Alexandria Engineering Journal 91 (2024) 12–29

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Review

A comprehensive survey of energy-efficient computing to enable sustainable massive IoT networks

Mohammed H. Alsharif [a], Anabi Hilary Kelechi [b], Abu Jahid [c], Raju Kannadasan [d], Manish Kumar Singla [e,f], Jyoti Gupta [g], Zong Woo Geem [h,*]

[a] *Department of Electrical Engineering, College of Electronics and Information Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, the Republic of Korea*
[b] *Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA*
[c] *School of Electrical Engineering and Computer Science, University of Ottawa, 25 Templeton St., Ottawa, ON, K1N 6N5 Canada*
[d] *Department of Electrical and Electronics Engineering, Sri Venkateswara College of Engineering, Sriperumbudur 602117, India*
[e] *Department of Interdisciplinary Courses in Engineering, Chitkara University Institute of Engineering & Technology, Chitkara University, Rajpura 140401, India*
[f] *Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan*
[g] *School of Engineering and Technology, K.R. Mangalam University, Gurugram 122505, India*
[h] *College of IT Convergence, Gachon University, Seongnam 13120, the Republic of Korea*

ARTICLE INFO

Keywords:
Green communications
Energy efficiency
Energy harvesting
Computing
Green computing
Green IoT
Cloud computing
Fog computing
Edge computing

ABSTRACT

Energy efficiency is a key area of research aimed at achieving sustainable and environmentally friendly networks. With the rise in data traffic and network congestion, IoT devices with limited computational power and energy resources face challenges in analyzing, processing, and storing data. To address this issue, computing technology has emerged as an effective means of conserving energy for IoT devices by providing high-performance computing capabilities and efficient storage to support data collection and processing. As such, energy-efficient computing, or "green computing," has become a focal point for researchers seeking to deploy large-scale IoT networks. This study provides a comprehensive Survey of recent research efforts aimed at achieving energy-efficient computing and green computing for IoT networks. To the best of our knowledge, none of the studies in the literature have discussed all types of green computing (edge, fog, cloud) and their role in enabling massive IoT networks in terms of energy efficiency. The article starts with an overview of computing technologies and then goes with a discussion of the empowering energy-saving techniques for computing (edge, fog, and cloud) environments including, energy-aware architecture, data aggregation and compression, low-power hardware, energy-aware scheduling, task offloading, switching on/off unused resources, virtualization, energy harvesting, and cooling optimization. This article is an outline of a roadmap toward realizing the vision of a sustainable computing environment for massive IoT networks; in addition, open the door for interested researchers to follow and continue the vision of Energy-Efficient Computing.

1. Introduction

This section serves as a comprehensive introduction to the core elements of this paper, encapsulating several pivotal aspects. Beginning with Subsection 1.1, titled "Background and Motivations," this segment lays the foundation by providing contextual insights and the rationale behind the study's inception. Following this, Subsection 1.2, "Contributions," delineates the specific contributions this paper aims to make within its field, elucidating the novel perspectives or advancements it introduces. Finally, the section culminates with an outline of the paper's organization, offering a roadmap that delineates the subsequent sections and their respective focuses.

1.1. Background and motivations

Internet of Things (IoT) technology is crucial for enabling massive machine communications and supporting various smart applications, such as smart buildings, smart cities, smart vehicles, environmental sensing and forecasting, and disaster management [1,2]. The proliferation of IoT devices has led to a significant increase in the number of

* Corresponding author.
E-mail address: geem@gachon.ac.kr (Z.W. Geem).

https://doi.org/10.1016/j.aej.2024.01.067
Received 9 October 2023; Received in revised form 19 January 2024; Accepted 26 January 2024
Available online 6 February 2024

devices per square kilometer in massive IoT networks, reaching up to one million devices [3]. Consequently, the amount of data generated by these devices has grown exponentially and is expected to continue increasing in the future. The surge in data traffic and network congestion poses a significant challenge for IoT devices, which often have limited computational power. This limitation makes it difficult to analyze, process, and store the vast amount of data generated by IoT devices [4]. Additionally, most IoT devices rely on battery power, which can quickly deplete due to the high energy demands of computationally intensive applications that require low latencies and real-time responses [5]. Therefore, conserving or boosting device energy to maximize the finite energy supply of IoT devices is a crucial challenge that researchers aim to address in the coming years.

Computing technology is considered one effective way to conserve power supply for IoT devices. Computing technology provides high-performance computing capabilities and high-capacity storage to support data collection and processing in IoT networks [6]. Furthermore, Fog and edge computing devices can reduce the workload on cloud servers by performing computations closer to the end devices, which improves computation latency [7]. As a result, computing plays a crucial role in the operation of massive IoT networks, performing critical computation tasks for these networks. To conserve energy, task offloading is widely used, given that computing devices and applications are known to be highly energy-intensive. Here comes the need for energy-efficient computing or green computing. The primary objective of energy-efficient computing is to ensure that computing resources are used efficiently while guaranteeing reliability and scalability [8,9].

1.2. Comparison and our contributions

Energy efficiency in IoT networks has been extensively researched, with numerous studies focusing on algorithms and techniques across various network levels and components. Several survey papers have been published on energy efficiency topics, including green IoT networks [10–15], cloud computing [16–22], fog computing [23–27], edge computing [28–30], and energy harvesting techniques [31]. However, to the best of our knowledge, none of these studies have discussed all types of green computing (edge, fog, cloud) and their role in enabling massive IoT networks in terms of energy efficiency. Green edge computing (EC), green fog computing (FC), and green cloud computing (CC) are the key focus areas of this study, which presents a survey of recent works on achieving energy efficiency through computing technology for massive IoT networks. Furthermore, this paper provides recommendations and potential future directions for researchers working on energy-efficient computing. A comparison of related reviews, tutorials, and survey articles concentrating on green computing is in Table 1.

Thus, we summarize the main contributions of this work as follows:

- A comprehensive overview of recent research works in energy-efficient techniques of computing including EC, FC, CC.
- Compare and analyze the effectiveness of these energy-efficient techniques across different computing paradigms, and highlight similarities and differences in their adoption and impact on energy savings.
- Opening the door for interested researchers to follow and continue the vision of energy-efficient computing by exploring and perpetuating the vision of energy-efficient computing by delving into the latest advancements and techniques discussed in recent literature.

This paper focuses on recent trends in energy-efficient techniques for computing in order to enable sustainable massive IoT networks. Section 2 provides an overview of computing technologies (EC, FC, CC). Section 3 presents a survey of recent works done to achieve energy efficiency in computing. Finally, the conclusion and recommendations are given in Section 5.

2. Overview of computing technologies

Computing technologies have emerged as a key component in the IoT network landscape for enabling sustainable massive IoT networks. This is due to the fact that IoT devices have limited computational power and energy resources, and the increase in data traffic and network congestion has made data processing and analysis by these devices a challenging task. In order to address this challenge, energy-efficient computing technologies, such as EC, FC, and CC, have been developed to

Table 1
Comparison of related reviews, tutorials, and survey articles concentrating on green computing.

Reference		Architecture	Aggregation and compression	Low power hardware	Scheduling	Task offloading	Switching on/off	Virtualization	Energy Harvesting	Cooling
Edge Computing (EC)	[32]	√		√	√	√		√		
	[33]		√	√						
	[34]	√			√	√				
	[35]					√			√	
	[36]						√			
	[37]									√
Fog Computing (FC)	[27]	√				√	√		√	
	[38]				√					
	[39]					√			√	
	[40]		√							
	[41]							√		
	[42]			√						√
	[43]		√	√			√		√	
Cloud Computing (CC)	[44]	√		√						√
	[45]		√	√	√					
	[46]					√		√		
	[22]			√		√	√	√		
	[47]		√					√		√
	[48]								√	
	[49]				√					
	[50]	√			√			√		
All-To-All; This article covers ***All*** energy-efficient techniques in ***All*** computing (EC, FC, and CC).		√	√	√	√	√	√	√	√	√

enable efficient data processing and analysis in IoT networks [51]. Fig. 1 shows the three layers of computing architecture.

- ➢ EC is a distributed computing paradigm that enables the computation and data storage to be performed closer to the data source, thereby reducing latency and network congestion. EC devices are located at the edge of the network, and they are responsible for performing data processing and analysis tasks in real-time. This technology is particularly useful in applications where low latency is crucial, such as in autonomous vehicles or real-time monitoring of industrial processes. Furthermore, EC reduces the amount of data that needs to be transmitted to the cloud, which reduces network congestion and improves network efficiency. However, the limitations of EC are the limited processing power and storage capacity, which restrict the types of tasks that can be performed [52–55].
- ➢ FC is another distributed computing paradigm that enables the computation and data storage to be performed closer to the data source, similar to EC. The key difference between FC and EC is that FC devices are located closer to the cloud, typically at the network edge or at the base of the cell tower. FC devices have more processing power and storage capacity than EC devices, and they are responsible for performing data processing and analysis tasks that cannot be performed by EC devices. FC devices also enable the data to be processed and analyzed in real-time, which is particularly useful in applications where low latency is crucial [27,56,57].
- ➢ CC is a centralized computing paradigm that enables the computation and data storage to be performed in the cloud, which can be accessed by IoT devices from anywhere in the world. CC provides virtually unlimited processing power and storage capacity, and it is responsible for performing data processing and analysis tasks that cannot be performed by edge or FC devices. CC is particularly useful in applications where massive amounts of data need to be processed and analyzed, such as in social media or e-commerce applications. However, the limitations of CC are the high latency and network congestion, which restrict the types of tasks that can be performed in real-time [21,22,58].

The resources of the computing infrastructure in the IoT network are organized in a hierarchical manner, with cloud computing/servers having the largest amount of resources available, followed by FC /nodes that have a moderate amount of resources and act as an intermediary layer between CC and edge computing/devices, which have the least amount of resources [59,60]. In terms of IoT networks, the choice of computing technology depends on the specific application requirements. EC is suitable for applications that require low latency and real-time processing, but have limited processing power and storage capacity. FC is suitable for applications that require low latency and real-time processing, but have more processing power and storage capacity than edge computing. CC is suitable for applications that require massive amounts of data processing and analysis, but have high latency and network congestion [61]. Fig. 2 summarizes the challenges and opportunities of energy-efficient computing.

By understanding the methodologies, energy-saving potential, and challenges associated with these techniques, organizations can make informed decisions to optimize their computing infrastructure, reduce energy consumption, and improve overall efficiency in their operations. In the upcoming sections, we will discuss the various energy efficiency techniques and strategies employed in these three computing technologies to enhance the overall energy efficiency in IoT networks.

3. Empowering energy savings techniques for computing environments

This section explores strategies in EC, FC, and CC environments to optimize energy use without compromising performance. This section covers techniques like energy-aware architecture, data aggregation and compression, low-power hardware, scheduling, task offloading, switching on/off unused resources, virtualization, energy harvesting, and cooling optimization.

3.1. Energy-aware architecture

Energy-aware architecture is a key strategy for achieving energy efficiency in computing systems, aiming to minimize power

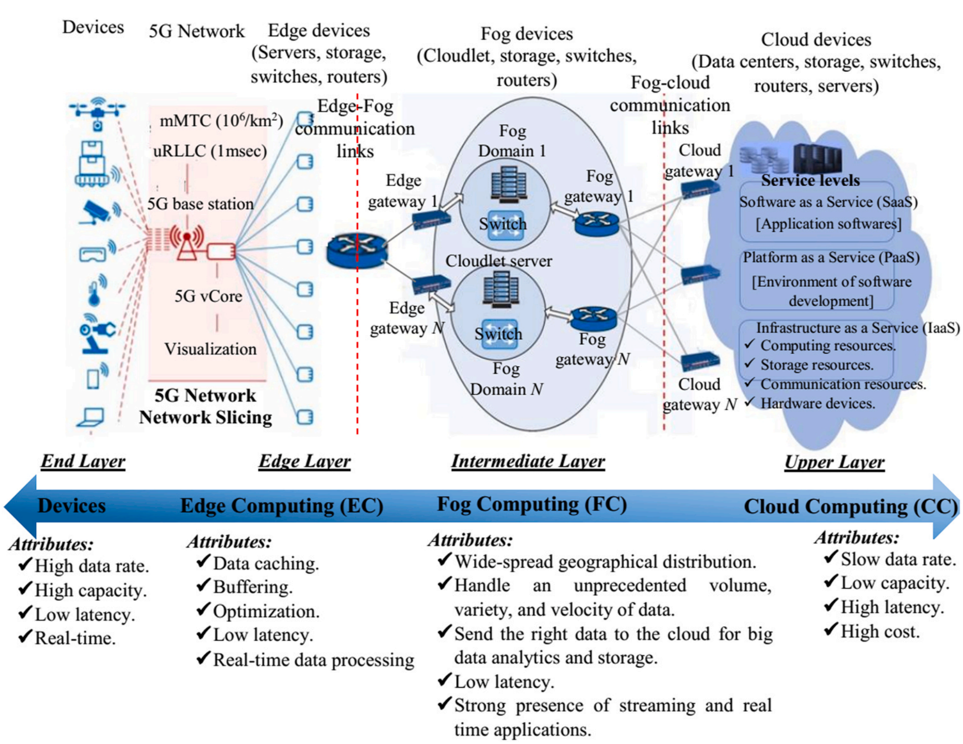

Fig. 1. Layers of computing architecture.

