Cluster Computing (2025)28:18
https://doi.org/10.1007/s10586-024-04686-y

Edge AI: A Taxonomy, Systematic Review and Future Directions

Sukhpal Singh Gill[1] · Muhammed Golec[1,2] · Jianmin Hu[3] · Minxian Xu[3] · Junhui Du[4] · Huaming Wu[4] · Guneet Kaur Walia[5] · Subramaniam Subramanian Murugesan[1] · Babar Ali[1] · Mohit Kumar[5] · Kejiang Ye[3] · Prabal Verma[8] · Surendra Kumar[6] · Felix Cuadrado[7] · Steve Uhlig[1]

Received: 8 July 2024 / Revised: 14 September 2024 / Accepted: 30 September 2024

Abstract
Edge Artificial Intelligence (AI) incorporates a network of interconnected systems and devices that receive, cache, process, and analyse data in close communication with the location where the data is captured with AI technology. Recent advancements in AI efficiency, the widespread use of Internet of Things (IoT) devices, and the emergence of edge computing have unlocked the enormous scope of Edge AI. The goal of Edge AI is to optimize data processing efficiency and velocity while ensuring data confidentiality and integrity. Despite being a relatively new field of research, spanning from 2014 to the present, it has shown significant and rapid development over the last five years. In this article, we present a systematic literature review for Edge AI to discuss the existing research, recent advancements, and future research directions. We created a collaborative edge AI learning system for cloud and edge computing analysis, including an in-depth study of the architectures that facilitate this mechanism. The taxonomy for Edge AI facilitates the classification and configuration of Edge AI systems while also examining its potential influence across many fields through compassing infrastructure, cloud computing, fog computing, services, use cases, ML and deep learning, and resource management. This study highlights the significance of Edge AI in processing real-time data at the edge of the network. Additionally, it emphasizes the research challenges encountered by Edge AI systems, including constraints on resources, vulnerabilities to security threats, and problems with scalability. Finally, this study highlights the potential future research directions that aim to address the current limitations of Edge AI by providing innovative solutions.

Keywords Edge computing · Artificial intelligence · Cloud computing · Machine learning · Edge AI

1 Introduction

Recent advancements in Artificial Intelligence (AI), the growing adoption of Internet of Things (IoT) devices, and the rise of edge computing are converging to unleash the full potential of edge AI [1]. Numerous analysts and businesses are conversing about and executing edge computing, which delineates its origins to the 1990 s when edge servers positioned near customers were used to serve web and video content over content delivery networks [2]. Edge computing is a paradigm transformation in this edge AI that brings data storage and processing closer to the data source, improving response times and reducing bandwidth usage. Unlike traditional cloud computing, where centralized data centers process data, edge computing processes data at the network's edge [3]. This proximity reduces latency, enhances real-time data processing capabilities, and supports the expansion of IoT devices and services [4]. The primary advantages of edge computing include improved agility of services, low latency, enhanced coherence, and the elimination of a single point of failure, making it highly relevant for applications in smart cities, self-sufficient vehicles, and industrial automation [5]. By distributing resources geographically, edge computing ensures that data processing occurs near the data source, satisfying the need for analytics and decision-making in real-time.

On the other hand, AI includes a wide array of technologies and methodologies that enable machines to carry out tasks that generally require human intelligence, such as learning, reasoning, and self-correction [6]. AI's

Sukhpal Singh Gill, Muhammed Golec, Jianmin Hu, Minxian Xu, Junhui Du, Huaming Wu, Guneet Kaur Walia, Subramaniam Subramanian Murugesan, Babar Ali, Mohit Kumar, Kejiang Ye, Prabal Verma, Surendra Kumar, Felix Cuadrado, Steve Uhlig contributed equally.

Extended author information available on the last page of the article

Published online: 18 October 2024

applications span various domains, including healthcare, finance, transportation, etc, where it is used to analyze large datasets, automate tasks, and provide predictive insights [7]. Integrating AI into different sectors has revolutionized processes by enhancing efficiency, improving decision-making, and creating new opportunities for innovation. With betterment in Machine Learning (ML) or Deep Learning (DL), AI approaches have become increasingly competent in performing complex tasks that require human-like cognitive functions [8]. AI algorithms, specifically those involving neural networks, have shown remarkable success in areas like image and speech recognition, autonomous driving, and predictive maintenance.

1.1 Edge AI

The fusion of edge computing and AI involves processing AI algorithms on users' devices, offering benefits like reduced latency, energy efficiency, and real-time applications. This integration allows for real-time data processing and decision-making at the source, significantly declining latency and bandwidth use [9]. The combination of edge computing and AI enables the development of smarter and more responsive applications, such as autonomous vehicles, industrial IoT, smart home systems, etc. By leveraging edge AI, organizations can achieve greater efficiency, enhanced privacy, and faster insights, driving innovation across various sectors [10]. Edge AI refers to integrating AI capabilities at the network edge, enabling distributed intelligence with edge devices. It intends to improve network connectivity, enable deployment of AI pipelines with defined quality targets, and allow adaption for data-driven applications. [11]. Embedding AI functionalities at the edge addresses the limitations of cloud-based processing for IoT, such as privacy concerns and network connectivity issues. The deployment of AI at the edge enhances latency-sensitive tasks and reduces network congestion, improving efficiency and security in wireless networks.

Furthermore, AI-based technologies play a vital role in addressing Quality of Service (QoS)-aware scheduling and resource allocation challenges in edge environments, ensuring QoS and user experience. Edge AI enables the deployment of AI as a Service (AIaaS) with configurable model complexity and data quality, enhancing performance and reducing costs [12, 13]. This innovative approach supports smart security applications by leveraging AI capabilities at the edge and enhancing security measures for distributed systems. Edge intelligence, a promising technology, empowers real-time applications by moving computing from cloud servers to IoT edge devices, creating intelligent enterprises with vast possibilities [14]. The utilization of AI at the edge, instead of centralized locations, unlocks the potential of AI with IoT devices and edge computing, deploying AI algorithms on resource-constrained edge devices for various applications like autonomous vehicles, healthcare, and surveillance.

Edge AI's significance is underscored by its ability to provide immediate insights and actions without sending significant amounts of data to several centralized locations [15]. This capability is particularly critical in scenarios where latency and bandwidth are significant constraints, such as in autonomous driving, where decisions must be made in real time, or in healthcare, where patient data must be processed quickly to provide timely interventions [16]. The rise of edge AI is also fueled by advancements in hardware, such as more powerful and energy-efficient processors, which make it feasible to run sophisticated AI models on devices like smartphones and IoT sensors [17].

1.2 Need of Edge AI

The motivation for integrating edge computing with AI is multifaceted, primarily driven by the imperative need for processing data in real time and navigating the inherent limitations of centralized cloud computing systems [18]. As we witness an exponential rise in the number of connected devices and a corresponding surge in data volume, traditional cloud-centric models increasingly grapple with issues such as latency, bandwidth constraints, and significant data privacy concerns. Edge AI emerges as a pivotal solution to these challenges, advocating for localized data processing [19]. This shift not only diminishes the reliance on distant cloud infrastructures, thereby slashing latency, but also significantly bolsters the responsiveness of applications to real-time data inputs. This paradigm shift is particularly pivotal for fueling the development of next-gen technologies that necessitate instantaneous data analysis and decision-making, encompassing sectors like autonomous vehicles, smart city infrastructures, and cutting-edge healthcare systems.

Moreover, Edge AI empowers applications to operate remarkably efficiently, even in scenarios characterized by sparse connectivity, by facilitating data processing directly at the source. This capability is indispensable in remote or highly mobile environments where consistent and reliable internet access is only sometimes assured [20]. By processing data onsite, edge AI considerably amplifies data privacy and security measures, mitigating the need to transmit sensitive information over vast distances to central servers. This feature is exceptionally critical in domains such as healthcare and finance, where the confidentiality and integrity of data are of utmost concern.

Additionally, Edge AI champions bandwidth efficiency by mitigating the volume of data that needs to be transmitted over networks, making it an economical choice for data-intensive applications [21]. This efficiency not only

reduces operational costs but also relieves network congestion, facilitating smoother and more reliable data flows. Scalability is another significant advantage offered by edge AI [6]. As the network of devices expands, edge computing allows for seamless scalability without the bottleneck of centralized processing power, enabling businesses and technologies to grow without being hampered by infrastructure limitations.

Essentially, the combined use of edge computing and AI is not just a technical progression but also a tactical imperative to fulfill the dynamic requirements of contemporary applications. By championing lowered latency, enhanced privacy and security, bandwidth efficiency, and scalability, edge AI is set to revolutionize how data-driven decisions are made, ushering in a new era of intelligence that is both efficient and privacy-centric.

1.3 Article organization

Section 2 offer present situation of Edge AI. Section 4 details the methodology adopted for the review. Section 3 discusses a related surveys and studies focusing on different applications in terms of algorithms, optimization techniques, security, and privacy concerns integrated with Edge AI. Section 5 outlines a taxonomy encompassing infrastructure, cloud computing, fog computing, services, use cases, machine learning and deep learning, and resource management. Section 6 compares existing Edge AI implementations based on taxonomy. Section 7 presents an analysis and the results obtained, and the future research directions are discussed in Sect. 8. Finally, Sect. 9 summarizes the survey.

2 Edge AI: background and current status

This section explains some concepts related to background and current status in Edge AI. Subsection 2.1 explains edge computing and its historical emergence. Subsection 2.2 provides information on the integration of AI and edge technologies. This section is completed by explaining Edge AI applications and challenges in subsection 2.3 and subsection 2.4, respectively.

2.1 Historical emergence of Edge computing

The concept of edge computing is a paradigm that brings computing resources closer to the data source, unlike the cloud, which provides services through a remote server [22]. In this way, it is aimed to reduce problems such as unnecessary bandwidth occupation and latency in today's world where huge amounts of data that need to be processed are produced [23]. To understand the emergence of edge computing, it will be more useful to examine previous paradigms such as cloud and fog computing. Figure 1 shows the advantages of cloud, fog, and edge computing over each other and their layer arrangement. These concepts are discussed briefly below:

- **Cloud Computing:** It is a paradigm that dates back to the 1970 s and refers to the use of common computing resources by users on a server via the Internet [24]. Today, it is offered to users with various service models, especially by large companies such as Microsoft Azure, Google Cloud Platform and IBM Cloud. The advantages of cloud computing are as follows [25]:
  - High processing power and central storage, so users can easily access resources from anywhere there is the Internet. This reduces the user's risk of data loss and provides the user with the freedom to work from any location with Internet access.
  - Scalability, in case the need for computing resources increases (demand fluctuations), cloud computing provides services such as more processing power and storage by scaling the resources. In this way, performance measures such as service-level agreement (SLA) and QoS are ensured.
  - Pay as you go, with the serverless (Function as a Service (FaaS) + Backend as a service (BaaS)) service model provided by cloud computing, users are charged only for the amount they use their computing resources. In this way, an economical model is provided and appealed to more users.
- **Fog Computing:** The concept of fog computing was introduced by Cisco in 2012 [26]. This paradigm recommends moving computing resources closer to the endpoints of the network (such as routers and gateways) to reduce the latency and bandwidth problems that occur in cloud computing. When Fig. 1 is examined, fog computing acts as a layer between the cloud and the edge. The advantages of fog computing are as follows [27]:
  - Fog computing has lower latency than cloud because it brings computing resources closer to the edge of the network.
  - By acting as a layer between the cloud and end devices, it reduces unnecessary bandwidth usage by processing some of the huge amounts of data to be sent to the cloud.
- **Edge Computing** The development of IoT and sensor technologies has increased the amount of data that needs to be processed to enormous levels. Processing all this data on cloud computing resources may cause unnecessary bandwidth occupation and latency

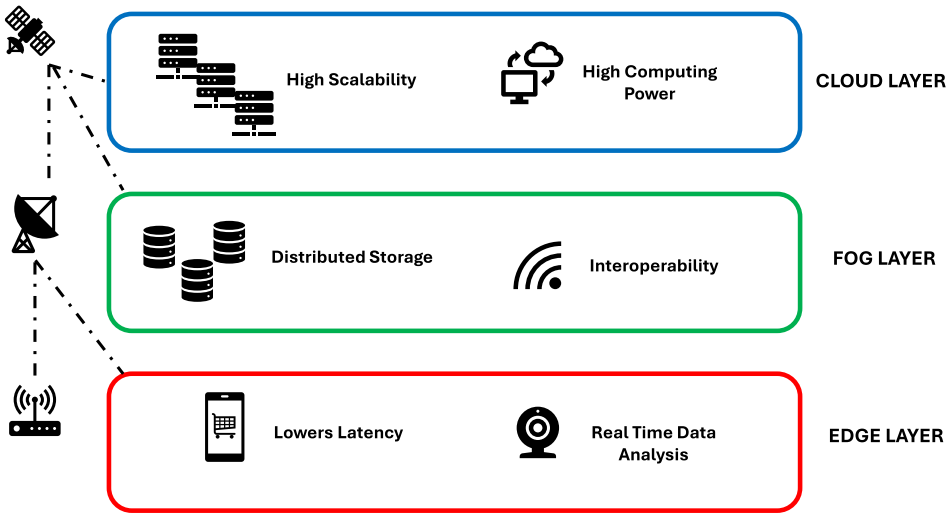


Fig. 1 Computing Paradigms and their Objectives

problems. For this reason, the concept of edge computing has emerged as a paradigm that aims to optimize latency and bandwidth usage by processing data close to the data source [28]. Additionally, edge computing is a good solution to address the complexity, security, and management challenges posed by fog computing, an extra layer [29]. The advantages of edge computing are as follows [30]:

- Reduces latency and bandwidth usage by moving data processing to the edge of the network,
- Compared to fog computing, it offers advantages such as less complexity and better security.

2.2 Integration of AI with Edge technology

Developing AI applications have begun to show themselves in many areas. One of these areas is EdgeAI, which is the combination of AI and Edge concepts [29]. EdgeAI is based on the principle of processing data on edge nodes such as mobile devices and IoT instead of processing it on cloud servers [31]. This is achieved by distributing AI algorithms to edge nodes close to the data source, as shown in Fig. 2, which shows how data is processed in the EdgeAI concept and how it performs fast and efficient computation. The advantages offered by these two technologies can be listed as follows [6]:

- Low Latency: In delay-sensitive scenarios such as e-health and autonomous vehicle applications where patients are monitored instantly, millisecond delays are critical [32]. In traditional cloud-based systems, data must be sent between the user and the cloud to be processed in the AI model deployed in the cloud. This process will cause serious delay and unnecessary bandwidth usage [23]. With Edge and AI integration, this problem can be overcome by processing data in real time. Because the data will be processed on the edge node closest to the source where it was produced, it will respond much faster than cloud-based systems.
- Increased Security and Privacy: In cloud systems, data is sent from the source where it is produced to central servers. This expands the attack surface for hackers in the communication channels and storage areas of sensitive data such as biometric and health data [32]. In EdgeAI systems, since the data is processed and stored locally compared to cloud systems, it can be said that the overall security of the system is higher. Similarly, privacy issues that may occur in the event of theft of sensitive data such as biometric data are reduced [22].
- Resource Optimization and Scalability: EdgeAI systems consist of heterogeneous devices such as laptops, network routers, mobile devices with different processing power and storage capabilities. This means that EdgeAI can share the resources of devices in the edge network if external processing power and resources are needed. In addition, balanced load distribution can be achieved by using advanced resource allocation algorithms to optimize resources.

Future Directions and Limitations: Despite the above-mentioned advantages and high potential, EdgeAI also brings with it challenges such as (i) limited processing power of devices in the edge network, (ii) management difficulties due to the heterogeneous structure of the edge network, and (iii) energy constraints due to resource limitations. If future researchers solve these challenges, it is expected that the areas of use of EdgeAI will expand.

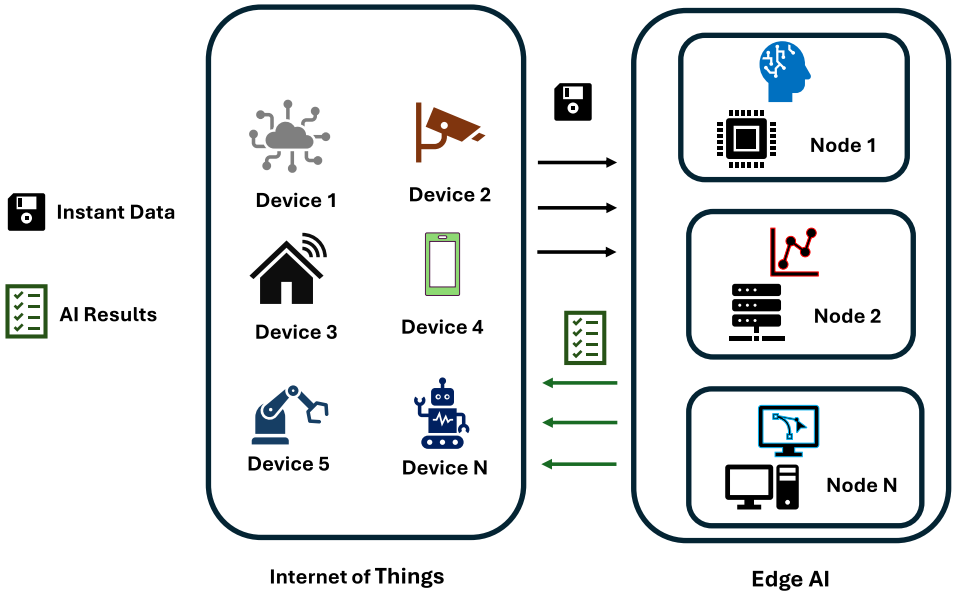


Fig. 2 Architecture of Edge AI

2.3 Edge AI applications

Edge AI applications, created by combining the concepts of Edge and AI, provide lower latency and higher security than Cloud-based AI applications. Figure 3 shows popular applications of Edge AI, which are discussed briefly below:

- Healthcare: Edge AI applications are based on the processing of data collected from wearable devices in distributed AI models at the edge of the network. Additionally, early diagnosis studies using portable medical imaging techniques can be given as examples [25].
- Smart Parking: With the increase in means of transportation, parking has become a big problem, especially in big cities. Edge AI-based solutions with the help of sensors and IoT devices can be used to solve these problems [33].
- Smart Home: Solutions used in modern homes such as home lighting systems and smart refrigerators can be given as examples of these applications. In this way, energy consumption can be optimized by preventing excess electricity consumption in cities [27].
- Computer Vision: Edge AI can identify people using methods such as biometric authentication [22]. Additionally, Edge AI provides great advantages in Industry applications that require real-time decisions [29].

