a current focus on Cloud-Edge, Edge Intelligence, and Data Centres. Paul has led/co-led over $5M in successfully completed research projects, authored over 60 peer-reviewed paper and articles, and served as General Chair for 15 IEEE international conferences.

Anil Singh received his B.Tech degree from Uttarakhand Technical University, Uttarakhand, India, his M.Tech degree from the National Institute of Technology Hamirpur, India, and his PhD degree from the Indian Institute of Technology Ropar, India. His research interests include scheduling and energy efficiency in cloud, fog, and edge computing. He is an assistant professor at Thapar Institute of Engineering and Technology, Patiala (India), and is currently working as a postdoctoral researcher at the Department of Computing, Umeå University (Sweden).

P-O Östberg is an Associate Professor at Umeå University and the founder and CTO of Biti Innovations, a spin-off from the Autonomous Distributed Systems Laboratory at Umeå University, Sweden. He has more than 15 years postgraduate experience of both academic research and industry work, and has held (visiting and staff) researcher positions at several universities including Uppsala University, Sweden, Karolinska Institutet, Sweden, Ulm University, Germany, and the Lawrence Berkeley National Laboratory at the University of California, Berkeley, USA. He has worked in the Swedish government's strategic eScience research initiative eSSENCE as well as in several high profile framework projects funded by the EU in the FP7 and H2020 programmes and the Swedish national research council (VR). His research interests are centered around distributed computing resource management and the use of machine learning, simulation, and optimization techniques to construct AI systems for planning and scheduling.