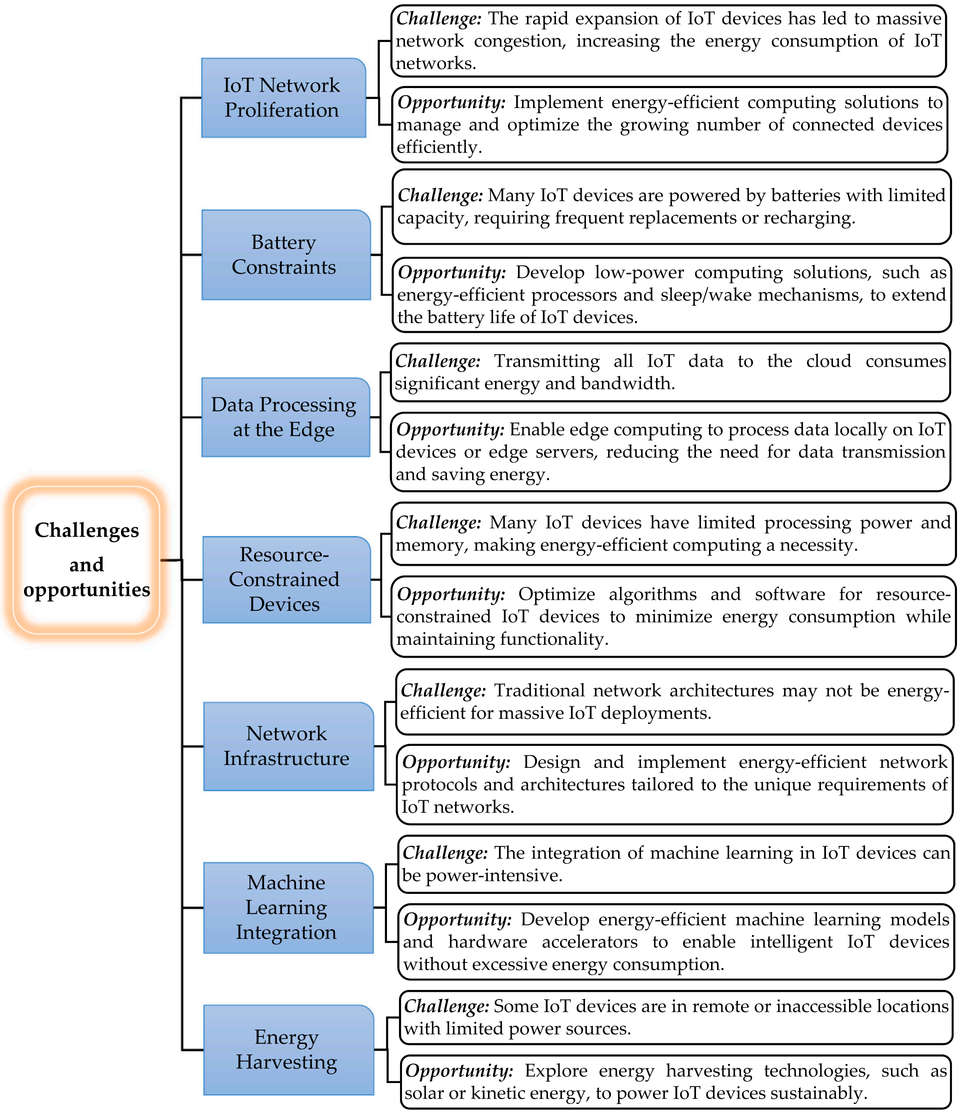

Fig. 2. Challenges and opportunities of Energy-Efficient computing to enable sustainable massive IoT networks.

consumption while maximizing performance. By designing and implementing architectures that prioritize energy optimization, it addresses the challenges of energy consumption in computing and promotes sustainable and environmentally friendly operations [62]. In general, energy-aware architecture can be categorized into three main types: cluster-based, centralized, and distributed.

- Cluster-based architecture involves grouping devices into clusters, where each cluster has a designated leader responsible for managing the energy consumption of the devices within the cluster. The leader device coordinates the energy optimization strategies and communicates with other clusters or higher-level management systems [63, 64].
- It is worth noting that the clustering architecture offers additional advantages such as scalability, extending the network's lifespan, reducing routing delays, decreasing network traffic, and facilitating effective channel access management [64].
- Centralized architecture centralizes the energy management control in a single entity or system. This central entity monitors and controls the energy usage of all devices in the system. It collects energy-related data, analyzes it, and makes energy optimization decisions based on the system's requirements and constraints [65,66].
- Distributed architecture distributes energy management control across multiple entities or devices within the system. Each device or entity independently manages its energy consumption based on local information and coordination with other devices or entities. Distributed energy-aware architecture promotes autonomy and flexibility in energy optimization decisions, allowing devices to adapt to local conditions and optimize their energy usage accordingly [27,67].

EC is based on distributed energy-aware architecture, where processing and storage are decentralized across the edge infrastructure. This approach offers advantages such as reduced data transmission energy, optimized resource utilization, and scalability [55]. By implementing energy-aware design principles, various elements within the EC environment can achieve energy savings: (i) edge devices are the foundation of the EC infrastructure. Using low-power hardware components, such as energy-efficient processors and power-saving sensors; power management techniques to adjust power usage based on workload demands, including dynamic voltage and frequency scaling; and additionally, energy-efficient communication protocols and algorithms implemented to reduce energy consumption during data transmission [68]. These measures contribute to significant energy savings, particularly for edge devices operating with limited energy resources within the distributed energy-aware architecture. (ii) Edge servers in EC play a crucial role in providing computational power and storage in proximity to edge devices. To optimize energy consumption, power management techniques such as CPU power scaling and disk spin-down are employed to minimize energy waste during idle periods. Additionally, resource allocation algorithms are implemented to ensure efficient utilization of computing resources, further enhancing energy efficiency [55]. These measures result in substantial energy savings, particularly for edge servers operating within the distributed energy-aware architecture and facing energy resource limitations. In addition, the implementation of energy-efficient routing protocols, data compression techniques, and traffic management strategies play a crucial role in minimizing data transmission volume and optimizing routing paths, in order to effectively reduce energy waste and enhance the overall energy efficiency of EC [69].

In FC, the clustered architecture has emerged as the most popular approach for achieving energy-efficient communications. This technique involves organizing sensor nodes into clusters of varying sizes based on factors like their location, density, and distance from the base station or fog node. Within each cluster, a capable sensor node is elected as the cluster head, responsible for managing communication within the cluster. The cluster member sensor nodes collect and transmit their data to the cluster head, which then aggregates the data and forwards it to the 5 G/B5G base station or fog node [27], as depicted in Fig. 3. To enhance the overall efficiency of the system, relay fog nodes are chosen based on their remaining energy and workload. However, selecting an efficient cluster head remains a significant challenge in improving energy efficiency. One approach, proposed by Sun et al. [70], establishes a communication hierarchy among cluster heads based on their proximity to the main fog node. Some cluster heads are selected as relays to assist in data transmission to the main fog node. The relay cluster head closest to the main fog node is designated as the network relay cluster head, facilitating data flow from sensor nodes to the main fog node through the cluster head, relay cluster head, and network relay cluster head. Cluster heads are rotated in each round to balance energy consumption among sensor nodes. In another study by Khalifeh et al. [71], sensor nodes are distributed into clusters, and the cluster head in each cluster is selected based on its remaining energy. Within a cluster, sensor nodes take turns serving as cluster heads to ensure fair energy consumption among them. Similarly, Rafi et al. [72] distribute sensors into clusters, allowing each sensor node an equal chance to become a cluster head. The selection criteria in this case depend on the node's workload, with the sensor node having the lowest workload chosen as the cluster head. Wang et al. [73] distribute sensor nodes into clusters and employ fog nodes to select the cluster head for each cluster. The selection is based on the node's residual energy and error-free communication history. Omoniwa et al. [74] propose the use of a mobile relay fog node positioned between sensor nodes and main fog node/cloud servers. The selection of the relay fog node is based on its ability to minimize transmission energy. Furthermore, the mobile relay fog node adjusts its location to enhance energy efficiency during transmission. Bozorgchenani et al. [66] utilize the clustered approach to propose a centralized and distributed architecture for task offloading among fog nodes (FNs), optimizing both energy and time. FNs are divided into two layers: the fog access point layer and the FN layer, based on their power supplies. Battery-operated FNs are categorized in the FN layer, while fixed FNs with electric power supply are placed in the fog access point layer. FNs in the FN layer are further classified as high-power FNs and low-power FNs, considering their energy levels. This classification is regularly updated to accommodate the most recent remaining energy levels of fog nodes. Low-power FNs offload their tasks to high-power FNs and fog access points, ensuring efficient task management. Silva et al. [75] identifies the ideal placement and resource setup for a fog node to deliver the highest level of service to end users. While, Wu et al. [76] discussed deploying computing servers in advance. Overall, these research studies demonstrate various approaches within the clustered architecture of WSNs to achieve energy efficiency and energy-awareness, employing strategies such as cluster head selection, relay nodes, workload and energy-based criteria, mobile relay fog nodes, and optimized task offloading among fog nodes. Table 2 presents a concise comparison of different energy-efficient architectures aiming to enhance power consumption in FC.

In CC, energy-aware architectures for CC can be either centralized or distributed, depending on the specific design and deployment

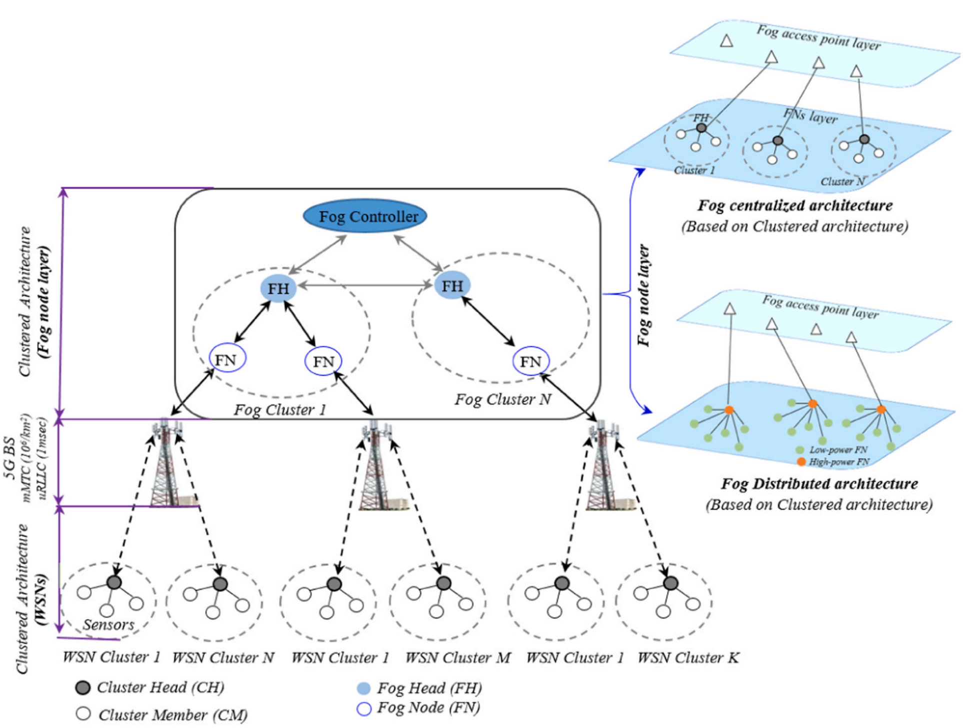

Fig. 3. Energy-aware FC Architectures.

Table 2
Comparison of energy-efficient architectures for improved power consumption in FC.

Reference	Architecture	Contribution	Energy savings
Sun et al. [70]	Clustered	Communication hierarchy among cluster heads based on proximity to the main fog node. Relay cluster heads assist in data transmission.	It is noteworthy that the energy savings achieved through different energy-efficient architectures can vary depending on various factors, including the specific implementation, network conditions, and application requirements.
Khalifeh et al. [71]		Cluster head selection based on remaining energy. Sensor nodes take turns as cluster heads for fair energy consumption.	
Rafi et al. [72]		Equal probability for sensor nodes to become cluster heads. The lowest workload determines cluster head selection.	
Wang et al. [73]		Fog node-based selection of cluster heads using residual energy and error-free communication history.	
Omoniwa et al. [74]		Mobile relay fog nodes between sensor nodes and fog/cloud servers. Selection based on transmission energy reduction.	
Bozorgchenani et al. [66]	Centralized	Centralized architecture for task offloading among fog nodes, optimizing energy and time.	
Silva et al. [75]	Distributed	Identifies the ideal placement and resource setup for a fog node to deliver the highest level of service to end users.	
Wu et al. [76]		Deploying computing servers in advance.	

considerations. While traditional CC architectures tend to be centralized, energy-aware architectures can adopt both centralization and decentralization strategies to optimize energy consumption [77]. In a centralized energy-aware architecture, the focus is on consolidating resources and centralizing control and management. This approach involves aggregating workloads onto a smaller number of high-capacity servers or data centers. By centralizing resources and management, it becomes easier to implement energy-saving techniques such as power management, workload consolidation, and efficient cooling strategies. Centralization allows for better coordination and optimization of energy usage across the entire infrastructure [78]. On the other hand, a distributed energy-aware architecture disperses computing resources across multiple geographically distributed locations. This approach involves deploying smaller data centers or EC nodes closer to the data sources or end-users. Distributed architectures are particularly suitable for latency-sensitive applications and scenarios where data needs to be processed and stored locally. By distributing resources, energy consumption can be minimized by reducing data transmission and reducing reliance on centralized infrastructure [79]. It's important to note that a combination of both centralized and distributed approaches can also be adopted in energy-aware cloud architectures. For example, centralizing management and control functions while distributing computing and storage resources to edge locations can offer a hybrid architecture that balances energy efficiency and performance [80]. At the end, the choice between centralization and distribution in energy-aware cloud architectures depends on factors such as workload characteristics, geographical distribution of users and data sources, latency requirements, and energy efficiency goals. The goal is to strike a balance between optimizing energy consumption, minimizing latency, and meeting application-specific requirements. Table 3 compares the centralized and distributed energy-aware architectures for CC.