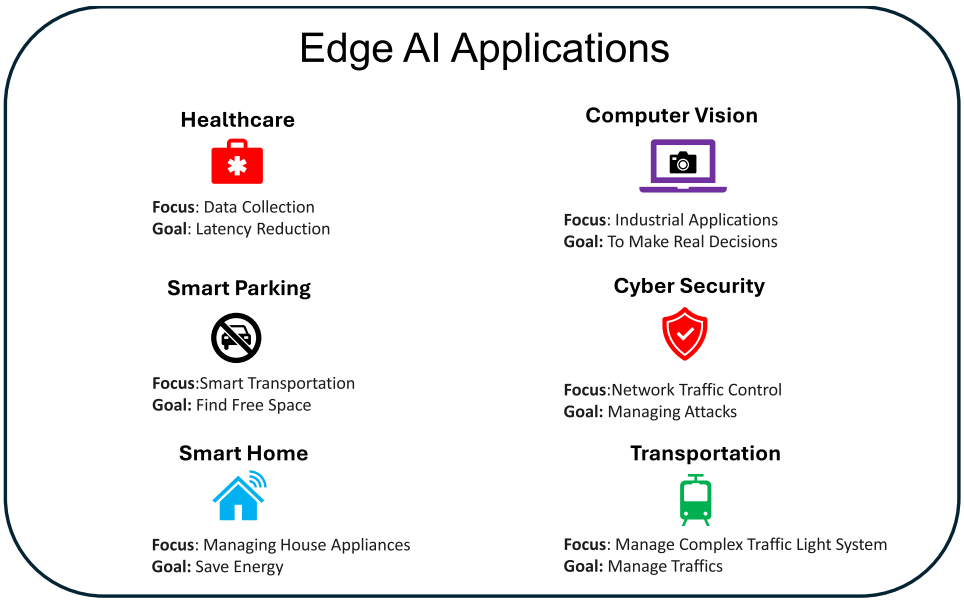


Fig. 3 The Edge AI Applications

- Cyber Security: Unauthorized access, suspicious objects, and armed individuals can be detected with Edge AI-based security applications. Additionally, anomaly detection can be made by detecting suspicious traffic on a network to prevent cyber attacks [34].
- Transportation: Edge AI-based solutions can be used for today's complex traffic light operations [35].

2.4 Edge AI implementation challenges

EdgeAI, which emerges by combining Edge and AI, brings with it the advantages it offers, but also challenges that are still waiting to be solved. These challenges are shown in Fig. 4. These challenges are discussed briefly below:

- Energy Efficiency: Edge devices generally consist of homogeneous and heterogeneous devices with low processing and storage capacity. Applications that require Natural Language Processing (NLP) and intensive image processing will cause excessive resource consumption on edge devices [36]. For this reason, new solutions such as special AI chips or task engineering are needed. Therefore, new solutions have emerged, such as dedicated AI chips or task engineering. Examples include Google's TPU and NVIDIA's JetsonAI chips, which use low-power algorithms to achieve energy efficiency [37]. Another energy-efficient method is quantization and pruning techniques. These techniques reduce energy consumption by reducing the size of models in neural networks.
- Maintenance and Updates: Since edge devices consist of devices distributed in different locations, this means more attack targets for hackers [38]. In addition, not all devices in the edge nodes have a homogeneous structure, which means separate system maintenance and updates for each node [39]. Measures such as automatic updating can be taken to solve these problems. In automatic update solutions, system incompatibilities may occur due to the heterogeneous structure of the devices. For this reason, containerization and orchestration solutions come to the fore in terms of service isolation and ease of management [40].

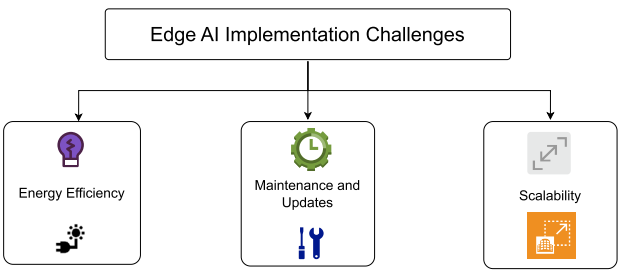


Fig. 4 Edge AI Implementation Challenges

- Scalability: Since edge devices generally consist of heterogeneous devices, the distribution of a single application to different devices is still a challenge (task scheduling, etc.) [41]. Additionally, it is difficult to synchronize data across all devices. Effective microservice architectures and load-balancing algorithms that prevent a node from being overloaded can be used to solve this problem. Examples include Kubernetes and Istio tools [42]. These tools can easily overcome load balancing and automating service discovery issues, while performance optimization should be considered for scenarios that require low latency.

3 Related studies and surveys

In this section, we discuss related studies and surveys, as well as our main contributions.

3.1 Related studies

Here, we discuss various studies, which are about the different applications consisting of smart cities, smart manufacturing, autonomous vehicles, the Internet of Vehicles (IoV), industrial automation, and healthcare monitoring systems. These are highlighted when edge computing meets AI and AI for edge computing. There are also considerations of traditional ML, computation offloading optimization, and concerns related to privacy and security, reflecting a comprehensive analysis of the challenges and strategies integrating AI and edge computing.

3.1.1 Smart cities

In the case of innovative city applications, the intersection of AI and edge computing in smart cities emphasizes the importance of optimizing computation offloading and fostering a Federation between edge, cloud, and fog computing for efficient operations, which have been discussed in the following articles. In 2020, authors [43] presented an intelligent offloading method (IOM) that preserves privacy, boosts edge utility, and improves offloading efficiency for smart cities. The mechanism of information entropy is utilized in conjunction with edge computing to achieve an equilibrium between the maintenance of privacy and the facilitation of collaborative services. Further, authors [44] employed a cooperative compute offloading method to obtain the aforementioned trade-off in the cooperation of three ends: IoT device, cloudlet, and cloud. Offloading, on the other hand, can significantly reduce the processing strain on IoT devices; yet, it may incur high transmission costs and cloudlet resource use. Furthermore, authors [45]

outlined the introduction of a cyclic branch network, a DL-based intelligent offloading scheme that makes full use of network edge computing power and data traffic to reduce overall energy consumption in dual connectivity and nonorthogonal multiple access computation offloading systems. Moreover, authors [46] suggested Robust Neural Networks From Coded Classification (CoDNN), a unique compute offloading method for multi-device collaborative pipelining processing of deep neural network (DNN) tasks. In [47], authors focused on an approximation technique called Accuracy Maximization using LP-Relaxation and Rounding (AMR2), which is suggested and shown to produce a makespan of no more than 2T and a total accuracy that is less than the optimal total accuracy by a small constant. Another work [48] presents a revolutionary deep neural network-based energy-efficient offloading strategy that trains a smart decision-making model to select a reliable pool of application components. Finally, researchers [49] suggested method for the heterogeneous scenario is consistently effective in identifying a superior offloading scheme than the chosen existing algorithms, according to empirical findings. On the other hand, for the homogeneous scenario, the suggested solution can effectively accomplish the ideal approach.

3.1.2 Smart manufacturing

In intelligent manufacturing, combining AI with Edge Computing significantly enhances efficiency, decision-making, and security, effectively addresses challenges, and improves data utilization. This integration streamlines operations, enables predictive maintenance, and supports autonomous decisions. The following papers are discussed to describe these advancements and subsequent challenges: In 2022, authors [50] explored the scope of AI and its application in India's intelligent manufacturing industry, concentrating on the technology's current state, constraints, and recommendations for resolving issues. In [51], authors have discussed how ML and AI may boost productivity, sustainability, and manufacturing efficiency. However, there are several difficulties with implementing AI in manufacturing, including problems with infrastructure and human resources, security threats, trust, and data management and acquisition. Further, the authors [52] suggested a new mode called "AI-Mfg-Ops" (AI-enabled Manufacturing Operations) with a supporting software-defined framework proposed as part of an open evolutionary architecture of the intelligent cloud manufacturing system. This mode can facilitate quick operation and upgrades of cloud manufacturing systems with intelligent assessment, analysis, planning, and execution in a closed loop. In 2021, authors [53] addressed the job shop scheduling challenge in the intelligent factory process while using a Deep Q-network (DQN). The suggested framework is contrasted and examined with other frameworks from the standpoint of offering an intelligent factory service. Furthermore, the authors [54] look at how it tends to integrate several productivity factors, such as big data analytics, Automation, and Operations Information, which connect machines via open platforms, resulting in real-time reactions to boost efficiency across the supply chain. Moreover, the authors [55] developed a service-oriented information model to standardized describe the functional characteristics and related operational data of heterogeneous manufacturing resources; additionally, a message middleware-based real-time transmission and integration method for high-volume operational field and sensor data is suggested to achieve the efficient distribution of related data and remote monitoring of distributed manufacturing resources. Finally, the authors [56] discuss the possible advantages and difficulties of a federated learning architecture based on data gathered from 5 G MSPs to enable predictive maintenance (PM) in industrial settings.

3.1.3 Autonomous vehicles and the internet of vehicles (IoV)

In the context of autonomous and IoT-enabled vehicles, advancements in control and task optimization are propelled by AI integration and Edge Computing (EC). This synergy supports real-time decision-making, highlighting the importance of AI and EC in addressing challenges like real-time processing and security and optimizing communication and privacy within the IoV. The application of DL and Reinforcement Learning with Multiple Agents (MARL) Underscores the need for efficient solutions in autonomous driving technologies. In these scenarios, the following papers are described: In 2023, authors [57] offer a thorough technical overview of the most recent studies conducted in the areas of lateral, longitudinal, and integrated control strategies for self-driving cars. They also examine a variety of strategies and tactics used to attain accurate steering control while taking longitudinal factors into account. [58] discusses key technologies, applications, solutions, and problems related to integrating Mobile Edge Computing (MEC) and ML in the Internet of UAVs and are covered in-depth by the author's thorough review. Further, authors [59] examined the most recent research on vehicular data offloading from the standpoint of communication, focusing on vehicle-to-vehicle (V2V), vehicle-to-roadside infrastructure (V2I), and vehicle-to-everything (V2X). The study also identified unresolved research issues in this area and forecasted future directions in the field. Furthermore, the authors [60] suggested a multi-access edge computing (MEC) framework to facilitate the cooperation of digital twins (DTs) into wireless networks and connected cars

(CVs) to reduce the unreliability of long-distance communication between edge servers and CVs. Moreover, researchers [61] outline a DL and edge computing-based vehicle intelligent control system that encourages the broad advancement of automation and intelligent technology. The findings show that the distance between the target and experimental vehicles is extremely close to the anticipated safe space. In [62], authors suggested a safe edge intelligence that combined the advantages of blockchain, local differential privacy (LDP), and federated learning (FL) for automotive networks. The authors in [63] focus on a fast task execution technique in heterogeneous IoT applications that are powered by AI. This technique reduces decision latency by considering various system parameters, including the task's execution deadline, the device's battery level, the channel conditions between mobile devices and edge servers, and the capacity of the edge servers. Finally, the authors [64] combined edge computing and the Web of Things, compared their functions, and showed how edge computing improves the efficiency of real-time IoT applications by focusing on transmission, storage, and computation elements.

3.1.4 Industrial automation

In industrial automation, several papers discuss revolutionary approaches to enhancing productivity by integrating AI, edge computing, robotics, and data analytics. The relevant papers look over the utilization of the Industrial Cyber Intelligent Control Operating System (ICICICOS), a cloud-edge computing-based system, for AI and industrial automation. It focuses on proposing AI with industrial processes at the edge. It emphasizes strategies for optimizing ML methods, deploying AI models on resource-constrained devices, and addressing security concerns through secure AI microservices at the edge. The relevant papers are described as follows: In 2022, the authors [11] presented a flexible working mechanism by permitting the combined design of data quality ratios (DQRs) and model complexity ratios (MCRs) for the AI tasks and suggested a configurable model deployment architecture for edge AIaaS. Furthermore, the authors [12] provide a systematic overview of methods for addressing the paucity of training data for different kinds of data, and a methodology for addressing data scarcity in cellular networks is suggested. In [14], the authors explore the privacy-enhancing solutions that are now in position, including the technologies, specifications, and process solutions to mitigate these risks. It also looks at privacy threats at various stages of the AI life cycle. Further, the authors [16] offer a thorough overview of edge intelligence and lightweight ML support for upcoming services and applications. The researchers have supplied a thorough analysis of cutting-edge intelligence applications, lightweight ML techniques, and their support for upcoming services and applications. In [65], authors provide a thorough review of AI/ML-based IDS/MDSs and set baseline measurements pertinent to networked autonomous systems, emphasizing the gaps and assessment metrics in the existing research. In [6], the authors wrapped up a thorough analysis of edge computing, covering both the shift to edge AI and related paradigms. Additionally, the history of every alternative put out for edge computing implementation, as well as the Edge AI strategy for putting AI models and algorithms on edge devices, were investigated.

3.1.5 Smart healthcare

Intelligent healthcare systems focus on integrating AI and edge computing and the challenges related to privacy and security, decision-making, and optimization. These systems make use of technologies, including genetic-based encryption for data security, federated learning in the Internet of Medical Things, and nanosensor-equipped systems to improve efficiency and security. Mobile computing has played a vital role in healthcare, particularly during the COVID-19 pandemic, enabling telemedicine and contact tracing, emphasizing the significance of technology in tackling healthcare challenges. The following research papers delve deeper into the intersection of technology, healthcare, and privacy. In 2018, the authors [66] proposed an intelligent home monitoring system based on edge-fog computing with AI capabilities. Latency issues and reliability are the main concerns for the authors in developing smart home real-time applications. Further, the authors [67] provide a comprehensive overview of the key elements of the MCPS from multiple perspectives, covering design, methodology, and significant supporting technologies such as cloud computing, edge computing, IoT, sensor networks, and systems with multiple agents. In [68], authors provided a distinctive and specialized route resource recommendation (R3) protocol to handle resource management and connection problems in autonomous, connected ambulances (ACA) for route optimization. Furthermore, the authors [69] provide a condition-aware analytical framework that may be used to recommend health conditions in IoT-based mobile healthcare systems. This framework corresponds to IoT devices that have limited resources, such as those with a memory utilization rate of 6.6%. Moreover, the authors [11] Present a flexible working mechanism that allows the combined configuration of data quality ratios (DQRs) and model complexity ratios (MCRs) for AI tasks and addresses a flexible model deployment architecture for edge AIaaS. In [70], researchers suggest using an edge-of-things (EoT) framework to implement centralized and federated transfer

Table 1 Comparison of our survey with existing related surveys

Work	Research domain	[2]	[4]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[14]	[16]	Proposed work
Year		2022	2023	2023	2020	2022	2024	2020	2023	2024	2023	2023	2024
Edge AI		✓(*)		✓(*)	✓(*)				✓(*)				✓
Taxonomy		✓(*)				✓(*)							✓
	Cloud				✓		✓	✓			✓		✓
Infrastructure	Fog		✓										✓
	Edge	✓(*)	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Application	Monolithic	✓(*)	✓										✓
Architecture	Microservice		✓(*)										✓
IoT	Static	✓(*)	✓		✓	✓	✓						✓
Use Cases	Mobile		✓	✓	✓	✓	✓					✓(*)	✓
	Heuristic			✓									
	Meta-Heuristic		✓										✓
Methods	Machine	✓	✓	✓	✓	✓	✓	✓	✓(*)			✓(*)	✓
	Learning												
	Deep	✓						✓	✓(*)	✓			✓
	Reinforcement												
	Learning												
	Provisioning					✓		✓	✓(*)	✓			✓
Resource	Resource Allocation			✓		✓(*)							✓
Management	Application Placement	✓(*)									✓(*)	✓	✓
	Workload Distribution and Prediction			✓				✓				✓	✓
ML Model Sizing	Reduced			✓						✓		✓	✓
	Full	✓			✓			✓					✓
					✓								
	Computational	✓(*)	✓(*)	✓(*)								✓	✓
Heterogeneity	Hardware												✓
	Platform												✓
	Platform												✓
Security	Host												✓
	Network			✓							✓(*)		✓
	Container		✓(*)					✓(*)					✓
Scheduling	Task												✓
	Pod												✓
	Service	✓											✓
	Stateful vs Stateless containers		✓										✓
Container Migration	Inter versus Intra cluster migrations												✓
	Migrations at cloud/ Edge/fog	✓(*)		✓(*)	✓(*)								✓
	Simulations versus real-world testbed migrations			✓(*)									✓
Container Scaling	Proactive versus reactive scaling decisions		✓										✓
	Horizontal, Vertical and Hybrid scaling							✓(*)					✓

✓ := method supports the property, *:= just an Overview/Visionary

learning (CMTL) for cyberattack detection systems in the healthcare industry. Finally, the authors [71] presents a new approach called CoDoC, which stands for complementary-driven deferral-to-clinical workflow. Its purpose is to decide when to rely on a diagnostic AI model and when to hand it off to a clinician.

3.2 Related surveys and our contributions

Table 1 shows the comparison of our systematic review with related surveys by focusing on the advancement of AI at the edge, optimizing algorithms in constrained environments, solving training data scarcity with AI techniques, and using AI/ML for resource management in fog and edge computing settings. There, the different columns are set on the basis of various applications, their methods of optimization, and fruitful utilization. This survey covers several edge computing concepts and emphasises the application of AI on edge devices with constrained resources. The survey examines the optimisation of ML algorithms for such restricted environments and covers current IoT applications across several sectors, including industrial automation, smart homes, and autonomous vehicles. It also highlights the difficulties and possible paths for edge computing and edge AI research, offering a strong basis for further investigation in the field.

3.2.1 Related surveys

Considering the dynamic, diversified, and resource-constrained character of fog and edge computing settings [2], this research emphasises the potential of AI/ML, particularly reinforcement learning techniques [4], in addressing resource management challenges in these systems [6]. A thorough analysis of the literature was done to look at how AI/ML applications may be used to effectively manage resources in these kinds of situations [11]. A taxonomy was established to categorise and contrast different approaches [7]. Enhancing explainability, reducing variance, and boosting online training of AI/ML algorithms are highlighted as critical future research directions to adapt to the constantly changing fog/edge computing landscapes [16]. The study emphasises the significance and changing challenges of resource management [8], whereas a presentation framework is described that addresses the issue of sparse training data in emerging radio access networks by utilizing a range of methods, such as interpolation, domain-knowledge techniques, generative adversarial networks (GANs), transfer learning, autoencoders, few-shot learning, simulators, and testbeds [12]. The challenges are highlighted and presented by insufficient training data, and the crucial role that Automation powered by AI plays in the operation, optimisation, and troubleshooting of cellular networks is described [14]. The technique suggests an integrated strategy to improve data availability in cellular networks and includes a survey and taxonomy of current approaches to lessen this scarcity. In addition, the study emphasises the necessity for scalable, reliable solutions that take conditional contexts into account for generating high-dimensional data in radio access network applications also [9, 10]. Table 1 reveals a dearth of reviews on Edge AI, with most existing surveys and review papers [2, 4, 6–12, 14, 16] presenting an overview or vision of the technology rather than a comprehensive survey, systematic review, and detailed taxonomy. To the best of the author's knowledge, this is the first survey paper on edge AI that provides a thorough taxonomy and a systematic review and highlights future research topics.

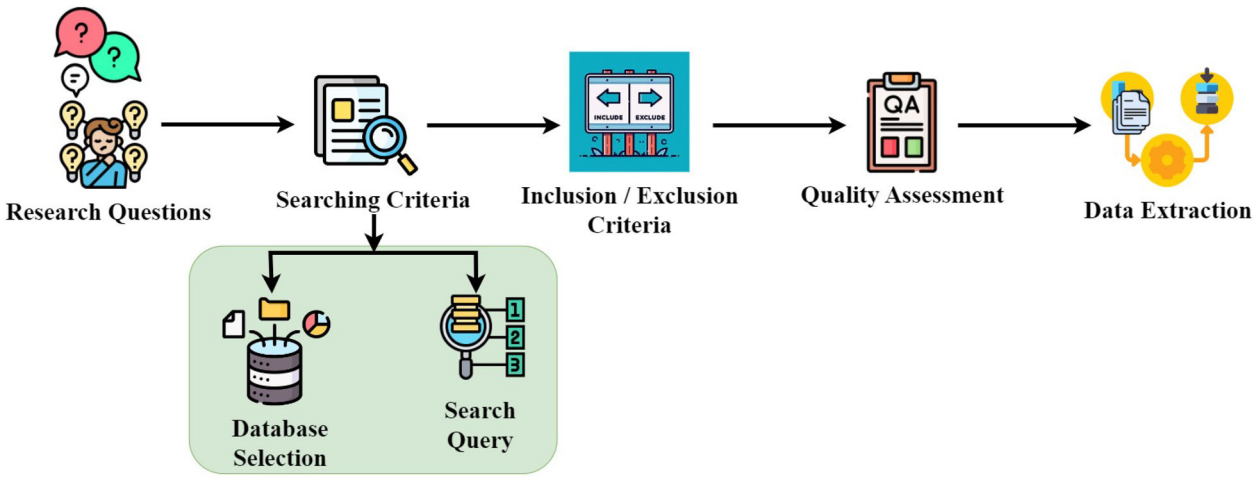


Fig. 5 Road-map for Research Methodology

Table 2 Research questions, motivation, category, and mapping

Sr. no.	Research question	Motivation	Category	Mapping section
RQ1	What are the fundamental techniques and strategies in Edge AI, and how do they impact data processing efficiency for different applications?	This research question aims to explain the effect of various approaches on the efficacy of Edge AI in real-world applications.	Methodologies and Strategies	Sects. 5.1, 5.2, and 5.4
RQ2	What strategies can be used to optimise heuristic and meta-heuristic algorithms for enhanced performance and resource management in Edge AI applications?	The question evaluates the efficacy of heuristic and meta-heuristic approaches in improving performance and resource management inside Edge AI systems.	Heuristic and Meta-Heuristic Methods	Sects. 5.4.1 and 5.4.2
RQ3	What are the challenges and solutions for using machine learning methodologies in Edge AI for real-time data processing?	The research question emphasises addressing the challenges encountered in the integration of machine learning techniques into Edge AI and possible solutions.	Machine Learning Techniques	Sect. 5.4.3
RQ4	How can edge AI applications, especially those that use dynamic environments, implement deep reinforcement learning to improve decision-making?	This question examines the function of deep reinforcement learning in enhancing real-time decision-making inside Edge AI environments.	Deep Reinforcement Learning	Sect. 5.4.4
RQ5	How can Edge AI increase scalability and resource efficiency with less computational power and storage size?	This question addresses the need for effective techniques for managing scalability and resources in Edge AI environments.	Resource Management and Heterogeneity	Sects. 5.7 and 5.11.2
RQ6	What are the principal issues in resource allocation and task distribution for Edge AI applications, and how can they be mitigated?	This question tries to determine challenges in the deployment of Edge AI and provide methods for efficient resource management.	Resource Provisioning and Workload Distribution	Sects. 5.5.1 and 5.5.4
RQ7	What effects do architectural choices have on the scalability and performance of AI applications running on the edge?	This question evaluates the efficacy of various architectural methodologies in Edge AI systems.	Architectural Comparisons	Sect. 6.2
RQ8	How does Edge AI resource management, including application placement and workload prediction, depend on important factors?	This question explores essential components that enhance efficient resource management techniques in Edge AI.	Resource Management	Sect. 5.5 and 6.5
RQ9	What are the main challenges to federated learning and how might it enhance data sharing and communication among distributed Edge AI devices?	Federated learning is potential for distributed Edge AI applications, but data synchronization is addressed in this question.	Federated Learning, Data sharing	Sect. 8.2
RQ10	What are the most important factors to consider when deciding between cloud, fog, and edge AI infrastructure, and how do these factors impact application performance?	This question examines factors impacting infrastructure decisions in Edge AI and its effects on performance.	Infrastructure Selection	Sect. 5.1 and 6.1
RQ11	When it comes to real-time applications, how can edge AI provide data privacy and security, especially in vulnerable domains ?	The significance and possible solutions for data privacy and security in Edge AI applications are the primary aim of this question.	Security and Privacy	Sects. 5.4 and 5.5

3.2.2 Our contributions

The ***main contributions*** of this paper are:

- We offer a thorough introduction to Edge AI, covering its history, challenges, and prospects.
- We conduct a systematic review that provided a thorough examination of edge AI research based on many applications, highlighting current trends and possible directions for the future.
- We propose a taxonomy for edge AI, which aids in the classification and arrangement of edge AI systems, and explore its potential impact across disciplines through various applications.
- We emphasize how important edge AI is for processing data in real time at the network's edge. It also highlights the challenges faced by edge AI systems, such as resource limitations, security risks, and scaling issues.
- We propose promising future directions that aim to address the current shortcomings of Edge AI by

Table 3 Overview of the criteria for determining inclusion and exclusion

S. no.	Inclusion	Exclusion
1	English articles issued at conferences, journals, and book chapters	Non-English articles
2	Articles that are included in a database source and are available in their entirety	Articles that are not available in their whole
3	Articles that specifically examine the process of choosing Edge AI infrastructure, such as Cloud, Fog, and Edge computing, and their effects on application efficiency and consumed resources	Articles that explore diverse domains such as federated learning, IoT-based approaches, and other classic methods
4	Articles published till 2024	Articles that were not published during the designated search timeframe
5	Relevant articles pertaining to the investigation queries	Articles that fail to meet the research requirements or receive a score of 3.5 or lower in the quality assessment standards
6	Systematic reviews often prioritise publications containing experimental or empirical research	Articles that do not contain such research

providing innovative solutions and opportunities for future research.

4 Review methodology

This article does a systematic review to categorize studies that are pertinent to this study domain or discuss specific research questions on "Edge AI". In this article, we used the guidelines established by Kitchenham et al. [72–74] to provide a comprehensive review of Edge AI. The review is the optimal and reliable approach to documenting and analyse current research works. The systematic approach enables researchers to carefully analyse the positive and negative aspects of recent studies, conduct a thorough examination to identify potential gaps in research and future trends and difficulties, and provide a solid foundation and context for establishing a new study field. Furthermore, the complete research approach is presented in Fig. 5, which represents the structure of the process that was used in the systematic review research.

4.1 Design and plan of review

The review procedure illustrates the methodologies applied to conduct a systematic review to minimize the potential for biased research. Therefore, possessing a pre-established process is crucial. In the absence of a systematic methodology, researchers' predispositions can influence the process of selecting and analyzing studies. This may result in the omission of crucial inquiries essential for a thorough analysis and comprehension of the subject matter. The review process encompasses the research inquiries, exploration approach, criteria for selecting studies, procedures for assessing quality, and techniques for extracting and synthesizing data [75].

4.2 Research questions

Determining the research queries is crucial in the method of planning to develop a strong systematic review. The design of the research problem requires a thorough examination of existing literature studies. The primary aim of the present systematic review is to thoroughly examine and evaluate the various methods and strategies being employed for edge intelligence or edge AI. Furthermore, in order to emphasise the research findings and effectively showcase the useful consequences, the research questions that follow have been defined in Table 2.

4.3 Search strategy

4.3.1 Database selection

The database selection includes conducting searches on various digital databases, such as IEEE Xplore, ACM Digital Library, Wiley, Taylor and Francis, Springer Link, Google Scholar, and Science Direct. These databases contain a wide range of impact factor journals, magazines, and significant conference proceedings, making them suitable for this systematic review.

4.3.2 Search query

A comprehensive search was conducted utilising Logical OR/AND operators to connect the keywords, concepts, synonyms, and abbreviations. The initial phase entails conducting an automated search using predetermined keywords that align with the study topics of this systematic

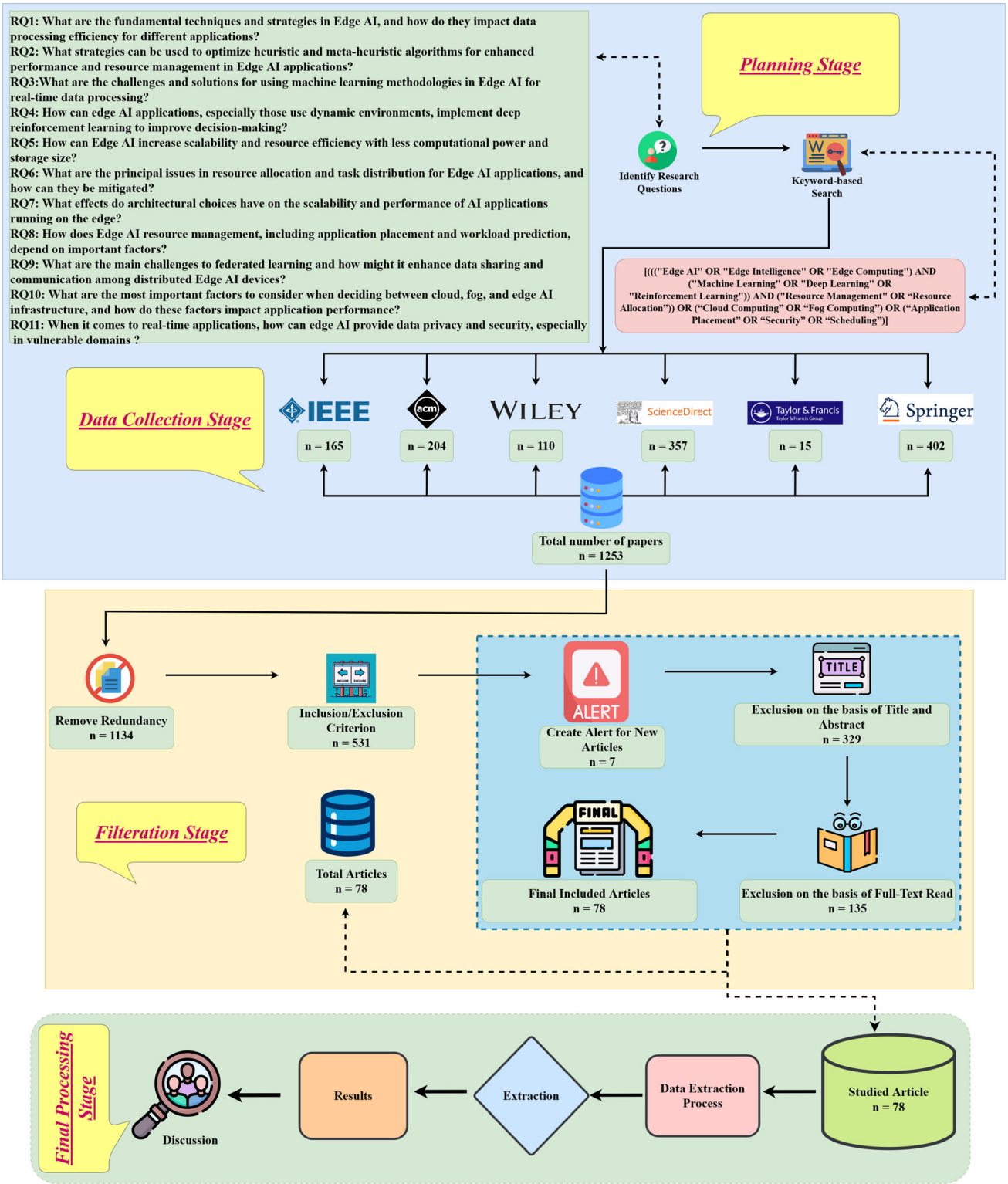


Fig. 6 Research Methodology Protocol

literature review (SLR). The keywords used are [((("Edge AI" OR "Edge Intelligence" OR "Edge Computing") AND ("Machine Learning" OR "Deep Learning" OR "Reinforcement Learning")) AND ("Resource Management" OR "Resource Allocation")) OR ("Cloud Computing" OR "Fog Computing") OR ("Application Placement" OR "Security" OR "Scheduling" OR "Simulation")]. The search terms are obtained from the

specified research topics and the framework of this systematic review in order to encompass the most significant and interrelated publications.

4.4 Inclusion and exclusion criterion

The systematic review needs to establish clear guidelines for inclusion and exclusion to guarantee that the chosen papers are relevant to the research topic and address the specific research objectives. The primary purpose of establishing the criteria is to guarantee that the studies featured are appropriate and connected to AI-based methodologies in Edge computing. Hence, the chosen research must satisfy all the predetermined criteria. Table 3 presents the specific phrases used to determine which criteria were included and excluded in this systematic review.

Furthermore, a method of screening is carried out to identify the appropriate research studies that are relevant to the context of this study shown in Fig. 6. The screening process consists of three distinct stages:

a. **Title and abstract Phase:** During this stage, papers that were deemed irrelevant were excluded based on their title and abstract. Subsequently, the studies that satisfy at least some of the criteria listed in Table 3 are chosen and advanced to the subsequent step for additional analysis.
b. **Full-text screening Phase:** During this step, studies were excluded while they failed to fulfill the criteria specified in Table 3, based on a thorough reading of the full-text or partial reading.
c. **Final selection Phase:** The next phase utilises the criteria terms outlined in Fig. 6 to make the final selection and eliminates studies that do not meet any of the specified criteria.
    i. The topic of the study must be pertinent and directly connected to the research topics.
    ii. The user did not provide any text. The research study examines the comprehensive solution for research advancements in edge intelligence and identifies four key components: monolithic and microservices architectures differ in terms of flexibility, performance, and resource utilisation in Edge AI for addressing practical challenges, finding solutions, and achieving optimisation goals.
    iii. The research paper presents essential factors to consider for managing resources in Edge AI, such as resource provisioning, allocation, deployment, and scheduling of workloads.
    iv. The user did not provide any text. The research study examines the factors that influence the choice of Edge AI infrastructure, such as Cloud, Fog, and Edge computing, and investigates their effects on application reliability and resource utilisation.
    v. The user did not provide any text. How do Edge AI systems maintain reliable and intelligent tasks in dynamic and ambiguous instances?

4.5 Quality assessment

In order to gather the most comprehensive and reliable information on this subject, we employed a systematic review methodology, following the standards outlined [76]. Furthermore, a plethora of research papers and conference papers exist on the topic of AI applied to edge computing. Once we applied the criteria for inclusion/exclusion, researchers conducted a thorough evaluation of the articles that met the standards to identify the ones that were most worthy of further examination. Employed the standards established by the methodology to evaluate the overall quality of the research, taking into account its impartiality, internal consistency, and objectivity.

4.6 Extraction and synthesis

This phase emphasises the process of extracting and combining data by thoroughly examining all 78 chosen studies and summarising and storing the relevant information. This stage involves the creation of a mechanism for extracting data items and compiling comprehensive reports that include all the information gathered from primary research studies [75].

Furthermore, this study specifically chose items that relate to the research objectives as well as research questions. The data was taken from primary studies and meticulously recorded to determine the ultimate findings of the systematic review. The analysis step involved employing descriptive synthesis. The subsequent section discusses the results obtained from the synthesis.

5 A taxonomy of Edge AI

In this taxonomy, we identify and categorize the current studies for Edge AI. To create a new taxonomy, the first three authors carefully examine the contents of the 60 papers available and then obtain the research summary as explained in Sect. 4. We classify the obtained solutions under the necessary headings. Based on the current advancements and existing literature on edge AI, we have proposed a taxonomy, as shown in Fig. 7. Leveraging edge AI-driven edge computing systems can be optimized. The best example of this is the EdgeAI-based latency-reducing

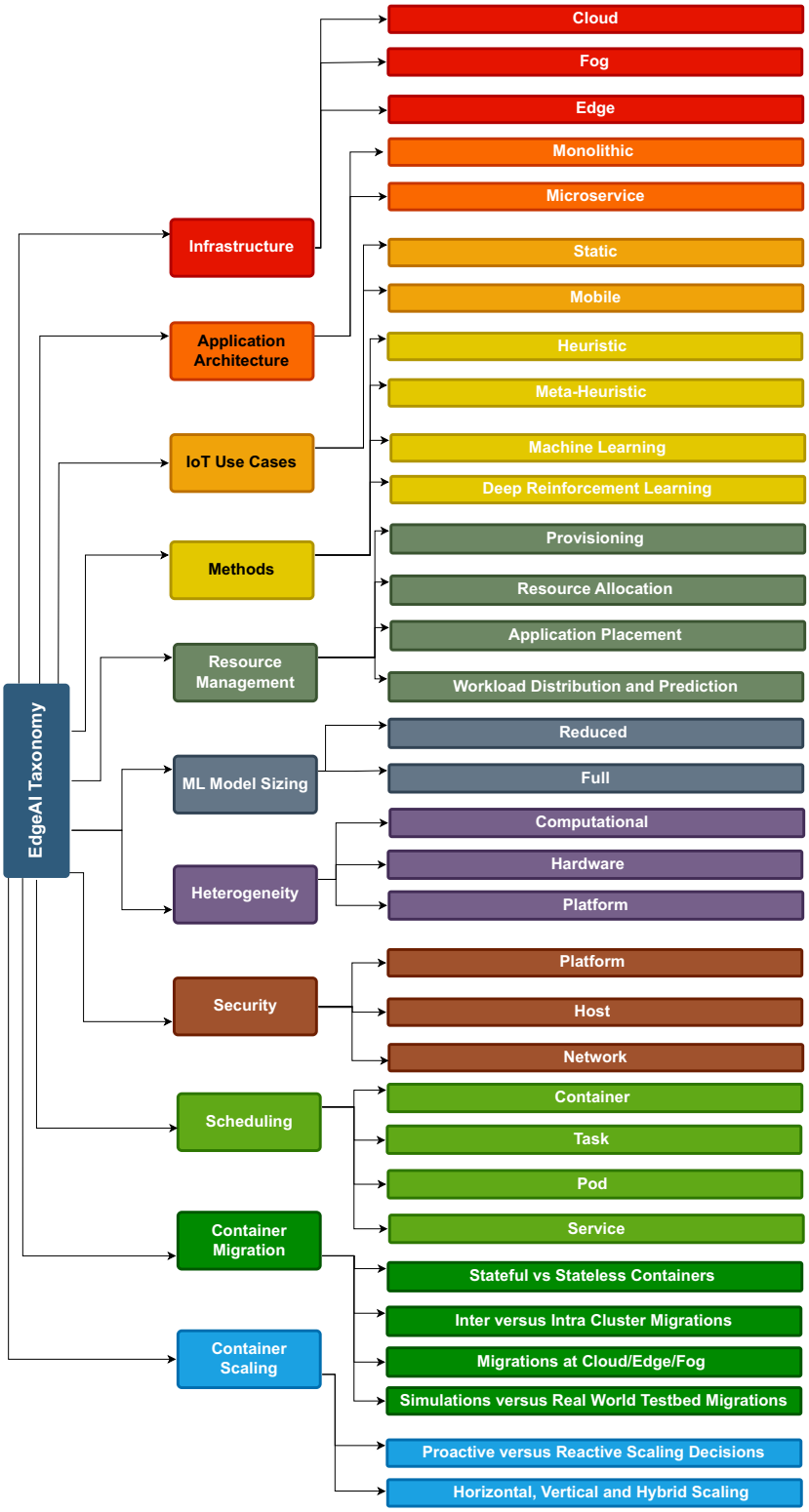


Fig. 7 The Taxonomy of EdgeAI

studies found in the literature. We generally examine literature studies under 11 subheadings: infrastructure, application architecture, IoT use cases, methods, resource management, ML model sizing, heterogeneity, security, scheduling, container migration, and container scaling.

5.1 Infrastructure

AI models and applications can be deployed in different infrastructures based on their target scenarios. The three most common infrastructures widely used are Cloud, Fog, and Edge.

5.1.1 Cloud

The essence of cloud computing is to pool resources with virtualization technology. Virtualization technology transforms a physical machine into several virtual machines, greatly changing the mode of application operation and deployment. The term 'cloud' here refers to remote data centers [21]. Users usually connect to the servers in the data center through the network to use the computing resources here. The essence of cloud computing is to pool resources, allowing users to purchase resources according to their own requirements, greatly reducing resource waste.

Therefore, through cloud computing, the performance of AI applications has also been significantly improved. We transmit local AI models to remote servers through the network. We can more efficiently utilize some computing resources in the cloud, such as CPUs, GPUs, and memory, and flexibly scale according to our business needs, improving resource utilization and reducing the cost of model training and prediction.