Moreover, incorporating energy-efficient routing protocols alongside energy-efficient architectures can further enhance energy savings. Faheem et al. [81] introduced a Quality of Service (QoS) aware evolutionary cluster-based routing protocol (QERP) to overcome limitations of existing cluster-based routing techniques, particularly in terms of link reliability and energy consumption. By organizing network nodes into stable small clusters in a hierarchical order, QERP facilitates balanced traffic distribution and energy utilization, thereby improving the overall network lifespan. In the event of node failures, the QERP protocol adapts power adjustments and employs routing table information to identify the most suitable next-hop neighbors, resulting in low Bit Error Rate (BER). Alternatively, Su et al. [82] proposed the Deep Q-Network based energy and latency-aware routing protocol (DQELR). This protocol selects the optimal next-hop based on the maximum Q-value, taking into account node energy and latency. DQELR outperforms the Q-learning--based adaptive routing protocol (QELAR) proposed by Hu and Fei [83], thereby enhancing network performance. Furthermore, Khan et al. [84] introduced the multi-layered clustering-based energy-efficient routing protocol (MLCEE). Their protocol leverages Bayesian probability to select optimal routes based on residual energy, energy consumption rate, and link quality for each node. By dividing the network into layers, MLCEE achieves improved network lifespan, energy efficiency, and reduced end-to-end delay compared to depth-based routing protocols such as DBR (Yan et al. [85]) and EEDBR (Wahid and Kim [86]).

3.2. Data aggregation and compression

Data aggregation and compression are important techniques used in various computing environments, to optimize data transmission, improve network efficiency, and reduce storage requirements [87]. The following paragraphs provide a discussion of the Data aggregation and compression techniques. Moreover, a comprehensive summary is in Table 4.

Data aggregation involves combining data from multiple sources into a single entity, reducing the overall volume of data that needs to be processed, transmitted, or stored [88]. The following techniques are commonly used for data aggregation:

Table 3
Comparison of the centralized and distributed energy-aware architectures for CC.

Energy-Aware Architecture	Centralized	Distributed
Description	Consolidates resources and centralizes control and management.	Disperses computing resources across multiple locations.
Energy-Saving	TechniquesPower management, workload consolidation, efficient cooling strategies.	Reduced data transmission, localized processing and storage.
Advantages	Better coordination and optimization of energy usage.	Minimized data transmission energy, reduced reliance on centralized infrastructure.
Considerations	Workload characteristics, geographical distribution, latency requirements, energy efficiency goals.	Workload characteristics, geographical distribution, latency requirements, energy efficiency goals.

Table 4
Data aggregation and compression techniques used in various computing environments.

Technique		Advantages	Limitations	EC	FC	CC
Data Aggregation	Time-based Aggregation	• Reduced data volume. • Simplified data analysis. • Improved efficiency. • Reduced storage requirements.	• Loss of fine-grained details. • Potential loss of real-time insights. • Trade-off between accuracy and performance. • Inability to capture rapid changes.	√	√	√
	Spatial Aggregation	• Reduction of redundant data. • Minimization of data transmission. • Simplification of data analysis. • Enhanced spatial data processing.	• Potential loss of granular details. • Overlapping or intersecting spatial regions. • Increased computational complexity. • Trade-off between accuracy and aggregation.	√	√	√
	Statistical Aggregation	• Summarizes data with meaningful insights. • Reduces size, preserving key details. • Simplifies analysis and visualization.	• Loss of details and individual data points. • Distorting distribution or outliers. • Limited capture of specific data traits. • Balancing accuracy and generalization.	√	√	√
Data Compression	Lossless	• Zero data loss, perfect for crucial data fidelity. • Maintains integrity and originality. • Precise original data reconstruction.	• Less effective for some data types. • Requires more resources. • Processing speed may be slower.	√	√	√
	Lossy	• Superior compression ratios. • Drastically reduces data size. • Suited for multimedia with minor quality trade-offs.	• Not for precise detail preservation. • Irreversible compression. • May impact multimedia quality.			√
	Dictionary-based	• Ideal for repetitive patterns. • Higher ratios for redundant data. • Fast compression and decompression.	• Overhead in dictionary management. • Lower efficiency for less redundant data. • Complexity grows with dictionary size.			√
	Run-Length Encoding	• Ideal for consecutive repetitions. • Simple and fast. • Ideal for binary image data.	• Ineffective with non-repetitive data. • Higher storage needs. • Lossless but with limited compression.	√	√	√

i. Time-based Aggregation: Data can be aggregated based on time intervals, such as averaging or summing sensor readings over a specific time period. This reduces the data volume while preserving important information [89,90].
ii. Spatial Aggregation: Data from multiple sources within a specific spatial range can be combined, eliminating redundancy and reducing the number of data points. For example, in a smart city application, data from multiple temperature sensors in a neighborhood can be aggregated into a single representative value [90].
iii. Statistical Aggregation: Statistical methods such as mean, median, or mode can be used to aggregate data, providing summary information while reducing the data size. This approach is commonly employed in analytics and reporting tasks [91].

Data Compression: Data compression techniques aim to reduce the size of data by encoding it in a more compact representation. Various compression algorithms and methods can be utilized, including:

i. Lossless Compression: Lossless compression techniques reduce data size without losing any information during compression and decompression. Examples include algorithms like gzip, zlib, or ZIP, which are commonly used to compress files, documents, and structured data [92].
ii. Lossy Compression: Lossy compression techniques achieve higher compression ratios by selectively discarding less critical information. This is commonly used for multimedia data such as images, audio, and video, where slight quality degradation is acceptable. Popular lossy compression algorithms include JPEG for images and MP3 for audio [93].
iii. Dictionary-based Compression: Dictionary-based compression algorithms, like Lempel-Ziv-Welch (LZW) or Deflate, identify repeated patterns or sequences within data and replace them with shorter codes. This approach is efficient for compressing text data and is widely used in applications like file compression and network protocols [94].
iv. Run-Length Encoding (RLE): RLE is a simple compression technique that replaces consecutive occurrences of the same data value with a count and the value itself. It is effective for compressing data with long sequences of repeated values, such as binary images or raster graphics [95].

3.3. Low power hardware

Fig. 1 provides an overview of the computing environments, including edge, fog, and cloud, which comprise various devices such as switches, routers, storages, and servers. Moreover, CC involves data centers housing high-performance and energy-intensive super servers. To enhance energy efficiency in these environments, low-power hardware plays a crucial and vital role. By employing low-power hardware components, these computing paradigms can effectively reduce energy consumption, mitigate heat generation, and improve overall system efficiency [96]. The following paragraphs will delve into key aspects and benefits of low-power hardware in computing, while Table 5 summarizes its significance in computing environments.

EC involves processing data closer to the data source or end devices, typically at the network edge. Low-power hardware is particularly important in EC due to deploying a large number of devices across distributed locations. Low-power hardware enables cost-effective and scalable deployments by minimizing power requirements and infrastructure constraints [97]. In addition, edge devices are often battery-powered or have limited power sources. Low-power hardware components such as low-power processors, low-power sensors, and energy-efficient storage devices can extend the device's battery life and optimize energy consumption [98]. Moreover, edge devices are often deployed in physically constrained environments where heat dissipation can be challenging. Low-power hardware generates less heat, reducing the need for elaborate cooling solutions and enhancing device reliability [99].

FC extends EC by incorporating intermediate fog nodes between edge devices and the cloud. These distributed fog nodes aggregate and process data from multiple edge devices, as shown in Fig. 1. By leveraging low-power hardware, fog nodes can efficiently utilize available resources, optimize power consumption, and enhance the performance of fog-based applications. This is particularly crucial as fog nodes often operate with limited power sources. Low-power hardware components enable fog nodes to execute computational tasks while consuming minimal energy, ensuring efficient operation within resource-constrained environments [100]. Besides, fog nodes leverage low-power hardware components, including CPUs such as ARM-based processors or energy-efficient server CPUs, as well as low-power network interfaces and communication modules [101]. By utilizing these low-power components, fog nodes can achieve a balance between

Table 5
Summary of the significance of low-power hardware in computing environments.

Low-power hardware benefits	Computing environments		
	Edge	Fog	Cloud
Energy Efficiency	Optimizes energy consumption in resource-constrained edge devices and prolongs battery life.	Minimizes energy usage in power-limited fog nodes.	Incorporating low-power servers, and storages can significantly reduce energy consumption in data centers.
Thermal Management	Generates less heat, improves reliability in physically constrained edge.	Reduces cooling requirements for fog nodes.	Enabling data centers to employ more energy-efficient cooling systems, such as air-side or liquid cooling.
Scalability	Enables cost-effective and scalable deployments by minimizing power requirements.	Optimizes resource utilization in fog-based architectures.	Maximizes resource consolidation and VM.
Reliability and Redundancy	Enhances system reliability in edge deployments.	Improves availability and fault tolerance in fog nodes.	Contributes to the reliability of data center infrastructure

energy efficiency and sufficient processing capabilities for FC tasks. Low-power hardware can contribute to system reliability. Low-power components often have lower failure rates, reducing the chances of system disruptions and improving the overall availability of FC infrastructure [102].

CC relies on centralized data centers to provide scalable and resource-rich computing resources. Large-scale data centers consume significant amounts of power. By incorporating low-power hardware components such as energy-efficient servers, storage systems, and cooling mechanisms, data centers can reduce energy consumption, leading to cost savings and environmental sustainability [22]. Cloud servers equipped with low-power processors, such as those based on ARM architecture or specialized low-power server CPUs, consume less energy while delivering adequate performance for cloud workloads. In addition, low-power RAM modules and memory technologies, such as LPDDR4 or DDR4L, help reduce overall power consumption in server systems. Besides, utilizing low-power solid-state drives (SSDs) or energy-efficient hard disk drives (HDDs) reduces power consumption and contributes to energy-efficient storage systems [103]. Virtualization technologies play a vital role in cloud computing. Low-power hardware allows for more efficient consolidation of virtual machines (VMs) on physical servers, maximizing resource utilization, and reducing power consumption [104]. On the other hand, cooling is a major component of data center energy consumption. Low-power hardware generates less heat, enabling data centers to employ more energy-efficient cooling systems, such as air-side or liquid cooling, to maintain optimal temperatures [58].

On the other hand, the lighting system in the computing environment, particularly in data centers, consumes a significant amount of energy, constituting around 3% of the total energy requirements [105]. To improve energy efficiency, it is crucial to minimize the energy consumption of the lighting system. Data centers commonly use incandescent bulbs, compact fluorescent lightbulbs (CFLs), and LEDs as light sources. Table 6 provides a comparison of luminosity between LEDs, CFLs, and incandescent bulbs. Luminosity is a measure of the brightness rate, expressed in lm/W (lumens per watt). It is evident from the table that LEDs consume less energy in terms of wattage while maintaining the same luminosity levels. Various studies have consistently supported these findings, confirming the higher energy efficiency of LEDs compared to CFLs and incandescent bulbs. For instance, in 2010, LEDs demonstrated an energy efficiency of 100 lm/W, which further improved to 200 lm/W in 2015 [106]. Another study investigated the potential annual energy savings achievable with LEDs and found it to be approximately 22% [107]. These results highlight the significant energy-saving advantages offered by LEDs over other lighting options.

Table 6
Comparing Luminosity between CFL, LED and Incandescent bulbs.

Lumens (Brightness)	Incandescent (Watts)	CFL (Watts)	LED (Watts)
400 - 500	40	8-12	6-7
650 - 850	60	13-18	7-10
1000 - 1400	75	18-22	12-13
1450 – 1700 +	100	23-30	14-20
2700 +	150	30-55	25-28

3.4. Energy-aware scheduling

Energy-conscious scheduling stands as a pivotal component in the realm of resource administration within computing environments. Its primary objective lies in the allocation of computational tasks to various devices, all the while taking into account their energy limitations and performance prerequisites. Through the meticulous arrangement and possible reassignment of tasks, it becomes possible to curtail energy usage, ultimately resulting in heightened energy efficiency [108,109]. These scheduling techniques, aptly termed "Energy-Aware Scheduling," harness the capabilities of algorithms, machine learning, and the dynamic reallocation of resources to astutely orchestrate task scheduling. In doing so, they adeptly strike a balance between energy efficiency and performance optimization.

In the domains of both EC and FC environments, the energy-aware scheduling technique assumes a pivotal role in the optimization of energy utilization and the enhancement of the overall energy efficiency within the computing ecosystem. The primary goal of the energy-aware scheduling technique is to judiciously distribute computational tasks among edge devices or fog nodes, all while taking into careful consideration their energy constraints and performance prerequisites [110]. One prominent approach employed in energy-aware scheduling is task migration (Subsection 3.5), which involves the dynamic relocation of tasks between edge devices or fog nodes to achieve a balanced workload distribution and optimize energy consumption. By continuously monitoring the resource utilization of edge devices or fog nodes, tasks can be efficiently transferred from heavily burdened devices or nodes to those operating below capacity. This proactive approach serves to diminish energy consumption and enhance the overall system performance [111]. Furthermore, energy-aware scheduling may encompass the reassignment of computationally intensive tasks from edge devices or fog nodes to more potent resources like cloud servers. This strategic offloading helps minimize the energy consumption associated with computing tasks, a particularly advantageous tactic when edge devices and fog nodes possess limited processing power or energy resources [112]. It is noteworthy that machine learning algorithms and optimization techniques constitute indispensable components of energy-aware scheduling. These algorithms factor in diverse criteria, including the characteristics of tasks, the energy profiles of devices, and performance metrics, to make astute scheduling decisions. By generating optimal schedules that effectively balance energy efficiency and performance, these algorithms significantly contribute to proficient energy management within computing environments [111]. Moreover, machine learning algorithms have the capacity to forecast and anticipate future workload demands, facilitating the proactive and efficient allocation of resources while concurrently minimizing energy consumption [113].

In the realm of cloud computing, the energy-aware scheduling technique places its emphasis on several key aspects: (i) VM consolidation, (ii) dynamic resource allocation, (iii) load balancing, and (iv) predictive scheduling. These techniques are adept at enhancing the energy efficiency of CC systems by mitigating energy wastage and optimizing overall performance [111]. Load balancing entails the equitable distribution of workloads across multiple servers, aiming to achieve harmonious resource utilization and prevent server overload [114]. This even distribution of workloads serves to optimize energy consumption, as it averts the scenario in which a few servers are heavily burdened while others remain underutilized. Dynamic resource allocation, on the other hand, involves the responsive provisioning and deallocation of resources based on the current workload demands. Furthermore, this technique ensures that resources are efficiently relinquished when they are no longer needed, thereby curtailing energy usage. Additionally, considerable energy savings can be realized through the practice of virtual machine (VM) consolidation. This approach entails consolidating multiple VMs onto a reduced number of physical servers, thereby reducing the count of active servers and enhancing their resource utilization [114]. The consolidation of VMs aids in minimizing energy consumption by either powering off idle servers or placing them into low-power states. Moreover, the integration of machine learning algorithms proves instrumental in predicting and anticipating future workload demands. This foresight enables the proactive and efficient allocation of resources, with the overarching goal of minimizing energy consumption [113].

It is worth noting that an effective technique within the domain of energy-aware scheduling involves the adjustment of the operating voltage and frequency of processors in response to workload fluctuations. During periods of reduced activity, the lowering of voltage and frequency can lead to substantial energy savings. Conversely, in times of heightened workload, dynamic scaling allows processors to meet performance requirements while optimizing energy utilization. Chen et al. [115] optimized offloading decisions at fog nodes using Lagrangian dual theory and dynamic voltage scaling, enhancing energy efficiency. S. Hou et al. [116] employed dynamic voltage and frequency scaling on cloud servers to dynamically adjust voltage supply, frequency, and active server count, reducing overall energy consumption. Dynamic modulation scaling, adjusting transmitted bits based on packet requirements, considered the trade-off between transmission energy and delay. Karimiafshar et al. [117] combined dynamic voltage and frequency scaling with dynamic modulation scaling, optimizing CPU and transmission energy consumption. Wu et al. [76] proposed server deployment near fog nodes for task offloading, conserving energy and reducing latency and transmission power. Silva et al. [75] addressed FN location and resource planning to balance QoS and energy efficiency, considering computation, power, and storage resources. Optimal FN placement maximized user service while conserving FN energy. Table 7 provides a summary of dynamic power management: Frequency, Voltage, and Modulation approaches and benefits.

Table 7
Summary of dynamic power management: Frequency, Voltage, and Modulation approaches and benefits.

References	Techniques	Key Techniques	Benefits
Chen et al. [126]	Lagrangian dual theory and voltage scaling	Task offloading, voltage scaling	Improved energy efficiency in fog node computations
S. Hou et al. [109]	Dynamic voltage and frequency scaling	Voltage and frequency scaling	Energy reduction in cloud servers
Karimiafshar et al. [127]	Voltage and frequency scaling	Dynamic modulation scaling	Optimal energy consumption in fog node computations
Wu et al. [76]	Server deployment near fog nodes	Task offloading	Energy conservation and reduced transmission power
Silva et al. [75]	FN location and resource planning	Optimal resource allocation	Improved QoS and energy efficiency in the network

3.5. Task offloading

Task offloading is a strategy aimed at conserving energy by leveraging the assistance and resource sharing capabilities of idle or underutilized fog nodes for task execution. Task offloading can be implemented either partially or completely, taking into account both network latency and energy efficiency considerations [118]. In the context of IoT networks, which are hierarchically classified as CC, FC, and IoT devices based on resource availability (from maximum to minimum), three types of task offloading can be highlighted: (i) IoT devices to fog, (ii) Fog to fog, and (iii) Fog to cloud.

- *IoT devices to fog:* Offloading tasks directly to cloud servers from IoT devices can result in significant transmission energy consumption and unacceptable task processing delays due to the long distance. To overcome these limitations, IoT devices can offload their computational tasks to nearby FC nodes, which conserve the energy resources of IoT devices [119]. Several studies, such as [120–122] have addressed the task offloading challenges faced by IoT devices and demonstrated how this approach can conserve transmission energy and improve latency.
- *Fog to fog:* When fog nodes experience a heavy workload or have an excessive number of tasks, it is recommended to partially or completely relay offloading tasks to neighboring fog nodes with lighter workloads or to CC in order to conserve energy. However, relaying tasks to neighboring fog nodes is preferable due to lower transmission energy requirements and reduced latency for task processing [123]. On the other hand, relaying tasks to CC demands higher transmission energy and introduces greater latency, despite the larger resources available in the cloud compared to fog nodes. Zu et al. [124] employed a many-to-one matching technique to investigate energy savings in fog-to-fog offloading. This approach selects a fog node with the best channel conditions and the lowest energy consumption to relay task offloading for fog nodes burdened with heavy workloads. In another study, Zhang et al. [125] focused on offloading tasks for battery-powered fog nodes to ensure their long-term availability in the network. However, the selection of the optimal fog nodes for task offloading remains a significant challenge, requiring further research, particularly with the widespread adoption of AI and neural network techniques that may offer valuable insights for selecting fog nodes based on optimal channel conditions and energy consumption.
- *Fog to cloud:* When fog nodes face resource limitations, they can offload tasks to cloud computing, which offers a highly resourceful infrastructure with extensive storage for complex computations [126]. Hou et al. [116] have explored various utilization strategies for fog nodes, enabling energy efficiency and load balancing in data centers by offloading computations from fog nodes. However, due to the typically large distance between fog nodes and cloud computing, offloading tasks to the cloud incurs high transmission energy and latency, making it unsuitable for real-time applications [123]. Kim et al. [127] employed a probabilistic model to analyze task offloading from fog to cloud, taking into account latency considerations. The authors do not recommend offloading tasks from fog to cloud for applications that require low latency, as the transmission energy and latency involved in such scenarios are not conducive to real-time requirements.

However, when implementing task offloading methodology, certain factors (Fig. 4) need to be taken into consideration, including latency, task execution time, and the type of applications involved. Efficient task offloading requires careful consideration of specific conditions:

i. *Latency*: The latency involved in task offloading should not exceed the permissible threshold for timely task execution.

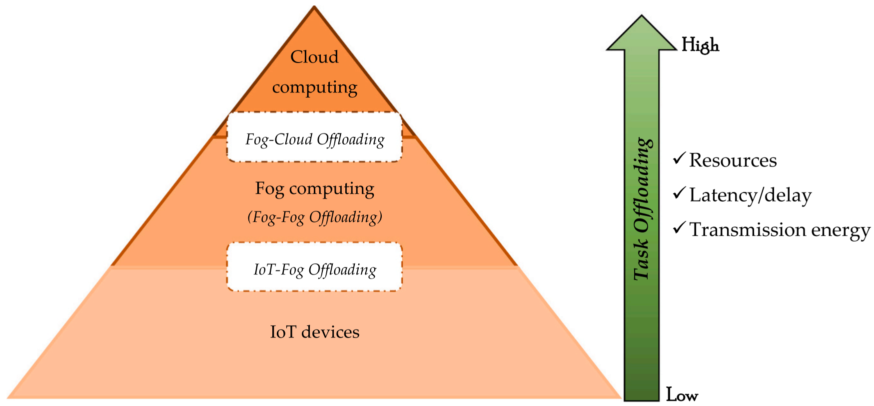

Fig. 4. Implementing task offloading methodology.

ii. *Energy Conservation*: Task offloading should result in higher energy conservation compared to the transmission energy required to transmit the offloaded task. The energy savings achieved through offloading should outweigh the energy consumed during transmission.
iii. *Real-Time Applications and Applications with Multiple Services*: Task offloading may not be recommended for real-time applications or applications that involve multiple services. These scenarios may have stringent requirements that make task offloading less suitable.

By considering these factors, the task offloading methodology can be implemented effectively, ensuring that the latency remains within acceptable limits, energy is conserved, and the specific requirements of real-time and multi-service applications are taken into account.

3.6. Switching on/off unused resources

Switching On/Off (sleep mode) techniques are energy-saving strategies that involve selectively powering down or putting devices into a low-power sleep mode when they are not actively used or required to perform tasks. This technique is employed in various computing environments, including EC, FC, and CC, to reduce energy consumption and optimize resource utilization. Detailed discussion is given as follows. In addition, Table 8 summarizes the criteria for switching On/Off techniques, highlighting their suitability for different computing environments.

i. *Criteria of Switching On/Off in Edge Computing Environment*

In EC, devices such as IoT devices, sensors, and small computing devices often operate on limited power sources, such as batteries. To maximize their energy efficiency, these devices can be configured to enter sleep mode during periods of inactivity. When a device is in sleep mode, it consumes significantly less power and conserves energy [62]. Sleep mode can be managed based on different criteria, such as time-based sleep or event-based sleep [9128]. By monitoring resource utilization in EC, network operators can gain visibility into the usage patterns of edge devices, identify potential bottlenecks, and optimize resource allocation. This monitoring can include metrics such as CPU usage, memory utilization, network bandwidth, and storage capacity. Real-time monitoring enables proactive management and maintenance of edge devices, ensuring efficient and reliable operation.

In time-based sleep, the device is programmed to enter sleep mode after a certain period of inactivity. This approach is suitable when the device has predictable patterns of activity and can estimate when it will not be used for a certain duration [128,129]. While time-based sleep mode in EC offers energy-saving benefits, it also has certain limitations or disadvantages that should be considered. Some of these limitations include:

1) Lack of Adaptability: Time-based sleep mode operates on a predetermined schedule, which may not always align with the actual device usage patterns. If the device's activity varies significantly or is unpredictable, time-based sleep mode may result in suboptimal energy savings. It may either put the device to sleep when it is still actively used or keep it awake when it could have been in a low-power state.
2) Inefficiency for Bursty Workloads: Time-based sleep mode may not be suitable for edge devices that experience bursty workloads or sporadic periods of activity. If the device enters sleep mode during an active burst of data processing or communication, it can cause delays and impact real-time or time-sensitive applications.
3) Delay in Responsiveness: When a device is in time-based sleep mode, it needs to wake up at predetermined intervals to check for any pending tasks or events. This wake-up process introduces a delay in responsiveness, as the device may need to transition from sleep mode to an active state before it can respond to requests. This delay can impact the real-time nature of certain applications and user interactions.
4) Lack of Contextual Awareness: Time-based sleep mode does not consider the context or environmental factors that may influence the device's energy needs. For example, if the device is in a high-activity environment or if there are critical events occurring, it may not be appropriate to enter sleep mode solely based on time. Context-awareness, such as considering the presence of user interactions or ongoing data streams, can help optimize energy savings more effectively.

On the other hand, event-based sleep involves triggering sleep mode based on specific events or conditions. For example, an IoT device may enter sleep mode when it has transmitted data to the cloud or when it detects no activity in its surroundings. Event-based sleep allows devices to conserve energy when they are not actively involved in data processing or communication [130]. While event-based sleep mode in EC offers energy-saving benefits, it also has certain limitations that should be considered. Some of these limitations include:

1) Difficulty in accurately predicting and configuring event-based sleep triggers: Identifying the appropriate events and conditions to trigger sleep mode can be challenging. Determining the optimal thresholds or criteria for event detection may require careful tuning and monitoring. Improper configuration can result in devices entering sleep mode too frequently or not entering sleep mode when necessary, leading to inefficient energy usage.

Table 8
Summary of the criteria for the Switching On/Off technique.

Computing	Criteria for sleep mode	Methodology	Limitations	Energy savings
Edge	Time-based sleep	Device enters sleep mode after a certain period of inactivity	• Limited adaptability to usage changes. • Inefficient for bursts or sporadic use. • Delayed response after sleep. • Lacks critical events.	20%–50%
	Event-based sleep	Device enters sleep mode based on specific events or conditions	• Relies on events, risking sleep inaccuracies. • May miss critical events in sleep. • Complex event-triggered sleep setup. • Potential increased power usage for constant detection.	
Fog	Resource Utilization Monitoring	Analyzing usage for identifying low-activity periods to power down resources.	• Complex resource monitoring overhead. • Difficulty in accurate dynamic measurement. • Privacy, security concerns in data collection. • Scalability challenges in growing fog systems. • Limited edge resource visibility. • Potential impact on fog device performance.	Up to 30%
	Duty Cycle Control	Adjusts the on and off time of the device transmitter to manage power consumption.	• Increased latency. • Trade-off between energy savings and performance. • Complexity of synchronization. • Impact on user experience. • Limited scalability. • Compatibility and standardization.	20%–30%
	Topology Control Protocols	Uses network redundancy to adjust the topology by reducing active nodes according to application needs.	• Overhead and complexity in protocol execution. • Lack of adaptability to dynamic network conditions. • Difficulty in achieving optimal network topology. • Dependence on accurate location information. • Limited scalability for large-scale fog environments.	Up to 50%
	BLE Interface Control	Conserves energy by selectively deactivating BLE interfaces until a BLE-advertiser is within range of a BLE-scanner.	• Limited range and coverage. • Signal degradation in crowded areas. • Constraints on data transfer rates and bandwidth. • Compatibility issues with different BLE versions. • Battery life limitations of BLE devices. • Security vulnerabilities. • Requires specific hardware and software. • Managing multiple BLE connections concurrently.	Up to 20%
Cloud	Server Power Management	Dynamically adjusting the power states of servers based on their workload.	• Overhead. • Complexity in Heterogeneous Environments. • Compatibility. • Reactive Nature. • Trade-off.	10%–30%
	VM Consolidation	Multiple virtual machines running on different physical servers are consolidated onto a smaller number of active servers.	• Resource contention. • Migration overhead. • Placement challenges. • Dependency and compatibility issues. • Limited flexibility. • Over-provisioning. • Performance isolation.	20%–50%
	Dynamic Resource Allocation	Involves dynamically provisioning and deallocating resources based on the current workload and demand.	• Resource fragmentation. • Increased management complexity. • Inefficient resource utilization. • Delay in resource allocation. • Potential performance drop in reallocating resources. • Limited control over allocation. • Difficulty in predicting resource demands.	10%–40%
	Dynamic Load Balancing	Plays a role in distributing the workload	• Overhead in load balancing	10%–30%

(*continued on next page*)

Table 8 (*continued*)

Computing	Criteria for sleep mode	Methodology	Limitations	Energy savings
		across multiple servers or data centers in a cloud environment.	algorithm execution. • Increased network communication overhead. • Possible performance drop in load balancing. • Difficulty predicting resource requirements. • Limited control over load balancing decisions. • Difficulty handling bursty workloads efficiently. • Scalability issues in large-scale clouds.	