5.1.2 Fog

As is well known, fog is closer to the ground than cloud, and the ground here refers to the user's local device. Fog computing technology adopts distributed computing technology, which arranges computing resources on the side closer to the user than cloud computing [6]. It can be said that fog computing has broadened the network computing mode of cloud computing, widening computing capacity from the network center to the network edge, and thus more widely applied to numerous services. Fog computing is more widely distributed geographically and has greater mobility, making it suitable for an increasing number of intelligent devices that do not require extensive computation. For some time delay-sensitive applications such as real-time interaction system, fog computing also has greater advantages.

Image, video and natural language processing (NLP) are the recently emerging applications of fog computing. The placement and processing of images in fog computing is one of the most widely used fields of AI in research and industry. Its goal is to distinguish objects and people from each other and classify and distinguish photos based on image processing algorithms. Using fog computing in image processing-based applications can shorten response delay and improve service quality. In medical applications that require image processing accuracy and fast processing of medical data, deploying effective scheduling algorithms in foggy environments may be beneficial. According to other works, DL algorithms such as Generative Adversarial Networks (GAN) and Convolutional Neural Networks (CNN) can commonly be used in the field of image processing in fog.

5.1.3 Edge

The concept of edge computing is relative to cloud computing. The processing method of cloud computing is to upload all data to the cloud data center or server in the computing resource set for processing. Any request to access this information must be submitted to the cloud for processing [35]. Edge computing is a computing model that moves resources and provides edge intelligent services at the network edge near objects or data sources, by which improve service quality and information security. In brief, edge computing analyzes the data produced by the terminal directly in the local device or network near the data generation, without transmitting the data to the cloud data processing center [68]. Compared with cloud computing, edge computing has shorter network latency, less resource consumption, and higher security of the data to the cloud data processing center.

The so-called edge AI combines two emerging technologies: edge computing and AI. However, the implementation of edge computing is based on the same basic premise, that is, generating, collecting, storing, processing and managing data locally rather than in remote data centers. Edge AI improves this concept to the device level, using ML to imitate human reasoning to reach user interaction points, such as computers, edge servers or IoT devices [69]. Typically, these devices can operate without an Internet connection and make decisions independently. Well-known examples of edge AI technologies include virtual assistants, such as GPT-4o, Apple's Siri, or Amazon Alexa. When the user says "hey", these tools will recognize and learn what the user is saying (i.e. ML), interact with cloud-based application programming interfaces (APIs), and store the learned knowledge locally.

5.2 Application architecture

Monolithic and microservice, as the two main application architecture patterns [77], each has unique advantages and application scenarios that can support AI applications and models.

5.2.1 Monolithic

Monolithic Architecture, was the traditional model for software development and deployment, used in the past by large companies. And its functionality is encapsulated into one single application. The advantage of this architectural pattern is that it is much easier to develop, deploy, debug and monitor [78]. Because the interaction between components is directly completed via memory, the performance is better. For some simple and initial AI systems, or applications with small business scales and infrequent changes in requirements, monolithic architecture may be a suitable choice.

However, with the arising of the business scale and demand, MA has encountered many challenges and shown obvious shortcomings. Due to its inherent tight integration, this design pattern has limitations in scalability, adaptability, and ease of maintenance. Whenever specific modules need to be extended, the entire application needs to be recompiled and deployed, which can lead to low resource efficiency and waste [20]. In addition, as the code repository grows, the complexity of development, testing, and deployment also increases, thereby increasing maintenance costs and the risk of errors. Therefore, in large and complex applications, developers are seeking more flexible and scalable architectural patterns, such as microservices architecture, as introduced in the next section.

5.2.2 Microservice

Microservice, as an emerging distributed system architecture model, is gradually changing the landscape of software development. The core idea is, unlike monolithic, to decompose complex large-scale applications into small and independent service units, each focusing on specific business functions or integration domains. Considering the fact that microservices architecture is composed of multiple small components, iterative upgrades of applications are more flexible and efficient [20]. Especially for large and complex AI systems, such as e-commerce platforms, social media platforms, etc., microservice architecture is a more suitable choice. These applications typically consist of multiple subsystems, each of which can be independently developed. For example, Taobao has dozens of independent systems, all of which are typical microservice architectures that can support rapid business development and iteration.

Even so, with the widespread application of microservice architecture, it also faces challenges such as service governance, network transmission efficiency, service expansion, and version iteration. Nevertheless, microservice architecture remains an important development direction for enterprise IT architecture, and its potential and advantages cannot be ignored [79]. Future research will require an in-depth exploration of how to overcome these challenges and further promote the development and application of microservice architectures.

5.3 IoT use cases

IoT is truly revolutionary, presenting an extensive array of diverse and impactful use cases that are reshaping numerous aspects of our modern world. We can classify IoT use cases into two main categories: static and mobile. As the name suggests, static use cases are like agricultural monitors, which are fixed and usually do not need to be moved, while mobile use cases such as in-car telemetries and wearable devices may frequently move [80].

5.3.1 Static

Static IoT user cases based on edge AI technology are gradually becoming a key driving force for digital transformation. In these cases, data collected by static IoT devices (such as cameras, sensors, etc.) is directly analyzed in real-time on edge devices, and rapid decision-making and response are achieved through edge AI [79]. In the field of agricultural ecological environment monitoring, the agricultural IoT predominantly employs high-tech approaches to establish sophisticated agricultural ecological environment monitoring networks, and uses wireless sensor technology, information fusion transmission technology, and intelligent analysis technology to perceive changes in the ecological environment [4]. In 2002, researchers at the University of California, Berkeley conducted a 9-month periodic environmental monitoring of the habitat of the Shanghai Swallow on Duck Island using wireless sensor networks. Regional static MICA sensor nodes were deployed to achieve unmanned and non-destructive monitoring of sensitive wildlife and their habitats. Some countries, including the United States, France, and Japan, have primarily focused on integrating the establishment of agricultural information platforms covering the whole country to achieve automatic surveillance of the agricultural ecological environment and ensure its sustainable development of the agricultural ecological environment.

5.3.2 Mobile

The emergence of emerging AI such as DL has brought new innovations to mobile animal networking. In the field of mobile IoT, edge AI achieves real-time and efficient data processing by deploying the computing power of AI at the edge of devices. For example, in smart health monitoring applications, edge AI enables wearable devices to analyze user physiological data in real-time, providing

real-time health feedback to users without uploading data to the "cloud" for processing [81]. These devices harness DL technologies to scrutinize user health metrics and furnish guidance for subsequent lifestyle modifications, yielding substantial benefits, particularly for infants, young children, and the elderly. Due to data not being uploaded to the cloud, it can effectively protect user privacy and security. In addition, in the field of intelligent transportation, AI technology provides timely guidance for vehicle operation by analyzing local data. Similarly, many frameworks utilize DL techniques to predict parking occupancy rates, reduce the time required for vehicles to search for parking spaces, and improve urban traffic management [6]. These user cases demonstrate the enormous potential of edge AI technology in improving the performance of mobile IoT applications and enhancing user experience.

5.4 Methods

In this part, we discuss the dominant methods employed in edge AI which include heuristic algorithm, meta-heuristic algorithm, ML and Deep Reinforcement Learning (DRL). These four types of algorithms have their own characteristics and application scenarios, aiming to optimize the accuracy and performance of AI models and applications.

5.4.1 Heuristic

People often refer to methods inspired by the laws of nature or the experiences and rules of specific problems as heuristic algorithms. The current heuristic algorithms are not entirely based on natural laws, but also come from human accumulated work experience. They supply a practical solution for each instance of the combinatorial problem to be solved and maintain an acceptable cost in terms of computation time and space, and the degree of deviation between the practical solution and the optimal solution may not be ascertainable before the implementation in advance [31]. Heuristic algorithms are a technique that enables the search for the best possible solution within an acceptable computational cost, but may not necessarily guarantee that the obtained solution is the best scheme. In most cases, it is impossible to describe the degree of approximation between the obtained solution and the optimal solution.

In the field of edge AI, heuristic algorithms have been widely used because they conform to the characteristics of human cognitive thinking. Usually, edge devices have limited computing resources [82], so reasonable resource scheduling is particularly important. Heuristic algorithms can find satisfactory solutions in a relatively large search space within a short amount of time, so they can help our applications deploy to suitable nodes. In addition, heuristic algorithms can perform efficient data analysis on edge devices, greatly reducing dependence on cloud computing resources and lowering network response latency [83].

5.4.2 Meta-Heuristic

Meta heuristic is a computational intelligence-based mechanism used to solve complex optimization problems for optimal or satisfactory solutions. The meta heuristic algorithm obtains a sufficiently good solution by searching the space [69]. Meta heuristic algorithms can be seen as an algorithmic framework that can be applied to different optimization problems with slight modifications. In edge AI, the application of meta heuristic mainly focuses on two aspects:

Model optimization: The reasonable deployment of AI applications in edge devices has always been a headache inducing issue. AI applications are computationally intensive services that require high computing resources, while computing resources in edge devices are usually limited [38]. The fully heuristic algorithm can help find the optimal parameter configuration for AI models, so that these AI applications can maintain high performance while occupying as few resources as possible.

Resource management: In edge AI systems, multiple tasks may need to run at the same time and they need to share limited resources [84]. The meta-heuristic algorithm can optimize the allocation of these resources, ensuring that each task receives sufficient resources to run efficiently without depleting the entire system's resources.

5.4.3 Machine learning

As is well known, the ML technique forms the foundation of AI and plays a pivotal role in many applications, such as recommendation systems, text generation, and so on. As edge AI gains prominence, ML is increasingly finding its way onto devices located on the fringes of the network. These edge devices prioritize data processing close to its source, aiming to minimize transmission delays, accelerate response times, and alleviate the reliance on centralized servers [6]. Consequently, edge AI systems are often tasked with handling vast volumes of data in real-time scenarios. Following certain ML algorithms, such as sophisticated DL models, these systems can swiftly analyze and process data at the edge, extract crucial insights, and empower a diverse array of real-time applications encompassing autonomous vehicle operations, intelligent manufacturing processes, and robust security systems. In addition, ML techniques can be employed to detect anomalies in real-time on edge devices, such as detecting product quality issues during the manufacturing process, or detecting changes in patient health status in the medical field [85].

5.4.4 Deep reinforcement learning (DRL)

Many application problems in AI require algorithms to make decisions and execute actions at every moment. For Go, each step requires determining where to place the pieces on the chessboard in order to defeat the opponent as much as possible; For autonomous driving algorithms, it is necessary to determine the current driving strategy based on road conditions to ensure safe driving to the destination; This type of problem has a common characteristic: to make decisions and actions according to current conditions in order to achieve a certain expected goal [86]. The ML algorithm used to solve such problems is called reinforcement learning (RL). Although traditional reinforcement learning theories have been continuously improved in the past few decades, they are still difficult to solve complex problems in the real world.

DRL is a type of DL with reinforcement learning methods, which enables models to have stronger learning abilities, indicating that machines can autonomously understand and learn the human visual world. Simply put, just like humans, this means inputting visual and other perceptual information, and then directly outputting actions through deep neural networks without the need for manual production. As introduced before, DRL can solve specific problems in edge devices, such as in car systems where the device perceives the surrounding environment and road conditions on its own without the need for human intervention, and selects the appropriate driving route [87]. In addition, DRL has also driven the development of other fields, such as smart homes [4].

5.5 Resource management

Edge computing, which deploys computing, storage, network, and other resources at the network's edge, can drastically minimize data transmission delay, enhance data processing efficiency, and relieve bandwidth demand on the core network [4]. However, with the growing number of edge devices and the complexity of applications, how to efficiently manage these resources has become an urgent problem that must be addressed. This section will review how to make resource management in an edge environment from the aspects of resource provisioning, resource allocation, application placement, and workload distribution and prediction.

5.5.1 Provisioning

As the name suggests, resource provision is the way of provision of resources by resource suppliers based on the users' pre-established needs and supply strategies. This type of strategy is usually divided into two types: dynamic and static strategies. Static strategies are determined based on user resource needs and constraints, such as QoS and SLAs [4, 88, 89]. Static strategies are more suitable for stable workloads. For applications with large fluctuations in resource demand, we usually use heuristic algorithms and ML algorithms to predict in advance, which are called dynamic strategies.

5.5.2 Resource allocation

Edge AI is an offline service during model training, different from traditional online services such as microservices, web applications, and API services. Sometimes, it is not sensitive to latency, but usually requires higher resources such as GPU and memory. The computing power of edge devices is often limited by their limited resources, which are often unable to support or complete computationally intensive tasks, such as model training, within an acceptable deadline. This approach at this point is to offload these offline tasks and transfer them to edge servers for completion [1]. Edge devices are used for services that need to meet low latency, such as web services and human-computer interaction. Of course, we can also combine some DL techniques to achieve more reasonable resource allocation. This approach aims to minimize resource waste and enhance application performance.

5.5.3 Application placement

Application placement is also an indispensable factor to consider in edge computing resource management, which means formulating an assignment of applications to servers that maximize the QoS for all users in order to optimize the performance for some components that are sensitive to latency such as interactive online games, face recognition, etc. Through reasonable application placement, such as some AI-based approaches [90], the computing and storage resources of edge nodes can be fully utilized, improving resource utilization and efficiency.

5.5.4 Workload distribution and prediction

Generally, the infrastructure architecture where our application can be described as three separate layers: cloud layer, edge layer, and IoT layer. The workload of the application will have its own characteristics distributed in these three layers [91]. The cloud layer is located at the top level of the architecture and The cloud layer, situated at the top of the architecture, is a robust cluster comprising thousands of virtual machines. Deploying large-scale AI models on the cloud layer is a good choice. The edge layer is located between the cloud layer and the IoT layer composed of a set of nodes that deploy several devices like

routers or switches, and it is responsible for load-balancing traffic from the cloud layer and aggregating and analyzing data from the IoT layer. We can deploy some web servers and small-scale AI applications on this layer. The IoT layer is the source of data in this architecture, mainly composed of sensors and wireless devices, such as temperature sensors, cameras, and Bluetooth [82]. These devices need to have high sensitivity and feedback very quickly to user instructions.

Furthermore, accurate predictions can lead to more rational resource management. This is beneficial for both users and service providers. Previous research and technologies such as ARIMA [92] and Holt Winter [93] were based on linear temporal prediction. However, these technologies often exhibit poor prediction accuracy. With the rise of DL, technologies such as neural networks have been widely applied in data prediction, such as weather, transportation, and finance systems. Especially for recurrent neural networks (RNNs), their inputs not only focus on current data, but also contain information from a period of time in the past. Therefore, it is often used to predict time-series related data. Using RNNs to predict workload in edge computing is a promising approach worthy of consideration.

5.6 ML model sizing

In the edge computing scenario, the size of the model becomes the key factor to determine whether it can be successfully deployed and applied. Especially in intelligent camera monitoring systems, due to the limitations of edge devices in computing power, storage space, and energy supply, using a full model size DL model is often impractical [94]. To overcome these limitations, we usually adopt a strategy of reducing model size. In this section, we analyze and compare the two different strategies of model sizing.

5.6.1 Reduced

The training cost and efficiency of AI models are important metrics to assess the quality of the model. Nowadays, many AI giants are progressively increasing the size of model parameters and the volume of training data. The model parameters of GPT−3.5 have reached 175 billion [95]. While this approach significantly enhances model accuracy, it also markedly escalates training costs and hardware requirements, necessitating a trade-off between accuracy and cost. In this way, we need to strike a balance between accuracy and cost. In recent years, there have been numerous studies in this area, such as DenseNet [96], EfficientNet [97] and EfficientNetV2 [98]. The goal of these works is to train models to achieve satisfactory accuracy with fewer model parameters. In the field of edge computing, where hardware resources are limited, the reduced-size model will certainly become an important trend of edge AI in the future.

Model pruning and model quantization are well-established methodologies for achieving model size reduction. However, nowadays, how to design a lightweight and high-precision neural network has become a focal point of research in the field of AI, such as MobileNet [99] and ShuffleNet [100]. These models have small parameters and high computational complexity, making them very suitable for running on edge devices.

5.6.2 Full

Unlike MobileNet and ShuffleNet, GPT-3 is a full model with 175 billion parameters, which is hundreds of times the number of GPT-2 parameters (3 billion). Tom Brown [101] demonstrated that GPT-3 has completed various NLP tasks, such as translation, question answering, etc., with minimal sample training. Due to its outstanding performance in the domain of NLP, this model has greatly promoted the development of large language models. Currently, many edge computing frameworks, such as KubeEdge [102], have integrated plugins that support the deployment of these extensive language models, thereby extending their applicability and utility in edge environments.

5.7 Heterogeneity

Heterogenous environments in edge devices are employed to run various IoT applications. Their diversities are embodied in three aspects: computational heterogeneity, hardware heterogeneity and platform heterogeneity.

5.7.1 Computational

Computational heterogeneity in edge computing emphasizes the variability in application behavior during computational operations. For AI applications, there is a large amount of vector operation logic in the model code, which determines that such applications are suitable for parallel computing rather than serial computing. For web services, universal computing is the main approach [103]. This distinction manifests in hardware requirements, where AI applications rely on GPU acceleration, whereas web services operate efficiently with CPU resources alone.

For many microservices, such as the web services mentioned earlier, their performance bottleneck often is not in CPU but in disk read and write speed, as most of the time is spent accessing databases. In other words, they are IO-intensive services rather than computationally

intensive. In order to reduce network latency, microservices are usually deployed in edge nodes. How to reduce the performance loss caused by slow disk read and write speeds is a problem we need to consider.

5.7.2 Hardware

Edge devices have many differences in processor and hardware architecture due to the computing characteristics of the applications deployed on them. The instruction set of CPU can be divided into two categories: ARM and AMD, and software running on different instruction sets may have differences in performance. Some infrastructure deployed on edge nodes, such as routers and switches, are responsible for tasks such as data forwarding and protocol conversion, and therefore require CPU support. The application of image generation and virtual reality requires high-performance graphics rendering and environment recognition. In addition to processing CPUs, GPUs and other chips are also necessary. Nowadays, many container frameworks have support for CPU and GPU hardware resource isolation, such as Docker and Container, which can eliminate the impact of service instability caused by hardware resource competition.

In 2018, Google launched Edge TPUs [104], specially designed for inference and training of neural networks on edge devices with limited resources. Edge TPUs demonstrate strong capabilities in computer vision [105]. Some IoT applications for autonomous driving and facial recognition can benefit greatly.

5.7.3 Platform

Due to the rise of edge computing, the world's major technology giants have also launched their own edge computing platforms. For example, Amazon's AWS IoT Greengrass, Microsoft's Azure IoT Edge, and Google's Cloud IoT Edge. They all support the effective operation of AI models on edge devices, providing service management and data analysis capabilities. Other open source platforms also deserve attention, such as KubeEdge and OpenYurt, which are extensions of Kubernetes in the field of edge computing and provide container management, automatic operation and maintenance and other functions.

5.8 Security

With the rapid development of edge computing techniques, an increasing number of enterprises and organizations are deploying edge computing solutions to meet high demands for real-time capabilities, security, and privacy protection. However, simultaneously, the edge computing environment also faces numerous security challenges [6, 106]. To ensure the stable operation of edge computing systems and data security, we need to consider and ensure security comprehensively from three aspects: Platform, Host, and Network.

5.8.1 Platform

Blockchain is a distributed, decentralized, and tamper proof database. It is often used to build a secure and trusted intelligent platform, which can solve the security problems in edge computing. Zhang et al. [107] utilized blockchain technology to construct a highly secure trusted edge platform, providing a secure environment for AI applications on edge nodes. Wang et al. [108] proposed an integrated trust evaluation mechanism based on cloud and edge computing, along with a new architecture of service templates and balanced dynamics, to address security challenges. In this architecture, the design of edge networks and edge platforms is aimed at reducing resource consumption and ensuring the scalability of trust evaluation mechanisms, respectively. Other security technologies such as Role Based Access Control (RBAC) have also been widely applied to some distributed platforms, such as Kubernetes.

5.8.2 Host

Host security is defined as the security of all hardware and software deployed on a single edge server or device. Due to the proximity of edge devices to the human body, such as healthcare systems and intelligent driving systems. Imagine that if a car is using intelligent driving and its intelligent driving system is hacked, it will pose a serious threat to the safety of passengers and other vehicles on the road [109].

We can take many measures to defend against external attacks on the host. Firewall rules can be configured to block access from unauthorized IP addresses. Moreover, by installing antivirus software and regularly updating patches, the security factor can also be improved.

5.8.3 Network

Distributed Denial of Service (DDoS) attack, which causes significant economic losses to society every year, is one of the most common attack methods in computer networks, and it also has strong destructive power on IoT devices. From the time of the 2016 botnet Mirai attack on KrebsOnSecurity [110] and Dyn [111], it can be seen that DDoS attacks are seriously threatening the security of IoT applications. With the development of edge computing, the threat of such attacks to large-scale IoT devices is growing, which may lead to incalculable economic losses. For example, in the field of automation, AI technology is

widely used to make decisions and adjust plans. If edge AI is subjected to network attacks, it can lead to AI models making incorrect decisions, resulting in product quality issues.

Although edge nodes exhibit the potential to isolate most of the IoT data at the network edge and detect and intercept attacks near the source in the first place, they encounter significant challenges in practical applications. The main reason is that edge nodes are unable to capture the aggregated network traffic required for IoT DDoS detection, nor can they scale and provide the necessary resources like elastic clouds [112]. Therefore, directly deploying existing cloud based defense solutions on edge nodes is far from achieving ideal results. We need to redesign the DDoS defense scheme based on edge computing to solve the special and severe security problems in the edge environment.

5.9 Scheduling

Resource scheduling is the process of efficiently allocating and managing system resources, ensuring optimal utilization of resources according to demand and priority. In edge environments, resource scheduling is exceedingly crucial for achieving real-time, low-latency services. Especially in AI scenarios that have high demands for computing and network resources, only by allocating edge device resources reasonably can we support the rapid response and efficient operation of AI applications, and improve overall system performance and user experience. Resource scheduling is also a hot research direction, and there has been a lot of work in this area before [4, 6, 106]. In this section, we discuss scheduling from the following four granularities, because the four constitute the core unit for application deployment and management in container orchestration systems such as Kubernetes and KubeEdge.

5.9.1 Container

Containers are a software virtualization technology that laid the foundation for the development of microservices. Nowadays, containers are also widely used in the field of edge computing. There is also much research on containers in edge computing, which stems from the growing demand of users for millisecond delay computing. In [113], the authors elucidate the concepts of container placement and migration between edge servers, and propose a container scheduling framework grounded in multi-objective optimization models or graph network models.

In addition, some open-source container orchestration and scheduling frameworks are worth paying attention to, such as KubeEdge. KubeEdge can extend its powerful cloud computing capabilities to edge devices. Especially suitable for some AI applications, model training can be completed in the cloud and then deployed to edge devices. In addition, KubeEdge can optimize scheduling performance based on different AI application business scenarios by configuring the algorithm and parameters of the kube-scheduler.

5.9.2 Task

In the scheduling task of edge computing, there are usually two problems to be solved: scheduling time and resource allocation. Most previous research [114–116] on task scheduling has focused on these two aspects. With the advancement of AI technology, ML technology has shown unique advantages in task scheduling. Markov Decision Process (MDP) is a robust and effective method for modeling temporal data and providing high-precision predictions. The problem of resource allocation in edge devices can be described as MDP, and the deep Q network (DQN) algorithm uses multiple replay memories to minimize the total delay and resource utilization. the study in [117] addresses the intricate issue of joint task offloading and resource allocation problems for computationally intensive tasks in fog computing. This intricate problem is formulated as a partially observable MDP, and the Deep Recursive Q-Network (DRQN) algorithm is adopted to approximate the optimal value function.

5.9.3 Pod

In container orchestration systems such as Kubernetes, Pod is the smallest unit of work composed of several containers. Pod scheduling is the process of assigning Pods to a node based on a certain algorithm strategy, which is of great significance for ensuring high availability, resource utilization, and performance of the system. Pod scheduling is mainly controlled by kube-scheduler, and its process includes two stages: screening and scoring [118]. During the filtering phase, the scheduler checks all nodes to determine which ones have the resources (such as CPU and memory) and other requirements (such as node selector labels) needed to run Pod. Then, the selected node will enter the scoring stage, and the scheduler will rate each node based on a series of criteria such as node affinity, resource utilization, etc. The node scoring the highest will be designated as the running location for Pod. kube-scheduler supports custom scheduling plugins, and users can develop some extension plugins based on the business characteristics of the enterprise.

5.9.4 Service

Service refers to software or system components deployed at the edge of a network that provides specific functions or resources to meet the real-time, low latency, and high bandwidth needs of users or devices. Service can be a computing service, data processing service, storage service, or any form of network service that optimizes resource utilization and reduces data transmission latency, bringing better service quality and experience to users.

Service scheduling is the deployment, allocation, and scheduling process for these services [59]. In the edge computing environment, Service Scheduling is responsible for arranging and scheduling the execution sequence and location of services reasonably based on application requirements, resource conditions, and network conditions. Effective service scheduling can ensure that the service can efficiently use limited edge computing resources, achieve load balancing, reduce service latency, and improve the performance and reliability of the entire system.

5.10 Container migration

In distributed and cloud computing environments, containers need to be migrated from one node to another due to node failures, load imbalance, resource upgrades, and other reasons. At this point, container migration technology is needed to achieve rapid migration and recovery of containers.

5.10.1 Stateful vs stateless containers

Stateful containers and stateless containers are two major classifications of containers, which are important criteria for container expansion, contraction, and migration.

(i) Stateful containers: The so-called state essentially refers to the data in the running container. When migrating such containers, it is usually necessary to migrate their data together, such as a database. Due to the involvement of data replication, such containers need to consider issues such as data loss and data integrity. Specific migration tools or strategies may be needed to ensure accurate migration and recovery of data [119]. All containers managed by a StatefulSet controller in Kubernetes are considered stateful.

(ii) Stateless containers: These containers are containers that do not save any state during runtime. For example, a web server that provides services for static pages, treats each request as independent, and the container does not need to remember previous interactions. The migration process is very simple, just pull up the container on other nodes and delete the container from the original node. All Pods under a Deployment in Kubernetes are stateless.

5.10.2 Inter versus Intra cluster migrations

Inter-cluster and intra-cluster migration are discussed briefly below:

(i) Inter-cluster migration: When a company or organization needs to migrate its data center from one geographic location to another, inter-cluster migration is an indispensable step. Inter-cluster migrations involve node migration between different clusters, typically requiring consideration of cross-cluster communication factors such as network latency and bandwidth limitations [120]. Due to the collaborative work of multiple clusters and nodes involved in cross-cluster migration, the migration process is relatively complex and requires ensuring data consistency and service continuity.

(ii) Intra-cluster migration: In a cluster, migration within the cluster can take effect when a node experiences performance degradation or longer response time due to excessive workload. Administrators or automation tools can migrate a portion of the workload (such as containers, virtual machines, or services) on that node to other nodes in the cluster to balance the load and optimize performance [63]. Compared to inter-cluster migrations, The complexity of intra-cluster migrations is relatively low because it only involves nodes and data migration within the same cluster.

5.10.3 Migrations at cloud/edge/fog

Migration at cloud is the process of migrating applications, data, and other business processes from traditional local devices or servers to cloud platforms, including the migration to IaaS, PaaS, and SaaS [121]. IaaS migration is the most ideal and applicable cloud migration solution. Because we can entrust all programs and data to cloud vendors such as Alibaba Cloud and AWS. Users do not need to consider all operational and deployment issues.

Edge computing has become an important technical support for the development of IoT [122]. A thorny problem in edge computing is service migration, especially in the mobile IoT device environment. Due to the limited coverage of a single edge server network, the migration of mobile services between servers is likely to reduce the QoS of the services. State preservation of services (such as stateful services), data loss, and cost control have become challenges in the migration of edge computing services.

Migration at fog is the process of migrating applications, services, or data from traditional centralized data centers or cloud environments to a fog computing environments. The purpose of this migration is to achieve low latency, bandwidth optimization, enhanced security, improved scalability, and fault tolerance. Fog migration involves redesigning applications to adapt to the distributed architecture of fog computing, including modular design and the ability to

handle network dynamics [66, 123]. Fog migration can provide more effective support for IoT devices, mobile devices and other applications that need rapid response, and achieve the goal of intelligent edge computing.

5.10.4 Simulations versus real-world testbed migrations

When discussing container migration, two different testing and validation methods are usually involved: simulations and real-world testbed migrations. Here is a comparison between these two methods:

(i) Simulations: It uses models to replace actual or conceptual systems for training, analysis, argumentation, experimentation, experimentation, and planning methods, techniques, and activities [4]. Simulations can predict system performance and efficiency, validate and iterate modeling and simulation through real experimental data, support, optimize and expand experimental identification, accelerate development and reduce risk costs [31].

(ii) Real world testbed migrations: Its definition is the process of testing and validating a system or application in a real physical environment, involving the migration of the system or application from one environment to another. Since it is conducted in a real-world environment, it can directly evaluate the performance, reliability, and safety of the system or application under actual operating conditions to ensure that it meets practical needs [1].

When conducting container migration testing and validation, simulations and real-world testing platforms are usually combined. Simulation can quickly validate concepts and strategies in the early stages, while real-world testing platforms are used to test and optimize migration strategies under conditions close to actual operational environments. This combination of methods can balance the cost, time, and accuracy of results, providing a comprehensive evaluation for fog migration.

5.11 Container scaling

With the continuous development of cloud computing and container technology, container scaling has become an important means to ensure application performance, high availability, and resource optimization. This section will explore the strategies and practices of container scaling from two key perspectives: firstly, the scaling decisions of proactive and reactive, which exhibit different characteristics and advantages in dealing with load changes; next horizontal vertical and hybrid scaling strategies represent how effectively container resources are in different scenarios.

5.11.1 Proactive versus reactive scaling decisions

The scaling decisions of Proactive and Reactive reflect two different strategies, which have a significant impact on the performance and resource allocation of container applications. The following are specific explanations of these two strategies:

(i) Proactive scaling decision: This method will use historical data of container load to train a specific AI model, through which future changes in container resource load can be perceived and predicted in advance [124]. It allows administrators or systems to automatically adjust resources to maintain optimal performance and efficiency. For example, this strategy can predict based on historical data that as long as it reaches 7 pm or 8 pm, the QPS of AI applications will significantly increase because everyone is off work, which is the entertainment time at night.

(ii) Reactive scaling decision: This is a strategy that utilizes third-party resource monitoring tools, such as Promethues [125], to make real-time decisions on the number of replicas and resource allocation in containers. The container orchestration tool determines whether to expand or dissolve based on the resource change data of the relevant containers in the monitoring tool. When the load increases, reactive scaling will start adding containers; When the load decreases, it will decrease the number of containers [126]. The decision-making of reactive scaling is based on real-time load data. When training the model, the utilization of GPU and GPU memory inside the container may reach 80%-90%. At this time, the system will immediately detect the high utilization rate and scale up the capacity promptly.

5.11.2 Horizontal, vertical and hybrid scaling

Three types of scaling techniques are described below:

(i) Horizontal scaling: It is a way to cope with load changes by increasing or decreasing the number of container instances [127, 128] (such as Pods, container groups, etc.). It can respond very quickly to load changes and adjust overall processing power by adding or removing container instances. Each container instance is independent and has good fault isolation, a fault in one instance will not affect other instances. Horizontal scaling is very suitable for scenarios with stateless services and the need to handle a large number of concurrent requests.

When AI applications need to handle a large number of concurrent requests and each request has a relatively short processing time, horizontal scaling is a good choice. For example, online recommendation systems, real-time advertising delivery systems, etc. In some application scenarios that require a large amount of computing resources (such as CPU, GPU, memory, etc.), horizontal

Feature	Cloud computing	Fog computing	Edge computing
Location	Data Centers	Edge nodes	Device local
Computing power	High	Medium	Low
Delay	High	Medium	Low
Bandwidth requirements	High	Medium	Low
Data processing	Large-scale data processing	Partial data processing	Real-time data processing
Safety	High	Medium	Low
Scalability	High	Medium	Low
Flexibility	Low	Medium	High
Applicable scenarios	Large-scale data analysis and model training	Regional data analysis, partial model training	Real-time response, model inference
Merit	Computing resources are abundant	Available for regional compute resources	Low latency, real-time response, strong privacy protection, high bandwidth utilization
Shortcoming	High latency, poor privacy protection, and high bandwidth requirements	Medium capability, partial privacy protection	Limited computing resources, security issues, and poor scalability

Fig. 8 Advantages, disadvantages and emphases of the three computational paradigms

scaling can provide sufficient resources by adding more machines. For example, DL model training, large-scale image recognition, etc.

(ii) Vertical scaling: It is adjusting the processing power of a single container instance by increasing or decreasing its resource allocation (such as CPU, memory, storage, etc.). It does not require managing multiple container instances, only adjusting the resource allocation of a single instance and can accurately adjust resource allocation based on actual load conditions, avoiding resource waste. Vertical scaling is suitable for stateful services [129]. However, this scaling strategy also has its drawbacks, as it poses a challenge to the computing and storage capabilities of individual machines.

When AI applications encounter performance bottlenecks stemming from the capabilities of individual nodes, vertical scaling offers an effective solution by enhancing hardware capabilities, such as deploying faster CPUs, increasing memory capacity, or leveraging more efficient GPUs. For AI applications that do not necessitate extensive concurrent processing or substantial computing resources, vertical scaling can serve as a more cost-effective and straightforward approach, minimizing complexity while maximizing performance within the confines of a single node.

(iii) Hybrid scaling: Hybrid scaling is a scaling strategy that combines both horizontal scaling and vertical scaling [88]. Based on the load characteristics and requirements of the application, use both horizontal and vertical scaling methods to optimize resource allocation and performance. Being able to flexibly choose scaling methods based on different scenarios and needs, and combining horizontal and vertical methods can more effectively utilize resources, and improve application performance and stability.

For AI applications where demand often changes or is difficult to predict, hybrid expansion can dynamically adjust the ratio of horizontal and vertical expansion based on actual demand.

6 Comparisons of existing Edge AI approaches based on taxonomy

In this section, we compare the existing edge AI approaches based on the proposed taxonomy.

6.1 Infrastructure

Cloud computing, fog computing, and edge computing play different roles in realizing offline, low-latency, privacy-preserving AI services [130]. Among them, cloud computing provides powerful computing and storage resources for training large-scale DL or other algorithm models, and processing massive amounts of data, which are usually used in Edge AI to handle time-insensitive tasks, such as large model training, multi-data analysis and model optimization, and finally, cloud computing distributes well-trained models to various user devices;

Aspect	Static Use Cases	Dynamic Use Cases
Data Processing	Edge AI analyzes data in static devices in real time for quick decision-making	Edge AI processes physiological or traffic data in real time without cloud processing
Device Type	Static sensors (e.g., environmental monitoring sensors, cameras)	Wearable devices (e.g., health trackers), smart traffic systems
Data Mobility	Stable data flow, focused on devices in specific locations	High data mobility, involving constantly changing mobile devices
Privacy Protection	Local data processing helps protect data privacy	Edge processing avoids offloading data to the cloud, enhancing privacy protection
Real-time Capability	Real-time analysis suited for long-term environmental monitoring and decision-making	Real-time analysis supports immediate feedback and dynamic adjustments
Environmental Adaptability	Suited for fixed environment monitoring	Adapts to changing environments

Fig. 9 Comparison of different IoT Use Cases

Fog computing moves computing resources to the edge of the network, reduces data transmission latency, improves response speed, and is typically used in Edge AI to handle tasks that require high real-time responses, such as speech recognition.

Edge computing deploys computing resources directly near terminal devices to further reduce data transmission latency, which is usually used for real-time reasoning and decision-making in Edge AI, such as intelligent monitoring, smart home, etc., edge computing can realize real-time processing of data on user devices, maximize the protection of users' data privacy, and at the same time reduce the dependence on network bandwidth and reduce the pressure on the core network.

In summary, cloud computing, fog computing, and edge computing have their own focus on Edge AI, and these three together build a complete edge intelligence ecosystem. Cloud computing provides powerful computing and

Model	Heuristics	Meta-Heuristics	Machine learning	Deep reinforcement learning
Scenario	Simple problem	More complex issues	Classification; regression	Decision-making ;control
Example	Resource allocation; Path planning	Optimization, scheduling issues	Image, speech recognition; Forecast	Game strategy; robotic arm control; Autonomous driving
Advantage	Simple and efficient; Good explainability	Dealing with more complex issues; Good flexibility	Suitable for a wide range of data types; Real-time processing	Continuous and discrete control; Self-directed learning
Data	No need for a lot of data	No need for a lot of data	A lot of labeled data is required	High-dimensional data; Environment interactions
Shortcoming	Poor performance in dealing with complex problems	Long solution time	Sensitive to outliers	Long training time; The training process is unstable
	It is easy to fall into local optimum	Parameter adjustment is difficult	Requires a lot of computing resources	Requires a lot of computing resources

Fig. 10 Comparison of different models

storage resources, fog computing emphasizes real-time and low latency at the edge of the network, and edge computing enables real-time data processing and decision-making closest to the end device. These three work together to provide comprehensive support for the development of AI at the edge.

As shown in Fig. 8, we compare the emphasis, advantages and disadvantages of cloud computing, fog computing, and edge computing under different indicators:

6.2 Application

Monolithic and microservices have their own advantages and disadvantages in edge AI, and we will compare the advantages and disadvantages of these two in terms of flexibility, performance and resource utilization, as well as deployment and scalability.

(i) Flexibility: Since all functional modules of the monolithic architecture run inside the same application, if we need to modify a module, we may have to recompile and deploy the entire application, so the monolithic architecture is not conducive to modularity and independent development, that is, it is less flexible; Each microservice in the corresponding microservices architecture can be deployed, scaled, and updated independently, making it easy to develop and maintain independently, thus increasing the flexibility of the overall application.

(ii) Performance and resource utilization: Monolithic architectures may have resource contention and performance bottlenecks because all modules share the same process and resources, but from a resource utilization perspective, monolithic architectures may make more efficient use of resources because they do not require additional communication and management overhead; The microservices architecture, on the other hand, can independently deploy and scale the corresponding microservices according to the needs, thereby improving the performance of the application. From the perspective of resource utilization, services in a microservice fabric need to communicate with each other, which may increase the latency and bandwidth consumption of the system, and the microservice system may require more resources to manage and run due to the need to maintain multiple services.

(iii) Deployment and scalability: Monolithic architectures are typically simple and easy to implement and deploy, but they often lack scalability to cope with frequently changing requirements; The corresponding microservices architecture, while more complex to deploy and often requires additional development and management efforts, scales flexibly and allows services to be added or removed quickly as needed.

In summary, the monolithic architecture focuses on simple deployment and performance optimization, which is suitable for simple, relatively fixed edge AI scenarios, while the microservice architecture focuses on flexible scaling and maintainability, and is suitable for complex edge AI application scenarios that need to be dynamically adjusted.