2) Increased system complexity and overhead in managing event detection and sleep mode activation: Implementing event-based sleep requires additional software and hardware components to monitor and detect events. This introduces complexity to the system, requiring resource allocation and processing power for event detection. The overhead associated with managing event-based sleep can impact the overall system performance and responsiveness.
3) Possibility of missing important events if the sleep mode is activated too frequently or for extended periods: If the sleep mode is activated too frequently or for extended durations, there is a risk of missing critical events. This can result in delayed or incomplete data processing, missed communication opportunities, or the inability to respond to time-sensitive events. It is crucial to strike a balance between energy savings and maintaining the necessary level of responsiveness.
4) Limited applicability in scenarios where events are infrequent or unpredictable: Event-based sleep may not be as effective in scenarios where events occur infrequently or are highly unpredictable. If the system spends most of its time in sleep mode due to the lack of triggering events, the energy savings may not justify the added complexity and overhead. In such cases, alternative sleep strategies or energy-saving techniques may be more suitable.

The choice of sleep mode criteria depends on factors such as device usage patterns, event predictability, and the balance between energy efficiency and system responsiveness. Table 6 summarizes the criteria for Switching On/Off techniques, highlighting their suitability for different computing environments. While in sleep mode, devices can periodically wake up to perform necessary tasks or respond to events. This wake-up process can be triggered by internal timers, external events, or user interactions. Once the device wakes up, it resumes its normal operation and actively participates in data processing or communication tasks [62].

The energy savings achieved through Switching On/Off (sleep mode) techniques in EC can vary depending on the specific devices, usage patterns, and sleep mode configurations. On average, sleep mode can result in energy savings ranging from 20% to 50% or even higher in some cases. However, the actual percentage of energy savings will depend on factors such as the duration and frequency of sleep mode, the power consumption of the device in sleep mode, and the overall usage patterns of the device. It's important to note that energy savings can also be influenced by other factors such as the efficiency of the sleep mode implementation and the energy consumption of the device in its active state [62].

i. *Criteria of Switching On/Off in Fog Computing Environment*

Switching on/off unused resources in FC is an effective strategy for achieving energy savings. By selectively powering down or disabling idle or underutilized resources, such as servers, storage devices, or network components, significant energy reductions can be achieved. This approach helps to minimize power consumption and optimize resource utilization, leading to improved energy efficiency in FC environments [117]. The decision to switch off or on resources can be based on various factors. For example, a resource monitoring system can analyze resource utilization patterns and identify periods of low activity or inactivity. Based on this analysis, resources can be automatically powered down during off-peak times or when their utilization falls below a certain threshold [131]. Furthermore, intelligent power management techniques can be implemented to dynamically adjust resource power states based on demand. For instance, resources can be powered up or down in response to workload fluctuations, ensuring that the necessary computing power is available when required while minimizing energy consumption during periods of low demand. On the other hand, the authors in [132] have presented three methodologies: duty cycle control, passive wake-up triggered by external events or signals, and topology control protocols that leverage network redundancy to dynamically adapt the topology based on application requirements. These techniques necessitate proper regulatory procedures to ensure availability when needed. In the context of IoT networks where fog nodes play a crucial role, an interesting proposal by the authors in [133, 134] involves controlling the on/off mechanism using Bluetooth Low Energy (BLE) interfaces with the assistance of fog nodes. Devices utilize the BLE interface to connect and communicate, employing a methodology comprising BLE-advertisers, which advertise their information to be discovered, and BLE-scanners that scan for BLE-advertisers to provide services/information. It is important to note that continuous BLE advertisement and scanning can result in high device energy consumption. To address this, the fog node selectively turns off the BLE interfaces of devices until a BLE-advertiser enters the discoverable range of a BLE-scanner. The fog node possesses geolocation capabilities to track device location and movement. By considering the devices to be moving at a slow pace, the fog node estimates the optimal wake-up time for the device's Wi-Fi link based on device speed and direction. However, the proposed approach faces limitations in high-density scenarios as BLE interfaces would need to remain continuously 'on,' impacting energy efficiency. It is important to note that the switching on/off of resources should be carefully managed to avoid disrupting critical operations or causing delays when resources need to be powered back on. Proper resource management and control mechanisms should be in place to ensure that powered-off resources can be quickly and efficiently brought back online when their services are needed [134].

ii. *Criteria of Switching On/Off in Cloud Computing Environment*

Switching On/Off (sleep mode) techniques in CC environment involve the management of power states for servers and infrastructure components to optimize energy consumption. These techniques aim to minimize energy waste by selectively putting idle or underutilized resources into a low-power sleep mode [135]. Here, we will discuss the techniques of switching On/Off in CC in more detail.

- *Server Power Management:* Server power management techniques involve dynamically adjusting the power states of servers based on their workload. When the server is idle or has low utilization, it can be transitioned into a low-power sleep mode to conserve energy. When workload demands increases, the server can be woken up and powered on to handle the incoming requests. Power management algorithms and policies play a crucial role in determining when to switch servers On or Off to maintain a balance between energy savings and performance requirements [58,136]. The server power management technique involves: Dynamic Power Scaling, and Wake-On-LAN.

- Dynamic Power Scaling: This technique involves dynamically adjusting the power states of servers based on the current workload. By monitoring the utilization of resources such as CPU, memory, and disk, servers can be scaled up or down in terms of power consumption. For example, during periods of low utilization, servers can be put into a low-power sleep mode or have their power reduced to minimize energy consumption.
- Wake-On-LAN: Wake-On-LAN (WoL) is a network-based technique that allows a powered-off server to be remotely woken up when there is a demand for its resources. This technique enables servers to be in a sleep mode when not needed, saving energy while still being available for workload handling.

1) *Virtual Machine (VM) Consolidation:* VM consolidation is a technique where multiple virtual machines running on different physical servers are consolidated onto a smaller number of active servers. By consolidating VMs, the remaining idle servers can be powered off or put into sleep mode, resulting in energy savings. VM migration and placement algorithms are employed to determine the optimal VM placement and consolidation strategy, considering factors such as resource utilization, workload demands, and energy efficiency [137]. The VM consolidation technique involves: Live Migration, and Dynamic VM Placement.

- Live Migration: Live migration is a technique where running virtual machines are moved from one physical server to another without disrupting their operation. VM consolidation can be achieved by migrating VMs from underutilized servers to a smaller number of active servers. This consolidation reduces the number of powered-on servers and results in energy savings.
- Dynamic VM Placement: Dynamic VM placement algorithms consider factors such as resource utilization, workload demands, and energy efficiency to determine the optimal placement of VMs on physical servers. By intelligently placing VMs, idle servers can be powered off or put into sleep mode, leading to energy conservation.

3) *Dynamic Resource Allocation:* Dynamic resource allocation techniques involve dynamically provisioning and deallocating resources based on the current workload and demand. When the workload is low or during periods of inactivity, resources such as CPU, memory, and storage can be dynamically scaled down or turned off to save energy. As workload increases, resources can be dynamically provisioned and powered on to meet the demand. This approach ensures efficient resource utilization and energy savings in CC environments [136, 138]. The dynamic resource allocation technique involves: Resource Scaling, and Power-Aware Scheduling.

- Resource Scaling: This technique involves dynamically adjusting the allocated resources for virtual machines or containers based on their current demand. During periods of low workload, resources such as CPU, memory, and storage can be scaled down or powered off to conserve energy. As workload increases, resources can be dynamically provisioned and powered on to meet the demand.
- Power-Aware Scheduling: Power-aware scheduling algorithms consider both performance and power consumption factors when allocating resources to workloads. By intelligently scheduling tasks and assigning resources, the algorithms aim to maximize energy efficiency while meeting performance requirements.

4) *Load Balancing and Traffic Routing:* Load balancing and traffic routing techniques play a role in distributing the workload across multiple servers or data centers in a cloud environment. By intelligently routing traffic to the least loaded servers, it helps to evenly distribute the workload and prevent overutilization of resources. Load balancing algorithms can also consider energy consumption as a factor in decision-making, favoring servers with lower power consumption when assigning incoming requests [139]. The load balancing and traffic routing technique involves: Dynamic Load Balancing, and Energy-Aware Traffic Routing.

- Dynamic Load Balancing: Dynamic load balancing techniques distribute the workload across multiple servers or data centers based on their current capacity and resource utilization. Load balancers continuously monitor the load and redirect incoming requests to the least loaded servers, preventing resource overutilization and enabling better energy distribution.
- Energy-Aware Traffic Routing: Energy-aware traffic routing algorithms consider the energy consumption of servers or data centers as a factor in deciding the routing path for incoming requests. By favoring servers with lower power consumption, the algorithms aim to reduce energy waste and promote efficient energy usage in the cloud environment.

3.7. Virtualization

Virtualization serves as a foundational method employed across EC, FC, and CC environments to elevate resource utilization, enhance scalability, and optimize the overall efficiency of the systems. This process revolves around the creation of virtual instances mirroring physical resources like servers, storage, and networks, affording them the ability to be dynamically shared and allocated as the need arises [137]. Here are the virtualization techniques tailored for each of these computing environments:

i. In the domain of EC, virtualization empowers the establishment of virtual servers situated at the edge nodes. These virtual servers have the capacity to host multiple virtual machines (VMs) running diverse applications or services. By virtualizing the edge servers, a consolidation of resources is achieved, thereby fostering efficient resource utilization and scalability improvements, all while maintaining cost-effectiveness [140].
ii. FC extends the application of virtualization techniques to fog nodes, which function as intermediate computing entities bridging the gap between edge devices and the cloud. The virtualization of fog nodes facilitates the creation of virtual instances, endowing them with computing, storage, and networking capabilities. This, in turn, enables the effective allocation and management of resources within the fog infrastructure [141,142].
iii. Within the realm of cloud computing, infrastructure virtualization takes center stage as a pivotal technique that abstracts physical resources, such as servers, storage, and networks, into virtual counterparts. Virtual machines (VMs) are fashioned atop this virtualized infrastructure, permitting the concurrent execution of multiple workloads that efficiently share the underlying resources [143].

Virtualization in all three computing environments brings several benefits besides energy savings [142]:

1) Resource Optimization: Virtualization allows for the efficient utilization of resources by enabling multiple virtual instances to share physical resources, reducing hardware costs and improving resource allocation.

2) Scalability: Virtualization provides the ability to scale resources up or down dynamically based on workload demands, enabling flexible and elastic provisioning of computing resources.
3) Isolation and Security: Virtualization provides isolation between different virtual instances, enhancing security and preventing interference between workloads.
4) Mobility and Migration: Virtualization facilitates the mobility and migration of virtual instances between different physical hosts, enabling workload balancing, fault tolerance, and live migration of VMs.

3.8. Energy harvesting

Energy harvesting is a promising approach for achieving energy-efficient computing in various domains, including edge, fog, and cloud computing. By harnessing renewable energy sources, such as solar, wind, or kinetic energy, energy harvesting systems can generate power to supplement or replace traditional energy sources, reducing the reliance on fossil fuels and minimizing the environmental impact. Table 9 highlights the main key of the renewable energy sources (solar and wind), and kinetic energy.

One potential application of energy harvesting is in EC. Edge devices, located closer to the data source, can benefit from energy harvesting technologies to power their operations. For example, solar panels or small wind turbines can be integrated into edge devices to generate electricity and enable self-sustainability. This reduces the need for frequent battery replacements or wired power connections, making EC more flexible and cost-effective [144,145].

Similarly, energy harvesting can be applied in FC, which extends the capabilities of edge devices by aggregating and processing data at intermediate nodes. By utilizing renewable energy sources, fog nodes can operate autonomously and reduce their dependence on external power sources. This enhances the energy efficiency of FC systems and enables their deployment in remote or resource-constrained environments [146].

In cloud computing, energy harvesting can be employed at data centers to supplement the power requirements of the infrastructure. Large-scale solar arrays or wind farms can be integrated into the data center's energy grid, providing a renewable energy source to offset the energy consumption of the servers and cooling systems. This approach reduces the carbon footprint of CC and contributes to a more sustainable and energy-efficient operation [147].