6.3 IoT use cases

IoT use cases can be divided into static and dynamic in terms of user mobility. As shown in Fig. 9 2, we will show the role of edge AI in two different IoT use cases from different perspectives.

6.4 Methods

For the four main AI methods, heuristics [131], meta-heuristics [78], machine learning [132], and deep reinforcement learning [133], we will compare them from the perspectives of applicable scenarios and problem complexity, data scale and training cost, real-time requirements and resource consumption, and generalization. Figure 10 shows the specific comparison.

In summary, choosing the right algorithm depends on the specific edge AI application scenario, data scale, data type, real-time requirements, and resource consumption. Heuristics and meta-heuristics are generally suitable for simple to medium-complexity problems, and the requirements for data and resources are generally not very high. ML and DRL are more suitable for dealing with some complex and nonlinear problems, and have high requirements on data volume, data quality, and computing resources.

6.5 Resource management

With respect to the methods of resource provisioning, resource allocation, application placement, and workload distribution and prediction in edge AI resource management [134], we will further describe the relationship between these methods in detail, as shown in Fig. 11.

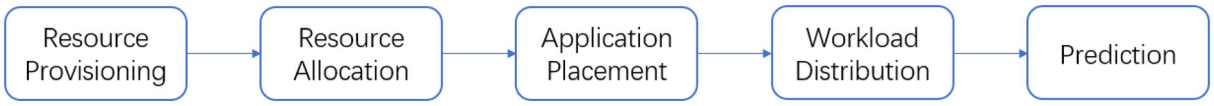


Fig. 11 Process of Resource Management

Table 4 Comparison of different deployment methods

Model	Full model	Reduced model
Model size	Large	Small
Inference speed	Slow	Fast
Accuracy	High	Slightly lower
Training and deployment cost	High	Low
Application scenario	Resource-rich equipment	Resource-constrained devices

Table 5 Comparison of different types of heterogeneity

Heterogeneity	Computing heterogeneity	Hardware heterogeneity	Platform heterogeneity
Definition	Different types of computing tasks	Different kinds of hardware	Devices with different functions
Example	Image recognition; NLP	CPU, GPU, FPGA, ASIC	Cloud server; Edge device
Difference	Differences in demand	Diversification of hardware	Differences between devices

Fig. 12 Comparison of different types of security

Security	Platform security	Host security	Network security
Definition	Focus on the security of the management platform module	The hardware and software of the device itself, the security of applications	Communication security between devices; Secure connections
Concerns	Maintain the management platform, prevent malicious access	Protect individual devices, prevent malicious attacks, and more	Secure data connections against data leaks or connection interruptions
Main measures	Data encryption; access control	Identity verification; access control	Encrypted communications; firewalls

As shown in Fig. 11, when deploying edge AI, we first need to ensure that the edge devices have sufficient computing, storage, and network resources, and once they have sufficient resource provision, we also need to allocate these resources to different applications or computing tasks. Resource allocation ensures that each application or compute task gets the resources it needs to meet its performance requirements. Once the resources are allocated, the application needs to be placed on the appropriate edge device. Application placement needs to take into account the characteristics and needs of each application, as well as the state information of the edge device to achieve the best placement strategy. Once the application is placed, we can send compute tasks and compute workloads to various edge devices, and the workload distribution enables parallel processing tasks and load balancing, thereby improving the performance and efficiency of the system. Finally, we can predict future resource demand and workload changes through ML models, for example, so that we can make adjustments and optimizations in advance.

6.6 ML model sizing

Regarding the deployment of AI models on edge devices, we generally have two deployment methods: reduced model [135, 136] and full model [137], and we compare these two methods from five aspects: model size, inference speed, accuracy, training and deployment cost, and application scenario, as shown in Table 4.

6.7 Heterogeneity

The different types of heterogeneity [138] involved in edge AI deployment mainly include computing heterogeneity, hardware heterogeneity, and platform heterogeneity, as shown in Table 5.

Scheduling	Container Scheduling	Task Scheduling	Pod Scheduling	Service Scheduling
Definition	Manage and schedule containers	Allocating and managing tasks	Scheduling Pods to available nodes	Assigning services to available nodes
Emphasis	Focus on resource utilization, performance, reliability, and scalability of containers	Focus on task efficiency, time and resource constraints	Optimize the allocation of container groups on edge nodes to improve system efficiency	Distribute services to different devices to meet user needs
Concerns	Container lifecycle management	Real-time; reliability	Network topology; load balancing	Reliability; performance; scalability
Tools	Docker Swarm, Kubernetes, Mesos, etc.	Apache Storm, Apache Flink, KubeEdge	Kubernetes, KubeEdge, etc.	Load balancers, service mesh, edge caching, etc.

Fig. 13 Different types of resource scheduling methods

Migration type	Definition	Characteristic
Stateful migration	Migrate applications (including data, state, etc.)	Fast operation recovery; High resource consumption
Stateless migration	Migrate only the code and logic of the application	Slow operation recovery; Low resource consumption
Intra cluster migration	Migrate within the same device or cluster	Fast migration; Prone to resource constraints
Inter cluster migrations	Migrate across nodes or clusters	Scheduling optimization is complex; Abundant resources
Cloud migration	Migrate containers to cloud servers	Take advantage of the abundant resources of cloud servers
Fog migration	Migrate containers to a fog computing environment	Respond quickly to user needs
Edge migration	Migrate containers to edge servers	Sink resources to the edge
Virtual migration	Migration in a simulated environment	Predict performance and efficiency
Real world testbed migration	Real-world migrations	Verification of true performance and feasibility assessment

Fig. 14 Different types of migrations

6.8 Security

Starting from the security of deploying edge devices, we mainly consider platform security, host security, and network security [139]. As shown in Fig. 12, we will compare the roles of the three in detail.

Fig. 15 Different types of container scaling

Container Scaling	Definition	Characteristic
Proactive scaling	Proactively increase or decrease the number of containers	Timely adjustments; High complexity
Reactive scaling	Trigger scaling based on preset thresholds or rules	Simple; Poor performance
Horizontal scaling	Increase or decrease the number of container instances	Respond quickly to load changes
Vertical scaling	Increase or decrease the resources of a single container instance	Suitable for a single task or container
Hybrid scaling	Combine horizontal and vertical scaling	Dynamic adjustment; Improve performance

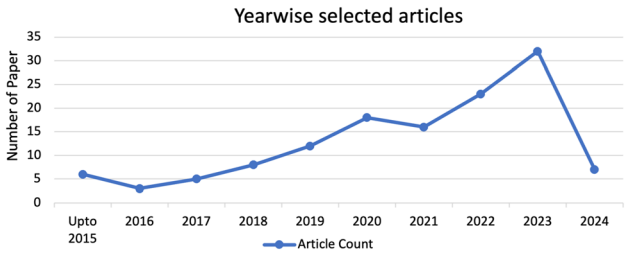


Fig. 16 Year wise Publication

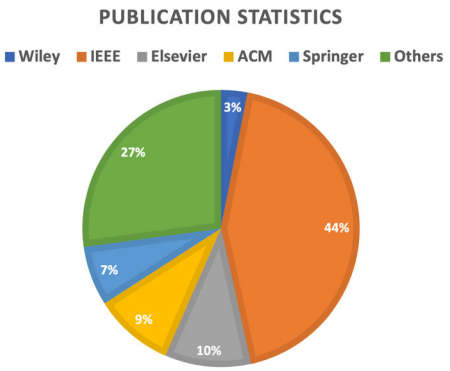


Fig. 17 Publication Statistics

6.9 Scheduling

For the resource scheduling categories in edge AI, there are mainly container scheduling, task scheduling, pod scheduling, and service scheduling [115], and we will compare these four different scheduling types from the aspects of emphasis, scheduling measures, and scheduling tools, as shown in Fig. 13.

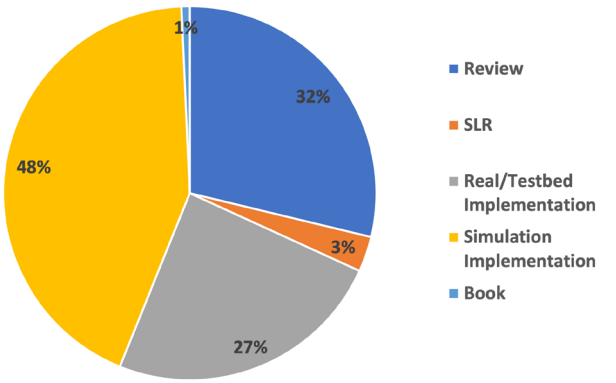


Fig. 18 Categorization of Articles

6.10 Container migration

There are four main types of container migration [140, 141] in edge AI: stateful migration and stateless migration, intra-cluster migration and inter-cluster migration, cloud/fog/edge migration, virtual migration and real-world testbed migration. We have made a detailed comparison of the different migration methods, as shown in Fig. 14.

6.11 Container scaling

Regarding container scaling [129] in edge AI, there are two main ways to actively scale and passively scale from the perspective of system response. From the expansion mode, there are mainly horizontal expansion, vertical expansion and hybrid expansion. As shown in Fig. 15, we make a detailed comparison of these two categories.

7 Analysis and result outcomes

The survey enriches various prospects associated with Edge-empowered AI such as infrastructure support, IoT use cases, resource management strategies, security concerns and many more in the form of various state-of-the-art studies. The authors have systematically reviewed numerous articles in order to understand the prevailing status of Edge AI in distinctive domains along with intelligent paradigms like ML and DL. There are lots of works going in this direction to improve the lifestyle of people and solve real-time problems. Hence, this section signifies the importance of our work referred to in the form of year-wise papers, publication count, type of implementation (Simulation or experimental-based) and nevertheless QoS parameters addressed. Figure 16 presents the year-wise analysis of related work carried out in the form of a number of papers referred from each year. The taxonomy of our study has been proposed with reference to articles from year 2015 to 2024. As depicted in Fig. 16, it is concluded that a major chunk of the referred articles is recent and are from the year 2023. This clearly illustrates the fact that our survey includes the latest work done by the researchers.

Apart from that, we rigorously reviewed the publication-based statistics for the extensive study conducted highlighting its importance in real-time data processing. In total 1253 articles were collected during the data collection phase from various sources such as IEEE, ACM, Wiley, Science Direct, Taylor & Francis and Springer. Afterwards, the filtering stage excluded collected articles based on redundancy and inclusion and Exclusion criteria. The final stage comprises articles that the authors believe contributed the most towards shaping up the survey as depicted in Fig. 17. Furthermore, the articles have been thoroughly reviewed and divided into 4 categories: review, Systematic Literature Review (SLR), implementation (simulation-based), implementation (Real/Testbed-based) and book. Figure 18 illustrates that the major portion of the articles referred, based on implementation (simulation-based), which signifies the fact that the implication of edge-empowered AI is yet to be tested on real-life IoT-based use cases. Several real time works have been proposed by the researchers in recent years to improve the IoT applications-based architecture using intelligent paradigms like ML, DL and reinforcement learning. This highlights a potential research direction for future studies to explore and validate edge-empowered AI in practical, real-world IoT environments. In addition, this article will motivate the researchers to propose novel solutions to improve society 5.0.

8 Future research directions

Edge AI is continuously evolving and showing potential across various domains, offering numerous opportunities for innovation and improvement. This section examines critical future research directions that promise to enhance the capabilities and applications of Edge AI as summarized in Fig. 19. These directions include optimizing energy use,

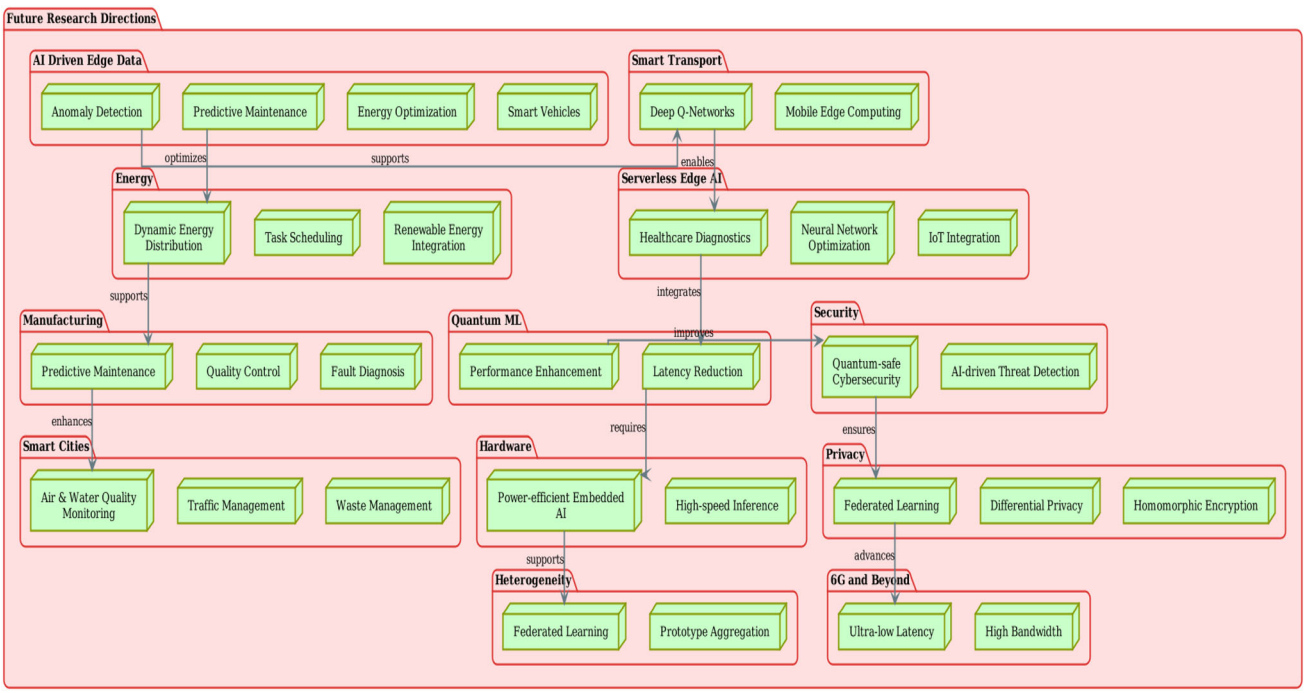


Fig. 19 Summary of Future Research Directions

strengthening security, and integrating with next-generation networks like 6 G, highlighting the transformative impact of Edge AI across multiple sectors.

8.1 AI-driven Edge data

AI-equipped edge devices process data locally, enabling real-time decision-making without the latency caused by cloud transmission. This is essential for latency-sensitive applications such as predictive maintenance, real-time video analytics, and autonomous systems. In predictive maintenance, for instance, AI models deployed at the edge process sensor data from machines to predict failures and schedule repairs before breakdowns occur. These systems often rely on time-series forecasting models, such as Long Short-Term Memory (LSTM) networks, which are optimized for real-time edge inference through techniques like pruning and quantization [142].

Using video, audio, and sensor data for anomaly detection, edge AI enhances security in smart cities by identifying threats in real-time [143]. Anomaly detection often employs CNNs or RNNs to detect irregularities in sensor streams. CNNs, such as MobileNet, use depthwise separable convolutions to reduce the number of operations while preserving spatial hierarchies, making them suitable for resource-constrained edge devices. To further optimize performance, techniques like model compression and pruning are used to reduce the model size without significantly compromising accuracy [142]. RNN-based architectures, including Gated Recurrent Units (GRUs), are also deployed at the edge for detecting anomalies in sequential data, such as traffic patterns or environmental sensors.

Reinforcement learning algorithms are widely used to manage energy consumption at the edge, particularly for dynamic resource allocation. In HVAC systems, RL models learn optimal policies to balance energy consumption and occupant comfort by interacting with the environment and receiving feedback in the form of rewards [144]. In these systems, deep Q-learning (DQN) is used to handle large state-action spaces efficiently. Edge devices also deploy on-policy methods like Proximal Policy Optimization (PPO), which allow continuous adjustments based on real-time data. These models are further optimized through techniques like experience replay, which reduces memory usage and computation load, critical for edge deployments.

Edge AI plays a pivotal role in smart vehicles and drones, where real-time sensor data processing is crucial for navigation and decision-making [145]. Autonomous vehicles rely on edge AI to process data from LIDAR, RADAR, and cameras using sensor fusion techniques, which combine multiple sensor inputs to improve accuracy and robustness [146]. For instance, Kalman filtering is employed to integrate noisy sensor measurements, while CNNs perform object detection and classification. Edge AI uses these techniques to make split-second decisions, such as obstacle avoidance, without cloud dependencies. Privacy-preserving models like federated learning are also crucial in this context, enabling local data processing on vehicles while sharing only model updates, ensuring that sensitive location data remains private. Advanced techniques such as differential privacy and homomorphic encryption are integrated into FL to protect against data leakage during model updates.

However, deploying AI at the edge presents significant challenges, particularly in terms of computational limitations and energy constraints. Techniques like model quantization, where neural network weights are reduced from 32-bit floating point to 8-bit integers, help decrease the model size and improve inference speed [142]. Moreover, hardware-specific optimizations, such as leveraging the parallelism of Tensor Processing Units (TPUs) or Graphics Processing Units (GPUs), play a crucial role. For example, Google's Edge TPU accelerates inferencing tasks with high energy efficiency, while Nvidia's Jetson platform provides scalable computing power for more complex tasks.

In edge computing, the communication paradigm plays a critical role in system performance, especially in distributed learning models like Federated Learning [145]. FL enables devices to compute local updates on their own datasets and only transmit model gradients, thereby reducing communication overhead. Challenges such as non-IID (non-independent, identically distributed) data among edge nodes, which can lead to model biases, are addressed through methods like FedProx, which adds a regularization term to prevent drastic divergence from the global model. Additionally, communication-efficient techniques such as gradient sparsification, where only significant gradients are transmitted, and asynchronous updates ensure that edge nodes can update models independently without waiting for synchronization, thus improving efficiency in bandwidth-constrained environments Figure 19.

The convergence of edge AI and advanced hardware solutions continues to drive innovations in AI deployments. Custom AI accelerators, such as Google's Edge TPU, are optimized for low-power inference tasks, supporting applications that require high-speed processing, such as object detection in real-time video feeds [146]. NVIDIA's Jetson platform, on the other hand, provides a scalable solution for more compute-intensive tasks, such as deep learning-based robotics or autonomous navigation. These platforms support parallelized inference operations, maximizing throughput while minimizing energy consumption, making them ideal for edge environments. Future research

directions include developing neuromorphic hardware, which mimics biological neural networks, offering significant reductions in energy consumption while maintaining high processing speeds.

8.2 Energy

Optimizing energy use in AIoT systems through intelligent edge computing requires focusing on sophisticated algorithms for dynamic energy distribution, task scheduling, ML for workload management, and the design of low-power hardware [147]. For example, dynamic energy distribution can be handled using RL-based approaches where the system continuously learns and adapts to energy usage patterns. Techniques such as DQN and Multi-agent Reinforcement Learning (MARL) are effective for decentralized energy optimization, allowing edge devices to work autonomously while coordinating with other nodes in the network to maximize efficiency. MARL models allow each device to act as an independent agent, optimizing energy at both local and global levels by sharing information across the network [148]. Furthermore, task scheduling in AIoT systems can benefit from heuristic algorithms like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), which provide near-optimal solutions for task allocation in energy-constrained environments. These methods are computationally efficient, making them well-suited for edge devices with limited resources [149, 150].

Improvements in communication protocols and integration of renewable energy sources will further enhance system efficiency and scalability. Protocols such as Low-Power Wide Area Networks (LPWANs), including LoRaWAN and NB-IoT, are particularly useful for AIoT systems as they allow for long-range communication with minimal power consumption. These protocols, when combined with AI models running at the edge, enable real-time data exchange between devices while conserving energy. Additionally, optimizing the scheduling of data transmissions based on energy availability or demand-response signals can significantly reduce communication overhead in energy-constrained environments [151].

Literature [152, 153] reported that AI techniques for managing renewable energy sources have been investigated, emphasizing advanced ML models for accurate forecasting and optimizing energy storage and grid integration. For instance, LSTM networks and Autoregressive Integrated Moving Average (ARIMA) models have been widely applied in energy forecasting tasks. LSTMs are especially effective in capturing the temporal dependencies in renewable energy generation data (e.g., solar and wind), allowing for more accurate predictions of energy availability. These forecasts can then be used to optimize the allocation of tasks across the grid, improving the balance between energy demand and supply. In energy storage, AI models like Gradient Boosting Machines (GBM) and Support Vector Machines (SVM) have been applied to optimize the charge–discharge cycles of batteries, maximizing the longevity and efficiency of storage systems [154]. Integrating these AI models with grid control systems enables real-time decision-making to manage fluctuations in energy generation and consumption effectively.

Implementing edge AI will reduce latency and enable real-time decision-making, increasing system responsiveness and resilience [155]. In edge computing environments, AI models can be used to make decisions locally without needing to transmit large amounts of data to a central cloud, which reduces the overall latency and bandwidth usage. This is especially beneficial in scenarios where immediate action is required, such as energy demand-response events or real-time fault detection in smart grids. Edge AI, combined with lightweight models such as MobileNet or TinyML, can process sensor data in real time, detecting anomalies or optimizing energy usage without draining significant computational resources. This local processing not only reduces decision latency but also enhances the system's resilience to network disruptions.

It has been identified that incorporating edge AI in the Internet of Energy (IoE) presents unique opportunities, particularly in areas like secure edge computing, blockchain for data security, lightweight AI algorithms, standardization for interoperability, and 5 G networks for low-latency communication [32, 79, 156]. Blockchain technology offers decentralized solutions for secure energy trading and data management within IoE networks. The integration of blockchain with edge AI can ensure the transparency and immutability of energy transaction data while minimizing the energy overhead typically associated with blockchain's consensus mechanisms. By using energy-efficient consensus protocols like Proof of Stake (PoS) or Proof of Authority (PoA), IoE systems can maintain security without incurring significant computational costs. Additionally, 5 G and the upcoming 6 G networks offer ultra-low latency and high-speed connectivity, enabling real-time energy optimization by allowing AI models to dynamically offload tasks between the edge and cloud based on energy constraints. Network slicing within 5 G/6 G networks ensures that energy-critical tasks receive prioritized resources, improving the overall performance and responsiveness of the IoE system.

Advances in federated edge AI and DRL will optimize energy distribution, enhancing the efficiency and resilience of IoE systems [157]. Federated Learning allows edge devices to collaboratively train a global model without exchanging raw data, thereby preserving privacy and reducing communication overhead. In energy management, FL can be adapted to enable energy-aware federated

updates, where devices with limited energy resources reduce their participation in the model training process [154]. This approach ensures that the overall system remains efficient, even when certain devices operate under low-power conditions. DRL further improves energy optimization by learning policies for real-time control of energy resources, such as dynamically adjusting the operation of energy storage systems or scheduling energy-consuming tasks during periods of high renewable energy generation. These algorithms, when combined with edge AI, enable IoE systems to respond to changes in energy supply and demand with minimal delay.

8.3 Manufacturing

AI meets rising customer expectations for customization and high-value production by integrating capabilities at the network edges. The integration of edge AI in manufacturing introduces opportunities for decentralized, low-latency processing, which is crucial for real-time operations in smart factories. Edge AI enables data-driven decision-making directly on the factory floor, minimizing delays due to communication with centralized cloud servers [158]. This shift towards distributed intelligence allows manufacturers to rapidly adapt to changing production demands, offering more customized products and services. Furthermore, AI-driven automation helps reduce manual labor, improving both productivity and safety in industrial environments.

It has been identified that integrating AI in manufacturing enhances collaboration with experts through tools like Google VisionAI for data science applications [159]. Vision-based AI tools, such as Google VisionAI, enable real-time monitoring and analysis of production lines by processing images and video streams at the edge. These tools can be integrated with machine vision systems to automatically inspect product quality, detect defects, and identify anomalies in manufacturing processes. By deploying AI models at the edge, manufacturers benefit from low-latency responses, which are critical for high-speed production environments where defects need to be detected and corrected in real time. This significantly reduces the waste associated with defective products, leading to increased production efficiency and cost savings [160].

Key research areas include manufacturing scheduling and planning due to abundant data and productivity improvement opportunities. In the context of scheduling and planning, AI can optimize the allocation of resources and machinery to minimize downtime and maximize throughput. Traditional scheduling algorithms, such as job shop scheduling, have limitations in handling the complexity and variability of modern manufacturing environments. AI techniques, such as RL and constraint satisfaction algorithms, are being applied to dynamically adjust production schedules in response to real-time data [161]. For example, RL can be used to model complex environments with multiple variables, learning optimal policies to allocate resources efficiently while adapting to unforeseen disruptions, such as machine failures or supply chain delays. Additionally, advanced optimization algorithms like GA and PSO can be integrated into manufacturing planning systems to solve multi-objective optimization problems, balancing factors like energy consumption, production speed, and quality [162].

There is a need to investigate edge AI's real-time analysis for predictive maintenance, quality control, and fault diagnosis in manufacturing, improving efficiency, reducing waste, and optimizing resources [163]. Predictive maintenance is a key area where edge AI can significantly enhance manufacturing operations. Using AI models like RNNs and LSTM networks, edge devices can analyze data from sensors embedded in machinery to predict when a machine is likely to fail. This allows manufacturers to perform maintenance only when necessary, reducing downtime and extending the life of equipment. In addition, real-time fault detection models deployed at the edge can immediately identify deviations from normal operating conditions, allowing for immediate corrective actions. Techniques such as CNNs can be employed for real-time image and video analysis in quality control, detecting surface defects, dimensional inaccuracies, and assembly errors with high accuracy [164]. Implementing AI for fault diagnosis can also leverage unsupervised learning methods, such as autoencoders and clustering algorithms, to identify abnormal patterns in sensor data without needing labeled fault data.

Implementing ML models at the edge allows continuous monitoring, early fault detection, and immediate corrective actions, enhancing intelligent manufacturing. The real-time processing capabilities of edge AI offer a significant advantage for continuous monitoring in manufacturing environments. By running ML models at the edge, data from production lines can be processed in real time, enabling early detection of equipment malfunctions and process anomalies. For example, edge devices can run anomaly detection algorithms using principal component analysis (PCA) or one-class SVMs to flag deviations from normal production patterns. This early detection helps prevent costly downtime and reduces the risk of producing defective products, enhancing overall manufacturing efficiency [165].

Additionally, edge devices can optimize power generation and consumption by analyzing real-time data, promoting renewable energy use and cost savings. Energy efficiency is becoming a critical aspect of modern

manufacturing systems, especially as manufacturers strive to reduce their carbon footprint. Edge AI can be integrated with energy management systems to monitor power usage in real time, allowing for dynamic optimization of energy consumption across various production processes. Machine learning algorithms, such as reinforcement learning and Bayesian optimization, can be employed to balance power consumption with production goals. For example, RL-based systems can learn optimal policies for turning machines on or off based on current energy prices, renewable energy availability, and production schedules. This integration of edge AI with energy optimization not only reduces operational costs but also aligns with sustainable manufacturing practices [166].

8.4 Smart cities

The future of Edge AI in smart cities holds the promise of several significant advancements [167]. Edge AI enables real-time data processing, allowing cities to respond to dynamic situations instantly without relying on cloud processing. By deploying AI models at the edge, data from sensors (e.g., air quality monitors, traffic cameras, and IoT waste bins) can be analyzed locally, reducing the need for constant data transmission to centralized servers. This not only lowers the load on cloud infrastructure but also enhances the responsiveness of city services. For example, in traffic management, edge AI can detect congestion or accidents in real time using computer vision models such as CNNs applied to video feeds. These systems can dynamically adjust traffic signals or reroute vehicles based on real-time conditions, reducing congestion and improving urban mobility. [165]

First, enhancing data processing capabilities at the edge reduces cloud load and latency, enabling real-time decision-making for applications like air and water quality monitoring, traffic management, and waste management. In air and water quality monitoring, edge AI can analyze sensor data from distributed nodes throughout a city to detect harmful pollutants or water contamination. Machine learning models such as random forests, SVMs, or deep learning models like LSTMs can be employed at the edge to predict air quality trends based on historical data and real-time inputs. These models can continuously adjust ventilation systems in buildings or trigger alarms in high-risk areas, ensuring a more responsive and automated environmental management system. Similarly, edge devices in smart waste management systems can monitor bin levels using IoT sensors, and AI models can optimize waste collection routes by predicting when and where waste accumulation is likely to occur, thus reducing operational costs and environmental impact [168].

Integration with AI algorithms will facilitate more intelligent decision-making by analyzing sensor data from various city domains. For example, AI-driven edge systems can aggregate and process data from traffic lights, parking sensors, public transport, and emergency services to optimize city-wide mobility. RL models can be used to manage traffic lights dynamically, learning from past traffic flow patterns to minimize delays and congestion. [169] RL algorithms such as DQN or PPO can be applied in such environments, where the model learns to optimize traffic signals based on real-time sensor data, improving traffic flow efficiency over time. Additionally, AI-based predictive models like LSTMs can forecast urban energy demand by analyzing sensor data from smart meters across the city. This allows utility companies to balance energy generation and distribution in real time, reducing the risk of blackouts and improving energy sustainability [170].

Innovations in low-power hardware and efficient communication protocols are crucial to ensure scalability and energy efficiency. Edge AI systems in smart cities must be designed with energy efficiency in mind, particularly given the large number of distributed edge devices required. Specialized hardware accelerators, such as Google's Edge TPU or Nvidia's Jetson Nano [171], are designed to perform AI inference with minimal energy consumption, making them ideal for edge deployments in smart cities. These hardware solutions are typically paired with lightweight AI models like MobileNet or TinyML to further reduce power consumption while maintaining high accuracy. Additionally, communication protocols such as **LoRaWAN and Narrowband IoT (NB-IoT)** [172] are optimized for low-power, wide-area networks, ensuring that edge devices can communicate over long distances without significant energy overhead. Integrating these protocols with AI-driven edge systems ensures that smart cities can scale without overwhelming energy resources.

Developing secure edge computing and blockchain technologies will also address data security and privacy concerns [109]. One of the key challenges in deploying AI at the edge is ensuring data security and privacy. Edge devices often handle sensitive information, such as personal location data or surveillance footage, making them attractive targets for cyberattacks. To address this, blockchain technology can be integrated with edge AI to create a decentralized and secure method of data management [173]. Smart contracts can automate the verification and exchange of data between devices, ensuring that only authorized entities can access or modify the data. Furthermore, blockchain-based systems can offer tamper-proof audit trails, ensuring transparency and accountability in data usage. Privacy-preserving AI techniques, such as federated learning and differential privacy [174], can also be implemented at the edge to allow AI models to be

trained on decentralized data without exposing sensitive information. Federated learning allows edge devices to collaboratively train a global model without sharing raw data, while differential privacy ensures that individual data points cannot be reverse-engineered from model outputs.

Finally, advancements in 5 G networks will provide the necessary infrastructure for high-speed, low-latency communication, further enhancing the responsiveness and resilience of intelligent city systems. The deployment of 5 G technology in smart cities enables real-time, high-bandwidth communication between millions of connected devices. 5 G's ultra-reliable low-latency communication (URLLC) [175] can significantly enhance applications that require instantaneous responses, such as autonomous vehicles, real-time traffic management, and emergency response systems. Additionally, network slicing [176] within 5 G allows different services to have dedicated virtual networks with tailored resources, ensuring that critical services like emergency response or traffic control always receive the necessary bandwidth and priority. As smart cities evolve, the combination of edge AI and 5 G will be crucial for enabling real-time decision-making, where AI models deployed at the edge can interact with central cloud systems when needed, without suffering from communication delays. Future advancements in 6 G networks [177] will likely take these capabilities further by offering even higher data transfer speeds and supporting more complex AI models in real-time applications.

8.5 Smart transport

Applying DRL, specifically DQN, to mobile edge computing in smart transportation systems plays a crucial role in balancing computing capability and traffic state. DQN, as a value-based reinforcement learning algorithm, operates by learning a Q-function that maps states (such as traffic conditions) to the expected rewards of taking certain actions (like adjusting traffic lights or rerouting vehicles). In smart transport, DQN-based models [178] are deployed at the edge to optimize local decision-making in real time, minimizing the delays caused by communication with centralized cloud servers. For instance, in an urban environment, a DQN-based system can learn to adjust traffic signals based on current traffic flow, historical patterns, and predicted congestion, thus improving traffic efficiency [179] and reducing fuel consumption. The use of edge computing here ensures low-latency decision-making, which is critical in dynamically evolving traffic situations where even minor delays can significantly impact traffic flow.

This approach highlights the need for further research on trade-offs and optimization techniques to enhance efficiency and performance in edge AI applications [180]. In the context of smart transport, there are several trade-offs to consider, particularly in the computational complexity of DRL models versus the energy and processing limitations of edge devices. DRL models like DQN, while effective, can be computationally intensive due to the large state-action space that must be explored. To address this, techniques such as **Double DQN** and **Dueling DQN** [181] have been introduced to improve the stability and efficiency of Q-learning by reducing overestimation biases and learning more granular value functions. These variants reduce the number of updates required for the Q-function to converge, which is critical in resource-constrained edge environments.

In addition to improving algorithmic efficiency, optimizing resource allocation in mobile edge computing environments is an ongoing area of research. Traffic state optimization is a multi-objective problem that involves balancing computational load, energy consumption, and communication latency. Techniques such as **multi-agent DRL (MADRL)** [182] can be applied, where each vehicle or edge device is treated as an independent agent learning to optimize its local performance while contributing to the global traffic management system. MADRL [183] allows for decentralized decision-making, where agents can communicate with each other or a central node to share state information (such as traffic density or road conditions), thus improving the coordination of traffic signals and vehicle routing across a city. The challenge lies in managing the communication overhead, which increases with the number of agents, while maintaining real-time performance.

Further research is also needed to explore hierarchical reinforcement learning (HRL) [184], which can decompose complex traffic control tasks into a hierarchy of simpler sub-tasks. HRL enables smart transport systems to break down large-scale optimization problems (e.g., optimizing city-wide traffic flow) into smaller, manageable sub-problems (e.g., optimizing traffic flow at individual intersections). This reduces computational overhead and makes the learning process more scalable, particularly when applied at the edge. Additionally, **PPO** [185], a popular policy-based DRL algorithm, could be explored for continuous control in smart transport applications. PPO is known for its robustness in high-dimensional environments and could improve the real-time adaptability of transport systems to unpredictable traffic conditions or sudden changes in road infrastructure.

To further optimize the integration of DRL in smart transportation, future research should focus on energy-efficient hardware accelerators such as Google's Edge TPU or Nvidia's Jetson [186], which are designed to handle AI workloads with minimal power consumption. These devices can run DRL models locally, allowing edge nodes (e.g.,

traffic lights or smart vehicles) to process large volumes of data in real time without overburdening the power infrastructure. Furthermore, low-power communication protocols, such as **Vehicular Ad-hoc Networks (VANETs)** [187], can be integrated with edge AI systems to ensure efficient data sharing among vehicles and infrastructure while minimizing energy usage. VANETs allow vehicles to communicate with roadside units (RSUs) and other vehicles in real time, enabling cooperative decision-making for optimized traffic control.

Finally, with the advancement of 5 G networks, mobile edge computing in smart transportation will benefit from ultra-low-latency communication, enabling more efficient interaction between vehicles, infrastructure, and edge devices. The use of **network slicing** in 5 G networks can provide dedicated virtual network resources for traffic management, ensuring that time-sensitive tasks, such as emergency vehicle routing or accident detection, are handled with priority. This integration of 5 G [176] with edge AI will allow transport systems to scale effectively, handling large volumes of data with minimal delay while maintaining energy efficiency. Future research may also explore the potential of **6 G networks**, which are expected to provide even faster data transfer speeds and greater network capacity, allowing for the deployment of more sophisticated AI models at the edge in smart transport systems [177].

Applying DRL to mobile edge computing in smart transport offers numerous opportunities for optimizing traffic management and improving transportation efficiency. However, ongoing research is required to address the trade-offs between model complexity, computational resources, and energy consumption in edge devices, while leveraging the latest advancements in communication protocols and network technologies.

8.6 Serverless Edge AI

Leveraging the flexibility and scalability of serverless architectures will significantly enhance the deployment of ML models in healthcare [120, 188]. Serverless computing offers a key advantage by abstracting the underlying infrastructure, allowing developers to focus solely on deploying and scaling machine learning (ML) models without needing to manage servers. This architecture also allows for automatic scaling based on demand, meaning ML models can be deployed in a cost-efficient manner. Specifically in healthcare, real-time diagnostics and monitoring are vital for patient care, and serverless edge AI [189] enables low-latency data processing at the edge without needing continuous cloud connectivity. For instance, ML models can be used to process patient data directly from IoT devices such as wearable sensors, providing immediate alerts for abnormal health conditions like arrhythmias or glucose fluctuations.

This approach will enable real-time, cost-effective diagnostics without managing backend infrastructure. One of the key challenges in serverless environments, however, is managing the "cold start" problem [190], where there is a delay in invoking a function due to the time it takes to spin up resources when a function is first triggered. In healthcare, where every second matters, reducing cold start latency is critical. Future research could explore techniques such as **pre-warming** serverless functions by maintaining a pool of ready-to-go containers or employing **just-in-time compilation (JIT)** to reduce function invocation time. Additionally, lightweight neural network architectures such as **TinyML** or **MobileNet** can be optimized for serverless environments, minimizing the computation load while maintaining high accuracy in diagnostics tasks.

Future research will optimize neural network models for serverless environments, reduce cold start latencies, and enhance model performance through adaptive learning techniques [191]. Adaptive learning techniques can further improve the efficiency of ML models deployed in serverless architectures. For instance, **on-demand model loading** could allow serverless functions to dynamically load specific portions of a neural network based on the current task, thus reducing memory and processing requirements. This concept, known as **model partitioning**, divides a model into smaller, callable sections, enabling efficient utilization of serverless resources. Furthermore, the use of **quantization** and **model pruning** techniques can reduce the model size, ensuring faster execution times and lower computational overhead, which is crucial for real-time diagnostics in healthcare settings. These optimizations not only reduce latency but also lower the costs associated with serverless computing by minimizing the resources consumed during function execution.

Integrating serverless edge AI with IoT frameworks will facilitate continuous monitoring and rapid response in medical applications [192]. In healthcare, IoT devices such as smartwatches, biosensors, and connected medical devices generate a constant stream of health data that requires real-time analysis. By integrating **serverless edge AI** with these IoT frameworks, the data can be processed locally on the device or at nearby edge servers, reducing latency and ensuring real-time feedback to both patients and healthcare providers. This continuous monitoring can be critical in managing chronic conditions like diabetes or cardiovascular disease, where immediate response to changes in vital signs is essential. In the case of **federated learning**, ML models can be collaboratively trained across multiple edge devices without sharing sensitive health data, further enhancing patient privacy and complying with healthcare regulations like HIPAA, GDPR etc.