To enable effective energy harvesting for energy-efficient computing, various technologies and techniques can be employed. These include efficient energy conversion mechanisms, energy storage systems, and power management algorithms. For example, maximum power point tracking (MPPT) algorithms can optimize the energy harvesting process by dynamically adjusting the load to maximize the power output from the renewable energy source. Additionally, energy storage systems, such as batteries or supercapacitors, can store excess energy generated during peak harvesting periods for later use when the energy supply is low [147].

Table 9
Comparison of renewable energy sources—solar, wind, and kinetic energy.

Renewable Energy Source	Solar Energy	Wind Energy	Kinetic Energy
Energy Source Availability	Sunlight Abundant, daytime hours	Wind Variable, depends on wind conditions	Motion Variable, depends on motion frequency
Environmental Impact	Minimal emissions, land use varies	Minimal emissions, land use varies	Minimal emissions, low land use
Energy Generation	Photovoltaic cells convert sunlight into electricity	Turbines convert wind into mechanical energy, then into electricity	Generators convert motion (vibration or movement) into electricity
Scalability	Scalable, from small-scale installations to large solar farms	Scalable, from small turbines to large wind farms	Scalable, from small-scale kinetic harvesters to industrial applications
Location Suitability	Suitable in areas with ample sunlight; rooftops, deserts, open spaces	Suitable in regions with consistent wind patterns; coastal areas, plains, elevated sites	Suitable in applications with frequent motion; industrial, wearable devices
Energy Storage	Requires energy storage (batteries) for nighttime or cloudy days	May require energy storage to manage intermittent wind	Often paired with energy storage to provide consistent power
Reliability	Highly reliable in sunny regions; less reliable in cloudy areas	Variable based on wind patterns; intermittent generation	Reliable in applications with consistent motion

3.9. Cooling optimization

Cooling optimization techniques in FC and CC environments focus on reducing energy consumption associated with cooling infrastructure while maintaining optimal operating temperatures [148]. Here are some details on these techniques and their energy-saving benefits. Moreover, Table 10 provides a summary of the cooling optimization techniques.

i. Thermal Management: Thermal management techniques aim to efficiently manage the heat generated by computing devices and infrastructure. This includes techniques such as temperature monitoring, heat dissipation mechanisms, and thermal insulation. By monitoring temperatures and optimizing cooling mechanisms, energy can be saved by ensuring that cooling resources are used only when necessary, reducing unnecessary cooling and energy waste [149].
ii. Adaptive Cooling Control: Adaptive cooling control techniques involve dynamically adjusting cooling mechanisms based on real-time temperature measurements and workload demands. This can include techniques such as variable fan speed control, adjusting airflow patterns, or utilizing intelligent cooling algorithms. By matching cooling efforts to the actual heat load and workload requirements, energy consumption can be optimized, reducing cooling-related energy costs [150].
iii. Airflow Management: Proper airflow management is crucial for efficient cooling in FC and CC environments. Techniques such as hot aisle/cold aisle containment, airflow redirection, and efficient rack layout design can optimize airflow and reduce the workload on cooling systems. By directing airflow efficiently and preventing hot and cold air mixing, cooling systems can operate more efficiently, resulting in energy savings [151].
iv. Free Cooling: Free cooling techniques leverage external ambient conditions, such as low outdoor temperatures, to reduce reliance on mechanical cooling systems. This can involve using outdoor air for cooling purposes, using economizers to take advantage of cooler outside air, or employing water-based cooling systems that utilize natural cooling sources. By reducing the need for mechanical cooling, energy consumption is significantly reduced, resulting in substantial energy savings [152].
v. Thermal-Aware Resource Allocation: Thermal-aware resource allocation techniques consider the thermal profiles of computing devices and distribute workloads in a way that minimizes heat generation and cooling requirements. By intelligently placing workloads on devices with lower thermal loads, the overall cooling demand can be reduced, leading to energy savings [153].

Table 10
Cooling optimization technique in fog and cloud computing.

Cooling Technique	Description	Suitability (Fog/Cloud)	Limitations
Thermal Management	Efficient heat management using temperature monitoring, heat dissipation, and insulation techniques.	Both	Limited effectiveness in high-density computing environments.
Adaptive Cooling Control	Dynamically adjusting cooling mechanisms based on real-time temperature and workload demands.	Both	Requires sophisticated control algorithms and real-time monitoring infrastructure.
Airflow Management	Optimizing airflow through techniques like containment, redirection, and efficient rack layout.	Both	Limited impact in environments with poor airflow design or high equipment density.
Free Cooling	Utilizing external ambient conditions for cooling, such as outdoor air or water-based cooling systems.	Both (More applicable to cloud)	Limited applicability in regions with extreme climate conditions.
Thermal-Aware Resource Allocation	Distributing workloads based on thermal profiles to minimize heat generation and cooling requirements.	Both	Relies on accurate thermal profiling and workload characterization.
Efficient Data Center Design	Optimized data center layout, rack placement, and energy-efficient cooling equipment.	Both (Design considerations vary)	Limited applicability for existing data centers with space and infrastructure constraints.

4. Conclusion and future outlook

This article outlines a roadmap for creating a sustainable computing environment tailored for massive IoT networks. Energy-efficient computing serves as a solution to the challenges stemming from limitations in battery-powered IoT devices and the need for efficient data processing. These challenges also offer opportunities for innovation and collaboration across diverse domains. Through optimizing hardware, software, and network infrastructure, we can curtail energy consumption, prolong IoT device battery life, and minimize the environmental impact of extensive IoT deployments. Integration of machine learning, AI, and energy harvesting technologies is expected to be crucial in this evolution. Looking ahead, the future of energy-aware computing envisions a fusion of advanced technologies, including:

- Convergence of AI and computing: The convergence of Artificial Intelligence (AI) and Fog computing stands as a transformative force. Future developments will heavily rely on AI-driven optimizations, enabling advanced decision-making at the edge. Machine learning models will dynamically allocate resources, manage energy consumption, and optimize task offloading strategies. These AI-driven optimizations are expected to enhance the efficiency and responsiveness of computing.
- Advancements in energy harvesting: Future research will delve deeper into energy harvesting techniques, focusing on novel methods to harness ambient energy sources. The integration of hybrid energy harvesting, combining solar, wind, and mechanical energy, will be explored to boost the sustainability of Fog infrastructure. Additionally, advancements in energy storage technologies will complement these harvesting techniques, ensuring consistent and reliable power sources for Fog nodes.
- Resource management: Ensuring energy-conscious resource utilization entails efficient resource management and dynamic task offloading. The challenge is judiciously using computing resources amidst variable IoT traffic patterns to prevent energy waste. While load-balancing techniques seem promising for resource optimization, their efficacy can vary with network loads. The future outlook in resource management and load balancing requires scalable and cost-effective strategies adaptable to computing capabilities and energy efficiency. Striking a balance in task distribution among computing resources, considering their capabilities and energy efficiency, is crucial for optimal resource utilization while reducing energy consumption in computing environments.
- Trade-off Energy Efficiency and QoS-Awareness: Energy-conscious methods in distributed computing have a substantial impact on Quality of Service (QoS). Approaches such as device consolidation to save energy affect QoS by introducing latency during IoT traffic rerouting, potential dropped requests, and delays in activating additional fog devices. Maintaining a balance between device activation and deactivation is pivotal to prevent undue QoS repercussions. Furthermore, methods like Dynamic Voltage and Frequency Scaling (DVFS) aimed at energy savings might influence the performance of IoT applications due to uncoordinated adjustments in processor frequency. Future endeavors should focus on devising energy-efficient strategies while upholding QoS in IoT environments based on Computing.
- Choose an optimal energy-efficient approach: Choosing the right approach is essential to achieve optimal energy savings without compromising the performance of the IoT network. Energy consumption in computing environments typically consists of static and dynamic energy consumption. Static energy consumption occurs when a device is powered on but idle, whereas dynamic energy consumption occurs during the processing of IoT requests. Typically, static energy consumption exceeds dynamic energy consumption. Determining the energy-efficient technique and deciding whether to target static or dynamic energy consumption poses a challenge due to the diverse environment and service-level agreements of IoT applications. Given that dynamic energy consumption is the primary part of total energy consumption, it's imperative to explore novel techniques that specifically address dynamic energy consumption.

Funding

This research was supported by the Energy Cloud R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (2019M3F2A1073164). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2020R1A2C1A01011131).

CRediT authorship contribution statement

Conceptualization, **M.H.A., A.H.K.** and **A.J.**; Methodology, **M.H.A.** and **R.K.**; Resources, **M.K.S.** and **J.G.**; Data curation, **M.H.A.** and **A.H.K.**; Writing – original draft, **M.H.A.** and **A.H.K.**; Writing – review & editing, **A.J., R.K., M.K.J., J.G.**, and **Z.W.G.**; Funding acquisition, **Z.W.G.** All authors have read and agreed to the published version of the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P.I.V. Padmanaban, M. Shanmugaperumal Periasamy, P. Aruchamy, An energy-efficient auto clustering framework for enlarging quality of service in Internet of Things-enabled wireless sensor networks using fuzzy logic system, Concurr. Comput. Pract. Exp. 34 (2022) e7269.

[2] M.H. Alsharif, A. Jahid, R. Kannadasan, M.-K. Kim, Unleashing the potential of sixth generation (6G) wireless networks in smart energy grid management: a comprehensive review, Energy Rep. 11 (2024) 1376–1398.

[3] M.H. Alsharif, A. Jahid, A.H. Kelechi, R. Kannadasan, Green IoT: a review and future research directions, Symmetry 15 (2023) 757.
[4] J. Ren, D. Zhang, S. He, Y. Zhang, T. Li, A survey on end-edge-cloud orchestrated network computing paradigms: transparent computing, mobile edge computing, fog computing, and cloudlet, ACM Comput. Surv. (CSUR) 52 (2019) 1–36.
[5] M.H. Alsharif, M.S. Hossain, A. Jahid, M. Khan, B.J. Choi, S.M. Mostafa, Milestones of wireless communication networks and technology prospect of next generation (6G), Comput., Mater. Contin. 71 (2022) 1376–1398.
[6] H. Li, K. Ota, M. Dong, Learning IoT in edge: deep learning for the Internet of Things with edge computing, IEEE Netw. 32 (2018) 96–101.
[7] C. Lin, G. Han, X. Qi, M. Guizani, L. Shu, A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks, IEEE Trans. Veh. Technol. 69 (2020) 5481–5493.
[8] M.H. Alsharif, Comparative analysis of solar-powered base stations for green mobile networks, Energies 10 (2017) 1208.
[9] M.H. Alsharif, R. Nordin, M. Ismail, Classification, recent advances and research challenges in energy efficient cellular networks, Wirel. Pers. Commun. 77 (2014) 1249–1269.
[10] C. Kapoor, H. Singh, V. Laxmi, A survey on energy efficient routing for delay minimization in iot networks, Int. Conf. Intell. Circuits Syst. (ICICS) 2018 (2018) 320–323.
[11] S. Popli, R.K. Jha, S. Jain, A survey on energy efficient narrowband internet of things (NBIoT): architecture, application and challenges, IEEE Access 7 (2018) 16739–16776.
[12] W. Yaïci, K. Krishnamurthy, E. Entchev, M. Longo, Survey of internet of things (IoT) infrastructures for building energy systems, Glob. Internet Things Summit (GIoTS) 2020 (2020) 1–6.
[13] A. Jangid, P. Chauhan, A survey and challenges in IoT networks, Int. Conf. Intell. Sustain. Syst. (ICISS) 2019 (2019) 516–521.
[14] G. Verma, S. Prakash, A study towards current trends, issues and challenges in internet of things (IoT) based System for intelligent energy management, 4th Int. Conf. Inf. Syst. Comput. Netw. (ISCON) 2019 (2019) 358–365.
[15] P. Gupta, S. Bharadwaj, V.K. Sharma, A survey to bridging the gap between energy and security in IoT and home, in: Proceedings of the Fifth International Conference on Image Information Processing, (ICIIP) 2019 (2019) 379–384.
[16] W.K. Hasan, Y. Ran, J. Agbinya, G. Tian, A survey of energy efficient IoT network in cloud environment, Cybersecur. Cyber Conf. (CCC) 2019 (2019) 13–21.
[17] X. You, X. Lv, Z. Zhao, J. Han, X. Ren, A survey and taxonomy on energy-aware data management strategies in cloud environment, IEEE Access 8 (2020) 94279–94293.
[18] M.M. Mahmoud, J.J. Rodrigues, K. Saleem, Cloud of Things for healthcare: a survey from energy efficiency perspective, Int. Conf. Comput. Inf. Sci. (ICCIS) 2019 (2019) 1–7.
[19] S. Puhan, D. Panda, B.K. Mishra, Energy efficiency for cloud computing applications: a survey on the recent trends and future scopes, Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA) 2020 (2020) 1–6.
[20] T. Vishrutha, P. Chitra, A survey on energy optimization in cloud environment, IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC) 2018 (2018) 1–5.
[21] S. Bharany, S. Badotra, S. Sharma, S. Rani, M. Alazab, R.H. Jhaveri, et al., Energy efficient fault tolerance techniques in green cloud computing: a systematic survey and taxonomy, Sustain. Energy Technol. Assess. 53 (2022) 102613.
[22] A. Katal, S. Dahiya, T. Choudhury, Energy efficiency in cloud computing data centers: a survey on software technologies, Clust. Comput. 26 (2023) 1845–1875.
[23] P. Bellavista, J. Berrocal, A. Corradi, S.K. Das, L. Foschini, A. Zanni, A survey on fog computing for the Internet of Things, Pervasive Mob. Comput. 52 (2019) 71–99.
[24] C. Arivazhagan, V. Natarajan, A survey on fog computing paradigms, challenges and opportunities in IoT, Int. Conf. Commun. Signal Process. (ICCSP) 2020 (2020) 0385–0389.
[25] R. Verma, S. Chandra, A systematic survey on fog steered IoT: architecture, prevalent threats and trust models, Int. J. Wirel. Inf. Netw. 28 (2021) 116–133.
[26] D. Lan, A. Taherkordi, F. Eliassen, G. Horn, A survey on fog programming: concepts, state-of-the-art, and research challenges, in: Proceedings of the Second International Workshop on Distributed Fog Services Design (2019) 1–6.
[27] U.M. Malik, M.A. Javed, S. Zeadally, S. ul Islam, Energy-efficient fog computing for 6G-enabled massive IoT: recent trends and future opportunities, IEEE Internet Things J. 9 (2021) 14572–14594.
[28] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, T. Zhou, A survey on edge computing systems and tools, Proc. IEEE 107 (2019) 1537–1562.
[29] Z. Ning, J. Huang, X. Wang, J.J. Rodrigues, L. Guo, Mobile edge computing-enabled Internet of vehicles: toward energy-efficient scheduling, IEEE Netw. 33 (2019) 198–205.
[30] Z. Zhou, J. Feng, Z. Chang, X. Shen, Energy-efficient edge computing service provisioning for vehicular networks: a consensus ADMM approach, IEEE Trans. Veh. Technol. 68 (2019) 5087–5099.
[31] M.H. Alsharif, Techno-economic evaluation of a stand-alone power system based on solar power/batteries for global system for mobile communications base stations, Energies 10 (2017) 392.
[32] C. Jiang, T. Fan, H. Gao, W. Shi, L. Liu, C. Cérin, et al., Energy aware edge computing: a survey, Comput. Commun. 151 (2020) 556–580.
[33] O. Väänänen, T. Hämäläinen, Requirements for energy efficient edge computing: a survey, in: Proceedings of the Internet of Things, Smart Spaces, and Next Generation Networks and Systems: Eighteenth International Conference, NEW2AN 2018, and 11th Conference, ruSMART 2018, St. Petersburg, Russia, August 27–29, 2018, Proceedings 18, 2018,3–15.
[34] T. Fan, Y. Qiu, C. Jiang, J. Wan, Energy aware edge computing: a survey, in: Proceedings of the High-Performance Computing Applications in Numerical Simulation and Edge Computing: ACM ICS 2018 International Workshops, HPCMS and HiDEC, Beijing, China, June 12, 2018, Revised Selected Papers 2, 2019, 79–91.
[35] Y. Sun, C. Song, S. Yu, Y. Liu, H. Pan, P. Zeng, Energy-efficient task offloading based on differential evolution in edge computing system with energy harvesting, IEEE Access 9 (2021) 16383–16391.
[36] M. Capra, R. Peloso, G. Masera, M.Ruo Roch, M. Martina, Edge computing: a survey on the hardware requirements in the internet of things world, Future Internet 11 (2019) 100.
[37] J. Li, G. Zhou, T. Tian, X. Li, A new cooling strategy for edge computing servers using compact looped heat pipe, Appl. Therm. Eng. 187 (2021) 116599.
[38] L. Kaur, R. Kaur, A survey on energy efficient routing techniques in WSNs focusing IoT applications and enhancing fog computing paradigm, Glob. Transit. Proc. 2 (2021) 520–529.
[39] A. Bozorgchenani, S. Disabato, D. Tarchi, M. Roveri, An energy harvesting solution for computation offloading in Fog Computing networks, Comput. Commun. 160 (2020) 577–587.
[40] A.A. Sadri, A.M. Rahmani, M. Saberikamarposhti, M. Hosseinzadeh, Data reduction in Fog computing and internet of things: a systematic literature survey, Internet Things (2022) 100629.
[41] F. Zhang, G. Liu, X. Fu, R. Yahyapour, "A survey on virtual machine migration: challenges, techniques, and open issues, IEEE Commun. Surv. Tutor. 20 (2018) 1206–1243.
[42] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42.
[43] T. Rault, A. Bouabdallah, Y. Challal, Energy efficiency in wireless sensor networks: a top-down survey, Comput. Netw. 67 (2014) 104–122.
[44] S.-Y. Jing, S. Ali, K. She, Y. Zhong, State-of-the-art research study for green cloud computing, J. Supercomput. 65 (2013) 445–468.
[45] K. Das, S. Das, R.K. Darji, A. Mishra, Survey of energy-efficient techniques for the cloud-integrated sensor network, J. Sens. 2018 (2018) 1–17.
[46] S.K. Mishra, S. Sahoo, B. Sahoo, S.K. Jena, Energy-efficient service allocation techniques in cloud: a survey, IETE Tech. Rev. 37 (2020) 339–352.
[47] S.S. Gill, R. Buyya, A taxonomy and future directions for sustainable cloud computing: 360 degree view, ACM Comput. Surv. (CSUR) 51 (2018) 1–33.
[48] M. Xu, A.N. Toosi, R. Buyya, A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing, IEEE Trans. Sustain. Comput. 6 (2020) 544–558.
[49] R. Medara, R.S. Singh, A review on energy-aware scheduling techniques for workflows in IaaS clouds, Wirel. Pers. Commun. 125 (2022) 1545–1584.
[50] R. Muralidhar, R. Borovica-Gajic, R. Buyya, Energy efficient computing systems: architectures, abstractions and modeling to techniques and standards, ACM Comput. Surv. (CSUR) 54 (2022) 1–37.
[51] R.K. Naha, S. Garg, D. Georgakopoulos, P.P. Jayaraman, L. Gao, Y. Xiang, et al., Fog computing: survey of trends, architectures, requirements, and research directions, IEEE Access 6 (2018) 47980–48009.
[52] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective, IEEE Commun. Surv. Tutor. 19 (2017) 2322–2358.
[53] L.A. Haibeh, M.C. Yagoub, A. Jarray, A survey on mobile edge computing infrastructure: design, resource management, and optimization approaches, IEEE Access 10 (2022) 27591–27610.
[54] X. Kong, Y. Wu, H. Wang, F. Xia, Edge computing for internet of everything: a survey, IEEE Internet Things J. 9 (2022) 23472–23485.
[55] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, et al., A survey on the edge computing for the Internet of Things, IEEE Access 6 (2017) 6900–6919.
[56] A. Hazra, P. Rana, M. Adhikari, T. Amgoth, Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges, Comput. Sci. Rev. 48 (2023) 100549.
[57] M. Songhorabadi, M. Rahimi, A. MoghadamFarid, M.H. Kashani, Fog computing approaches in IoT-enabled smart cities, J. Netw. Comput. Appl. 211 (2023) 103557.
[58] S. Bharany, S. Sharma, O.I. Khalaf, G.M. Abdulsahib, A.S. Al Humaimeedy, T. H. Aldhyani, et al., A systematic survey on energy-efficient techniques in sustainable cloud computing, Sustainability 14 (2022) 6256.
[59] D.P. Mahato, J.K. Sandhu, N.P. Singh, V. Kaushal, On scheduling transaction in grid computing using cuckoo search-ant colony optimization considering load, Clust. Comput. 23 (2020) 1483–1504.
[60] O.L. Molua, Security and Privacy Aspects of Cloud, Edge, and Fog Paradigms: a Systematic Literature Review, 2021.
[61] D. Yu, Z. Ma, R. Wang, Efficient smart grid load balancing via fog and cloud computing, Math. Probl. Eng. 2022 (2022) 1–11.
[62] M.H. Alsharif, R. Kannadasan, A. Jahid, M.A. Albreem, J. Nebhen, B.J. Choi, Long-term techno-economic analysis of sustainable and zero grid cellular base station, IEEE Access 9 (2021) 54159–54172.
[63] S. Rani, H. Babbar, P. Kaur, M.D. Alshehri, S.H.A. Shah, An optimized approach of dynamic target nodes in wireless sensor network using bio inspired algorithms for maritime rescue, IEEE Trans. Intell. Transp. Syst. (2022).
[64] N. Merabtine, D. Djenouri, D.-E. Zegour, Towards energy efficient clustering in wireless sensor networks: a comprehensive review, IEEE Access 9 (2021) 92688–92705.

[65] Z. Zhang, J. Feng, Q. Pei, L. Wang, L. Ma, Integration of communication and computing in blockchain-enabled multi-access edge computing systems, China Commun. 18 (2021) 297–314.
[66] A. Bozorgchenani, D. Tarchi, G.E. Corazza, Centralized and distributed architectures for energy and delay efficient fog network-based edge computing services, IEEE Trans. Green Commun. Netw. 3 (2018) 250–263.
[67] M.D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, A. Vakali, Cloud computing: distributed internet computing for IT and scientific research, IEEE Internet Comput. 13 (2009) 10–13.
[68] D. Haripriya, K. Kumar, A. Shrivastava, H.M.R. Al-Khafaji, V. Moyal, S.K. Singh, Energy-efficient UART design on FPGA using dynamic voltage scaling for green communication in industrial sector, Wirel. Commun. Mob. Comput. 2022 (2022).
[69] X. Cao, L. Liu, Y. Cheng, X. Shen, Towards energy-efficient wireless networking in the big data era: a survey, IEEE Commun. Surv. Tutor. 20 (2017) 303–332.
[70] Z. Sun, L. Wei, C. Xu, T. Wang, Y. Nie, X. Xing, et al., An energy-efficient cross-layer-sensing clustering method based on intelligent fog computing in WSNs, IEEE Access 7 (2019) 144165–144177.
[71] H. Abid, K.A. Darabkh, Improving energy conservation level in WSNs by modifying CH node location, in: Proceedings of the Fifth International Conference on Fog and Mobile Edge Computing (FMEC) (2020) 280–283.
[72] A. Rafi, G. Ali, J. Akram, Efficient energy utilization in fog computing based wireless sensor networks, in: Proceedings of the Second International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (2019) 1–5.
[73] T. Wang, L. Qiu, A.K. Sangaiah, G. Xu, A. Liu, Energy-efficient and trustworthy data collection protocol based on mobile fog computing in Internet of Things, IEEE Trans. Ind. Inform. 16 (2019) 3531–3539.
[74] B. Omoniwa, R. Hussain, M. Adil, A. Shakeel, A.K. Tahir, Q.U. Hasan, et al., An optimal relay scheme for outage minimization in fog-based Internet-of-Things (IoT) networks, IEEE Internet Things J. 6 (2018) 3044–3054.
[75] R.A. da Silva, N.L. da Fonseca, Location of fog nodes for reduction of energy consumption of end-user devices, IEEE Trans. Green. Commun. Netw. 4 (2020) 593–605.
[76] Y. Wu, Y. Wang, Y. Wei, S. Leng, Intelligent deployment of dedicated servers: rebalancing the computing resource in IoT, IEEE Wirel. Commun. Netw. Conf. Workshops (WCNCW) 2020 (2020) 1–6.
[77] B. Varghese, R. Buyya, Next generation cloud computing: new trends and research directions, Future Gener. Comput. Syst. 79 (2018) 849–861.
[78] M. Ahmed, A.S.M.R. Chowdhury, M. Ahme, M.M.H. Rafee, An advanced survey on cloud computing and state-of-the-art research issues, Int. J. Comput. Sci. Issues (IJCSI) 9 (2012) 201.
[79] Z.N. Rashid, S.R. Zebari, K.H. Sharif, K. Jacksi, Distributed cloud computing and distributed parallel computing: a review, Int. Conf. Adv. Sci. Eng. (ICOASE) 2018 (2018) 167–172.
[80] A.F.S. Devaraj, M. Elhoseny, S. Dhanasekaran, E.L. Lydia, K. Shankar, Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments, J. Parallel Distrib. Comput. 142 (2020) 36–45.
[81] M. Faheem, G. Tuna, V.C. Gungor, QERP: quality-of-service (QoS) aware evolutionary routing protocol for underwater wireless sensor networks, IEEE Syst. J. 12 (2017) 2066–2073.
[82] Y. Su, R. Fan, X. Fu, Z. Jin, DQELR: an adaptive deep Q-network-based energy- and latency-aware routing protocol design for underwater acoustic sensor networks, IEEE Access 7 (2019) 9091–9104.
[83] T. Hu, Y. Fei, QELAR: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks, IEEE Trans. Mob. Comput. 9 (2010) 796–809.
[84] K.Y. Islam, I. Ahmad, D. Habibi, A. Waqar, A survey on energy efficiency in underwater wireless communications, J. Netw. Comput. Appl. 198 (2022) 103295.
[85] H. Yan, Z.J. Shi, J.-H. Cui, DBR: depth-based routing for underwater sensor networks, in: Proceedings of the NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet:Seventh International IFIP-TC6 Networking Conference Singapore, May 5–9, 2008 Proceedings 7, 2008, 72–86.
[86] A. Wahid, D. Kim, An energy efficient localization-free routing protocol for underwater wireless sensor networks, Int. J. Distrib. Sens. Netw. 8 (2012) 307246.
[87] G. Dhand, S. Tyagi, Data aggregation techniques in WSN: survey, Procedia Comput. Sci. 92 (2016) 378–384.
[88] M. Azeem, A. Ullah, H. Ashraf, N. Jhanjhi, M. Humayun, S. Aljahdali, et al., Fog-oriented secure and lightweight data aggregation in iomt, IEEE Access 9 (2021) 111072–111082.
[89] M.B. Krishna, N. Vashishta, Energy efficient data aggregation techniques in wireless sensor networks, in: Proceedings of the Fifth International Conference and Computational Intelligence and Communication Networks 2013 (2013) 160–165.
[90] S. Randhawa, S. Jain, Data aggregation in wireless sensor networks: previous research, current status and future directions, Wirel. Pers. Commun. 97 (2017) 3355–3425.
[91] B. Sun, X. Shan, K. Wu, Y. Xiao, Anomaly detection based secure in-network aggregation for wireless sensor networks, IEEE Syst. J. 7 (2012) 13–25.
[92] T.N. Gia, L. Qingqing, J.P. Queralta, H. Tenhunen, Z. Zou, T. Westerlund, Lossless compression techniques in edge computing for mission-critical applications in the IoT, Twelfth Int. Conf. Mob. Comput. Ubiquitous Netw. (ICMU) 2019 (2019) 1–2.
[93] S.W. Son, Z. Chen, W. Hendrix, A. Agrawal, W.-K. Liao, A. Choudhary, Data compression for the exascale computing era-survey, Supercomput. Front. Innov. 1 (2014) 76–88.
[94] A. Jain, K.I. Lakhtaria, Comparative study of dictionary based compression algorithms on text data, Int. J. Comput. Sci. Netw. Secur. (IJCSNS) 16 (2016) 88.
[95] D. Díaz-Domínguez, S.J. Puglisi, L. Salmela, Computing All-vs-All MEMs in run-length-encoded collections of HiFi reads, in: Proceedings of the International Symposium on String Processing and Information Retrieval (2022) 198–213.
[96] J. Li, C. Gu, Y. Xiang, F. Li, Edge-cloud computing systems for smart grid: state-of-the-art, architecture, and applications, J. Mod. Power Syst. Clean. Energy 10 (2022) 805–817.
[97] K. Sivaprakasam, P. Sriramalakshmi, P. Singh, M. Bhaskar, An overview of low power hardware architecture for edge computing devices, 5G IoT Edge Comput. Smart Healthc. (2022) 89–109.
[98] L. Martin Wisniewski, J.-M. Bec, G. Boguszewski, A. Gamatié, Hardware solutions for low-power smart edge computing, J. Low. Power Electron. Appl. 12 (2022) 61.
[99] G. Nain, K. Pattanaik, G. Sharma, Towards edge computing in intelligent manufacturing: past, present and future, J. Manuf. Syst. 62 (2022) 588–611.
[100] G. Peruzzi, A. Pozzebon, Combining lorawan and nb-iot for edge-to-cloud low power connectivity leveraging on fog computing, Appl. Sci. 12 (2022) 1497.
[101] S. Iftikhar, S.S. Gill, C. Song, M. Xu, M.S. Aslanpour, A.N. Toosi, et al., AI-based fog and edge computing: a systematic review, taxonomy and future directions, Internet Things (2022) 100674.
[102] S. Tuli, F. Mirhakimi, S. Pallewatta, S. Zawad, G. Casale, B. Javadi, et al., AI augmented Edge and Fog computing: trends and challenges, J. Netw. Comput. Appl. (2023) 103648.
[103] S. Rai, B. Talawar, Nonvolatile memory technologies: characteristics, deployment, and research challenges, Front. Qual. Electron. Des. (QED) AI, IoT Hardw. Secur. (2023) 137–173.
[104] M.A. Khan, S.K. Mishra, A. Kumari, B. Sahoo, Task allocation in containerized cloud computing environment, in: Proceedings of the International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC) (2022) 1–5.
[105] Y. Zhang, K. Shan, X. Li, H. Li, S. Wang, Research and technologies for next-generation high-temperature data centers–State-of-the-arts and future perspectives, Renew. Sustain. Energy Rev. 171 (2023) 112991.
[106] J.-H. Jou, Z.-K. He, D.K. Dubey, Y.-F. Tsai, S. Sahoo, Y.-T. Su, et al., Approach for designing human-centered and energy saving lighting luminaires, Photonics (2022) 726.
[107] M. Zhang, T. Yan, W. Wang, X. Jia, J. Wang, J.J. Klemeš, Energy-saving design and control strategy towards modern sustainable greenhouse: a review, Renew. Sustain. Energy Rev. 164 (2022) 112602.
[108] B. Kocot, P. Czarnul, J. Proficz, Energy-aware scheduling for high-performance computing systems: a survey, Energies 16 (2023) 890.
[109] M. Warade, J.-G. Schneider, K. Lee, Towards energy-aware scheduling of scientific workflows, Int. Conf. Green Energy, Comput. Sustain. Technol. (GECOST) 2022 (2022) 93–98.
[110] N. Kumari, A. Yadav, P.K. Jana, Task offloading in fog computing: a survey of algorithms and optimization techniques, Comput. Netw. 214 (2022) 109137.
[111] R.M. Singh, L.K. Awasthi, G. Sikka, Towards metaheuristic scheduling techniques in cloud and fog: an extensive taxonomic review, ACM Comput. Surv. (CSUR) 55 (2022) 1–43.
[112] M.K. Alasmari, S.S. Alwakeel, Y.A. Alohali, A multi-classifiers based algorithm for energy efficient tasks offloading in fog computing, Sensors 23 (2023) 7209.
[113] M. Faraji-Mehmandar, S. Jabbehdari, H.H.S. Javadi, A self-learning approach for proactive resource and service provisioning in fog environment, J. Supercomput. 78 (2022) 16997–17026.
[114] H. Kaur, A. Anand, Review and analysis of secure energy efficient resource optimization approaches for virtual machine migration in cloud computing, Meas. Sens. (2022) 100504.
[115] S. Chen, Y. Zheng, W. Lu, V. Varadarajan, K. Wang, Energy-optimal dynamic computation offloading for industrial IoT in fog computing, IEEE Trans. Green. Commun. Netw. 4 (2019) 566–576.
[116] S. Hou, W. Ni, S. Zhao, B. Cheng, S. Chen, J. Chen, Frequency-reconfigurable cloud versus fog computing: an energy-efficiency aspect, IEEE Trans. Green Commun. Netw. 4 (2019) 221–235.
[117] A. Karimiafshar, M.R. Hashemi, M.R. Heidarpour, A.N. Toosi, Effective utilization of renewable energy sources in fog computing environment via frequency and modulation level scaling, IEEE Internet Things J. 7 (2020) 10912–10921.
[118] M. Maray, J. Shuja, Computation offloading in mobile cloud computing and mobile edge computing: survey, taxonomy, and open issues, Mob. Inf. Syst. 2022 (2022).
[119] Z. Wu, B. Li, Z. Fei, Z. Zheng, Z. Han, Energy-efficient robust computation offloading for fog-IoT systems, IEEE Trans. Veh. Technol. 69 (2020) 4417–4425.
[120] Y.-L. Jiang, Y.-S. Chen, S.-W. Yang, C.-H. Wu, Energy-efficient task offloading for time-sensitive applications in fog computing, IEEE Syst. J. 13 (2018) 2930–2941.
[121] X. Huang, W. Fan, Q. Chen, J. Zhang, Energy-efficient resource allocation in fog computing networks with the candidate mechanism, IEEE Internet Things J. 7 (2020) 8502–8512.
[122] K. Gai, X. Qin, L. Zhu, An energy-aware high performance task allocation strategy in heterogeneous fog computing environments, IEEE Trans. Comput. 70 (2020) 626–639.
[123] P. Kortoçi, L. Zheng, C. Joe-Wong, M.Di Francesco, M. Chiang, Fog-based data offloading in urban IoT scenarios. in: Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, 2019, pp. 784–792.

[124] Y. Zu, F. Shen, F. Yan, L. Shen, F. Qin, R. Yang, Smeto: stable matching for energy-minimized task offloading in cloud-fog networks, in: Proceedings of the IEEE Nintieth Vehicular Technology Conference (VTC2019-Fall) (2019) 1–5.
[125] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, M.-T. Zhou, FEMTO: fair and energy-minimized task offloading for fog-enabled IoT networks, IEEE Internet Things J. 6 (2018) 4388–4400.
[126] M. Al-Khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, Y. Jararweh, Improving fog computing performance via fog-2-fog collaboration, Future Gener. Comput. Syst. 100 (2019) 266–280.
[127] J. Kim, T. Ha, W. Yoo, J.-M. Chung, Task popularity-based energy minimized computation offloading for fog computing wireless networks, IEEE Wirel. Commun. Lett. 8 (2019) 1200–1203.
[128] R. Borawake-Satao, R. Prasad, Green Internet of Things schemes and techniques for adaptive energy saving in emergency services, Internet Things Smart Comput. Technol. A Roadmap Ahead (2020) 173–188.
[129] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, R.S. Tucker, Fog computing may help to save energy in cloud computing, IEEE J. Sel. Areas Commun. 34 (2016) 1728–1739.
[130] C. Perera, Y. Qin, J.C. Estrella, S. Reiff-Marganiec, A.V. Vasilakos, Fog computing for sustainable smart cities: a survey, ACM Comput. Surv. (CSUR) 50 (2017) 1–43.
[131] A. N. Toosi, C. Agarwal, L. Mashayekhy, S.K. Moghaddam, R. Mahmud, Z. Tari, GreenFog: a framework for sustainable fog computing, Int. Conf. Serv. -Oriented Comput. (2022) 540–549.
[132] J. Haimour, O. Abu-Sharkh, Energy efficient sleep/wake-up techniques for IOT: a survey, IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol. (JEEIT) 2019 (2019) 478–484.
[133] R. Venanzi, L. Foschini, P. Bellavista, B. Kantarci, C. Stefanelli, Fog-driven context-aware architecture for node discovery and energy saving strategy for Internet of Things environments, IEEE Access 7 (2019) 134173–134186.
[134] I. Natgunanathan, N. Fernando, S.W. Loke, C. Weerasuriya, Bluetooth low energy mesh: applications, considerations and current state-of-the-art, Sensors 23 (2023) 1826.
[135] D.-M. Bui, Y. Yoon, E.-N. Huh, S. Jun, S. Lee, Energy efficiency for cloud computing system based on predictive optimization, J. Parallel Distrib. Comput. 102 (2017) 103–114.
[136] M. Hijji, B. Ahmad, G. Alam, A. Alwakeel, M. Alwakeel, L. Abdulaziz Alharbi, et al., Cloud servers: resource optimization using different energy saving techniques, Sensors 22 (2022) 8384.
[137] W. Yao, Z. Wang, Y. Hou, X. Zhu, X. Li, Y. Xia, An energy-efficient load balance strategy based on virtual machine consolidation in cloud environment, Future Gener. Comput. Syst. 146 (2023) 222–233.
[138] J. Kavitha, P.S. Rao, G.C. Babu, Energy efficient resource utilization of cloud computing environments for deployment models, in: Proceedings of the Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) 2023 (2023) 1111–1119.
[139] N. Sadashiv, Load balancing in fog computing: a detailed survey, Int. J. Comput. Digit. Syst. 13 (2023) 729–750.
[140] L.U. Khan, I. Yaqoob, N.H. Tran, S.A. Kazmi, T.N. Dang, C.S. Hong, Edge-computing-enabled smart cities: a comprehensive survey, IEEE Internet Things J. 7 (2020) 10200–10232.
[141] C. Avasalcai, I. Murturi, S. Dustdar, Edge and fog: a survey, use cases, and future challenges, Fog Comput. Theory Pract. (2020) 43–65.
[142] M. Mukherjee, L. Shu, D. Wang, Survey of fog computing: fundamental, network applications, and research challenges, IEEE Commun. Surv. Tutor. 20 (2018) 1826–1857.
[143] A. Ullah, N.M. Nawi, S. Ouhame, Recent advancement in VM task allocation system for cloud computing: review from 2015 to2021, Artif. Intell. Rev. (2022) 1–45.
[144] M. Guo, Q. Li, Z. Peng, X. Liu, D. Cui, Energy harvesting computation offloading game towards minimizing delay for mobile edge computing, Comput. Netw. 204 (2022) 108678.
[145] Y. Chen, F. Zhao, Y. Lu, X. Chen, Dynamic task offloading for mobile edge computing with hybrid energy supply, Tsinghua Sci. Technol. 28 (2022) 421–432.
[146] H. Herath, R. Prematilake, B. Madhusanka, Integration of Fog Computing and IoT-Based Energy Harvesting (EHIoT) Model for Wireless Sensor Network, Energy Conserv. Solut. Fog-Edge Comput. Paradig. (2022) 215–231.
[147] M.H. Alsharif, R. Nordin, M. Ismail, Green wireless network optimisation strategies within smart grid environments for Long Term Evolution (LTE) cellular networks in Malaysia, Renew. Energy 85 (2016) 157–170.
[148] L.R. Jahangard, A. Shirmarz, Taxonomy of green cloud computing techniques with environment quality improvement considering: a survey, Int. J. Energy Environ. Eng. 13 (2022) 1247–1269.
[149] K. Zhang, Y. Zhang, J. Liu, X. Niu, Recent advancements on thermal management and evaluation for data centers, Appl. Therm. Eng. 142 (2018) 215–231.
[150] B.R. Park, Y.J. Choi, E.J. Choi, J.W. Moon, Adaptive control algorithm with a retraining technique to predict the optimal amount of chilled water in a data center cooling system, J. Build. Eng. 50 (2022) 104167.
[151] S. Iftikhar, M.M.M. Ahmad, S. Tuli, D. Chowdhury, M. Xu, S.S. Gill, et al., HunterPlus: AI based energy-efficient task scheduling for cloud–fog computing environments, Internet Things 21 (2023) 100667.
[152] H. Zhang, S. Shao, H. Xu, H. Zou, C. Tian, Free cooling of data centers: a review, Renew. Sustain. Energy Rev. 35 (2014) 171–182.
[153] L. Cupertino, G. Da Costa, A. Oleksiak, W. Pia, J.-M. Pierson, J. Salom, et al., Energy-efficient, thermal-aware modeling and simulation of data centers: the CoolEmAll approach and evaluation results, Ad Hoc Netw. 25 (2015) 535–553.