Additionally, improving interoperability and security protocols will ensure the safe and efficient handling of sensitive healthcare data in serverless architectures, paving the way for the next generation of intelligent, responsive, and secure healthcare systems. One of the major challenges in serverless edge AI for healthcare is ensuring the security and privacy of sensitive patient data. The distributed nature of serverless architectures, combined with the use of IoT devices, increases the attack surface for cyber threats. Future research should focus on developing secure execution environments such as **trusted execution environments (TEEs)**, which provide hardware-based security features that isolate sensitive data during function execution. TEEs can be integrated with **differential privacy** and **homomorphic encryption** [193] to allow secure data processing and model training without exposing raw data. Moreover, ensuring **interoperability** across different healthcare systems and IoT devices is crucial to enable seamless data sharing and processing. **Standardization efforts** in APIs, data formats (such as HL7 or FHIR for healthcare data) [194], and communication protocols will be vital to ensuring compatibility across platforms, reducing the complexity of integrating serverless edge AI into existing healthcare infrastructure. These advancements will make it possible to deploy scalable, secure, and efficient healthcare systems that leverage the full potential of serverless computing.

8.7 Quantum machine learning (QML)

Integrating Quantum Machine Learning (QML) with edge AI technologies, such as large, intelligent surfaces and visible light communications, will significantly reduce latency and enhance performance [195]. QML leverages the principles of quantum computing, such as quantum superposition and entanglement [196], to process data more efficiently than classical computing methods. By integrating QML with edge AI, it becomes possible to execute complex machine learning algorithms in parallel, drastically reducing computation time. For example, quantum algorithms such as the **Quantum Approximate Optimization Algorithm (QAOA)** [197] and **Variational Quantum Eigensolver (VQE)** [198] can solve optimization problems faster than their classical counterparts, enabling real-time decision-making in edge AI systems. These quantum algorithms are particularly well-suited for tasks like traffic flow optimization, supply chain management, and resource allocation in smart cities, where traditional algorithms struggle with computational complexity.

This combination enables efficient processing and decision-making at the network edge, which is crucial for managing the vast data generated by IoT devices and other edge sources [199]. As the number of connected IoT devices grows exponentially, classical edge AI systems face significant challenges in processing large volumes of data in real time. By incorporating quantum-enhanced models, edge AI can offload more complex computations to quantum processors, allowing for faster and more efficient data analysis. For instance, **Quantum SVMs (QSVMs)** [200] can be used to classify high-dimensional data generated by IoT sensors more efficiently than classical SVMs, leading to faster and more accurate decision-making at the edge. Additionally, **Quantum Neural Networks (QNNs)** [201] have the potential to significantly reduce the training time for deep learning models, allowing edge AI systems to adapt more quickly to changing environments. This is especially important in autonomous systems, such as self-driving cars or drones, where real-time responsiveness is critical.

Leveraging quantum computing capabilities in edge AI applications will achieve unprecedented network performance, leading to more responsive and adaptive AI systems [17]. Quantum edge AI systems can utilize **quantum parallelism** to explore multiple solutions simultaneously, which significantly enhances the performance of optimization and search tasks. For example, in a smart transportation system, quantum-enhanced algorithms could simultaneously evaluate multiple traffic routes to find the most efficient path, significantly reducing computation time compared to classical algorithms. Furthermore, **quantum annealing** can be used to solve combinatorial optimization problems in real time, which is highly beneficial for applications like dynamic resource allocation in 5 G networks.

The convergence of QML and edge AI in 6 G networks will drive innovative solutions for real-time analytics and intelligent automation, meeting the increasing demands for low-latency and high-efficiency edge computing environments [202]. 6 G networks are expected to provide ultra-reliable low-latency communication (URLLC), which is essential for supporting the high-speed data exchange required by quantum-enhanced edge AI systems. By integrating QML with 6 G networks, edge devices can harness the power of **quantum communication protocols**, such as **quantum key distribution (QKD)**, to ensure secure data transmission between devices and the cloud. QKD offers provably secure communication, which is particularly important for applications like autonomous vehicles and smart grid management, where data integrity is critical. Additionally, the high bandwidth offered by 6 G will enable edge devices to offload quantum computations to nearby quantum processors with minimal delay, further enhancing the system's overall responsiveness [17].

Another area of research is the use of **quantum machine learning for federated learning (QFL)** in edge AI environments. In traditional federated learning, edge

devices collaboratively train a global model without sharing their local data, which minimizes privacy risks. However, with the increasing complexity of AI models, the communication overhead in FL can become a bottleneck. By using QFL, edge devices can leverage quantum algorithms to reduce the amount of data that needs to be shared, as quantum communication enables the transmission of compressed information with higher efficiency. This can lead to faster convergence times in federated learning models while maintaining the privacy of sensitive data, which is crucial in sectors like healthcare and finance.

Moreover, future research should focus on developing **quantum-classical hybrid algorithms** [203] that can efficiently combine the strengths of both quantum and classical computing. In scenarios where quantum hardware is limited, hybrid approaches can offload specific sub-tasks (such as optimization or matrix inversion) to quantum processors while performing other parts of the computation on classical hardware. These hybrid systems are particularly well-suited for edge AI, where devices may not have direct access to quantum hardware but can offload tasks to quantum cloud services. This will enable a more scalable and flexible approach to integrating QML in edge AI applications.

As 6 G networks and quantum hardware continue to evolve, the convergence of these technologies will drive new innovations in areas like intelligent automation, secure communications, and resource optimization across smart cities and autonomous systems. The combination of quantum computing and AI at the edge will address critical challenges in real-time processing, allowing for the handling of vast data streams with minimal latency and maximizing the use of network and computational resources [17].

8.8 Hardware

The physical boundaries for AI systems are set by the hardware of edge nodes, which drives significant efforts in designing specialized edge AI hardware. Edge AI hardware must balance several constraints, including computational performance, power efficiency, size, and cost. This has led to the development of specialized hardware like **Nvidia's Jetson TX2**, which is designed for power-efficient embedded AI computing, and **Google's Edge TPU**, optimized for high-speed inference at the edge [20]. The **Jetson TX2** integrates GPU-based architectures with AI acceleration cores to handle tasks requiring significant parallel processing power, such as deep learning model inference. It is particularly well-suited for applications like autonomous robots, drones, and intelligent surveillance systems, where real-time processing of sensor data is critical. On the other hand, **Google's Edge TPU** focuses on accelerating machine learning models in a cost-effective and energy-efficient manner, often used in IoT devices, smart cameras, and wearable technology for lightweight, high-speed AI tasks. These hardware platforms allow AI models to run at the edge with reduced latency and power consumption compared to traditional cloud-based AI systems.

However, these devices mainly concentrate on handling entire tasks, especially local edge inference. Moving forward, edge AI hardware design will evolve to address the diverse requirements of various AI workloads. For example, high-performance tasks such as 3D object recognition, complex signal processing, and multi-modal data fusion require more powerful hardware accelerators that can manage large-scale computations at the edge. This is where **Field Programmable Gate Arrays (FPGAs)** [204] and **Application-Specific Integrated Circuits (ASICs)** come into play. **FPGAs** offer reconfigurable hardware that can be customized for specific AI tasks, such as accelerating CNNs or RNNs. The flexibility of FPGAs makes them ideal for environments where the AI workload can change dynamically, such as in autonomous vehicles or industrial automation. **ASICs** [205], on the other hand, provide dedicated hardware that is optimized for specific AI algorithms, offering superior performance and energy efficiency for large-scale deployment in fixed-function systems like smart grids and edge data centers.

To further enhance energy efficiency, **neuromorphic processors** [206] are being explored as a promising solution. These processors mimic the structure and operation of the human brain, using spiking neural networks (SNNs) that only consume power when active. This event-driven computation model is highly advantageous for edge applications such as continuous sensor monitoring, where devices need to remain energy-efficient while processing intermittent data streams. For example, neuromorphic chips can be integrated into wearable devices for health monitoring or in environmental sensors for smart city applications, reducing overall power consumption without compromising performance. Research in neuromorphic computing is focusing on increasing the scalability and accuracy of these processors for more complex AI tasks at the edge.

Moving forward, we'll see a variety of edge AI hardware designed specifically for different AI system architectures and applications [145]. Future advancements in edge AI hardware will likely focus on co-designing hardware and software to optimize the performance of AI models. This includes developing **domain-specific architectures (DSAs)** that are tailored for specific AI workloads, such as natural language processing (NLP), computer vision, and reinforcement learning. DSAs will allow hardware to process AI algorithms more efficiently by

exploiting the characteristics of the models they are running, such as sparsity in neural networks or the locality of reference in data. Additionally, **heterogeneous computing architectures**, which combine different types of processing units (e.g., CPUs, GPUs, TPUs, and FPGAs) in a single system, will become more prevalent in edge AI deployments. These architectures enable systems to allocate tasks to the most appropriate processing unit based on the computational and energy requirements of each AI model, optimizing overall system performance.

As the demand for edge AI grows, there will also be a focus on improving hardware for secure AI processing. This includes developing **trusted execution environments (TEEs)** that protect sensitive data and AI model integrity in edge devices. TEEs create isolated environments where AI computations can be executed securely, preventing unauthorized access or tampering with data. TEE [207] will be particularly important in industries such as healthcare, finance, and autonomous driving, where security and privacy are paramount. Hardware support for **privacy-preserving AI**, such as homomorphic encryption and secure multi-party computation, will also be integrated into edge devices to enable secure, decentralized AI processing without exposing raw data to external threats.

Edge AI hardware is evolving rapidly to support the increasing complexity and diversity of AI workloads at the edge. This evolution will be driven by advances in specialized hardware accelerators, energy-efficient processing architectures, and secure computation technologies, ensuring that edge AI systems can meet the performance, energy, and security requirements of modern applications.

8.9 Heterogeneity

In edge AI environments, heterogeneity refers to the diverse nature of data, devices, and communication networks across edge nodes. This diversity introduces challenges in federated learning, where edge devices typically possess non-identically distributed (non-IID) data, varied computational capacities, and inconsistent communication bandwidths. As traditional FL approaches often rely on a single global model, they may struggle to capture the diverse patterns and distributions present across different edge devices. To address these challenges, multi-prototype-based federated learning [208] has emerged as a promising approach for enhancing model inference by leveraging multiple weighted prototypes rather than relying on a single prototype, which can be incomplete and ambiguous [94].

A key aspect of this approach involves calculating local prototypes at each edge device, ensuring that the diverse distributions of client data are effectively represented. These prototypes capture the unique characteristics of each client's data distribution, enabling a more nuanced aggregation process during global model updates. Clustering algorithms like **k-means** [208] can be employed locally at each edge device to generate multiple prototypes that correspond to different data clusters within the client's dataset. By calculating prototypes that reflect the underlying structure of local data, this approach enhances the model's ability to generalize across heterogeneous client distributions. Furthermore, these multiple weighted prototypes provide a richer representation of client data compared to traditional single-prototype methods, which often oversimplify local data distributions.

Once local prototypes are calculated, these prototypes are aggregated across devices during the global model update process. Rather than averaging model updates from each device, as in traditional FL approaches, multi-prototype-based FL aggregates the prototypes in a weighted manner, where each prototype's contribution is proportional to the importance of the corresponding data cluster. This approach improves robustness against non-IID [209] data distributions by ensuring that the global model is not overly influenced by outliers or overrepresented data points. Instead, the weighted aggregation process captures the full diversity of data across the edge devices, leading to higher test accuracy and better generalization across all devices.

An important challenge in this approach is ensuring communication efficiency, especially in bandwidth-constrained edge environments. By reducing the need to transmit the entire local model or large datasets, multi-prototype-based FL [210] minimizes communication overhead. Instead of sharing raw data or full model updates, devices only transmit the prototypes and their associated weights. This reduces the amount of information exchanged between the server and the devices, while still enabling effective global model updates. **Quantization** techniques can also be applied to further reduce the size of transmitted prototypes, allowing for more efficient communication without compromising model performance.

In addition to improving communication efficiency, multi-prototype-based FL demonstrates significant improvements in both **accuracy** and **convergence rates**. The ability to capture more representative patterns from local data allows the global model [210] to converge faster, particularly in heterogeneous environments where traditional FL methods tend to suffer from slow convergence due to non-IID data. The richer and more representative model built through multi-prototype aggregation helps the system achieve higher accuracy in a shorter period, reducing the number of communication rounds required to achieve an optimal model.

This approach demonstrates significant improvements in accuracy and convergence rates, making it a promising

direction for handling heterogeneity in edge AI. The multi-prototype strategy opens new avenues for edge AI applications where data heterogeneity is a major bottleneck, such as in **personalized healthcare**, where patient data varies significantly across different locations, or in **smart city infrastructures**, where sensor data may differ drastically based on environmental conditions. As future research progresses, integrating other advanced clustering techniques like **Gaussian Mixture Models (GMMs)** [211] or **spectral clustering** could further enhance the capability of multi-prototype FL systems, enabling even better handling of non-IID data and improving performance in highly heterogeneous environments.

8.10 Security

As edge AI becomes more prevalent, the security of edge devices, data, and communications becomes a critical concern, particularly with the growing threat of quantum computing, which can break classical encryption methods. One of the primary directions for securing edge AI systems is the integration of **AI-based quantum-safe cybersecurity automation**. Quantum-safe cryptographic techniques are essential for protecting sensitive data from the computational power of quantum computers, which could easily break traditional public-key encryption systems. Algorithms such as **lattice-based cryptography**, **hash-based signatures**, and **code-based cryptography** are being researched as quantum-resistant alternatives that can secure edge AI devices and communications against quantum attacks [212]. These quantum-safe solutions ensure that even as quantum computing capabilities advance, edge AI systems remain resilient against cryptographic threats.

Improving device and sensor security is another critical focus area. Edge devices, by their nature, are distributed and often deployed in insecure environments, making them vulnerable to physical tampering and cyberattacks. Integrating AI-based security mechanisms that can detect abnormal behavior at the device level is essential for enhancing the security of edge networks. For example, **deep learning-based intrusion detection systems (IDS)** can be implemented at the edge to monitor incoming traffic for potential threats, such as denial-of-service (DoS) attacks or unauthorized access attempts. These IDS systems can utilize **anomaly detection algorithms**, such as autoencoders or generative adversarial networks (GANs), to identify deviations from normal traffic patterns and flag potential security breaches in real time. Such AI-driven systems can adapt over time, learning from new attack vectors and updating their detection models to address emerging threats. Additionally, lightweight **blockchain-based solutions** can be integrated to ensure the secure exchange of data between edge devices by creating immutable records of transactions, further reducing the risk of tampering [213].

In the context of quantum-safe solutions, edge AI systems must also adopt **post-quantum cryptographic algorithms** to ensure long-term data security. Post-quantum cryptography (PQC) focuses on developing encryption algorithms that are resistant to both classical and quantum attacks. Integrating PQC into edge AI devices ensures that secure communications are maintained even when quantum computers become widely available. Furthermore, using AI to optimize the implementation of PQC algorithms [214], such as by reducing their computational overhead, can make these solutions more practical for deployment in resource-constrained edge environments. Research is also focusing on **quantum key distribution (QKD)** [215], a technique that leverages quantum mechanics to generate provably secure cryptographic keys. QKD can be used to secure communication between edge devices and central servers, ensuring that keys cannot be intercepted or tampered with, even by quantum adversaries.

The research will focus on developing scalable and efficient cybersecurity systems, including AI-driven automation for threat detection and mitigation and using blockchain for secure communications [213]. A key challenge in securing edge AI is the need for **scalability**. As the number of connected devices in edge networks increases, cybersecurity solutions must be able to scale to protect millions of devices without introducing significant latency or computational overhead. AI-driven security systems can help address this challenge by automating threat detection and response processes. For example, machine learning models can be trained to detect anomalies in device behavior or network traffic, identifying potential cyberattacks before they can cause damage. In addition to intrusion detection, AI can also automate **patch management** by identifying vulnerabilities in edge devices and applying security updates in real time, ensuring that devices remain protected against known threats.

Additionally, creating robust test environments for cybersecurity validation will ensure the effectiveness of these solutions in diverse operational scenarios [216]. Test environments, such as **digital twins** of edge networks, can be used to simulate cyberattacks and validate the effectiveness of AI-driven security solutions. By replicating real-world conditions, these environments enable researchers to fine-tune their algorithms and improve the resilience of edge AI systems. For example, by simulating distributed denial-of-service (DDoS) attacks on a digital twin of an edge network [217], AI-based security systems can be stress-tested and adjusted to ensure their ability to respond to large-scale cyber threats in real time.

Furthermore, AI-based systems can be used to predict potential vulnerabilities in edge networks by analyzing historical data and identifying patterns that could lead to future security breaches. This proactive approach to security helps to mitigate risks before they can be exploited by attackers.

These developments aim to provide a comprehensive security framework for future edge AI systems, ensuring resilience against evolving cyber threats. The future of edge AI security lies in the combination of quantum-safe cryptography [218], AI-driven threat detection, and automation. These technologies will allow edge AI systems to remain secure in the face of both classical and quantum-based cyber threats, ensuring that they can operate reliably in increasingly complex and hostile environments. As the number of connected devices in smart cities, autonomous vehicles, and industrial IoT grows, maintaining robust security at the edge will be critical to safeguarding sensitive data and ensuring the integrity of AI-driven processes.

8.11 Privacy

Privacy enhancement in early health prediction through federated learning would be another interesting area to investigate [219]. Future directions in this field involve developing more advanced privacy-preserving techniques within federated learning frameworks to keep patient data secure during model training. Improvements in differential privacy and homomorphic encryption are essential for protecting sensitive health information [80]. Additionally, optimizing the communication efficiency between edge devices and central servers will mitigate privacy risks associated with data transmission. Integrating privacy-conscious AI models with real-time health monitoring systems, like wearable devices, can deliver immediate and secure health insights. Collaboration among healthcare providers, AI researchers, and policymakers is vital to creating standardized privacy protocols. Future research should also focus on scalable and adaptive federated learning methods capable of handling diverse and large-scale health data while maintaining high privacy standards.

8.12 6 G and beyond

In the context of 6 G and beyond, Edge AI is set to make several significant advancements; utilizing the ultra-low latency and high bandwidth of 6 G networks will improve the deployment of AI models at the edge, facilitating real-time applications such as autonomous vehicles and smart cities [5]. The research will prioritize optimizing AI algorithms to meet the requirements of 6 G, including dynamic resource allocation and energy efficiency. Combining quantum ML with 6 G will enable more complex computations at the edge, enhancing predictive accuracy and decision-making processes. Additionally, improvements in secure edge computing and blockchain technology will address data privacy and security issues, ensuring robust and reliable edge AI systems. These advancements will collectively enhance the performance, scalability, and security of edge AI applications in a 6 G environment [21].

9 Summary and conclusions

The systematic review analysis of Edge AI provides a comprehensive overview of the present status of research in edge intelligence and its applications. The significance of these findings lies in the need for Edge AI systems to consider infrastructure, resource management, and the scale of ML models. According to the findings of the study, it is essential to conduct a thorough examination of both the positive and negative aspects of prior research to identify any potential research gaps and to estimate prospective developments and concerns.

The study emphasises the importance of using a systematic approach to record and evaluates the existing research in the field of Edge AI. Moreover, it highlights the importance of implementing a standardized procedure to reduce the possible impact of discrepancies in the study. The results of this review have the capacity to ignite a new field of investigation in Edge AI and offer direction for prospective research in this field. The exhaustive examination of Edge AI offers profound insight into the most current study on edge intelligence and its realistic implementations. Moreover, this emphasises the importance of gaining additional understanding about the key factors that govern the choice of Edge AI infrastructure, as well as the effect of the scale of the model used for ML on efficiency and resource allocation.

To summarize, the comprehensive study on Edge AI is to fully evaluate the various AI methodologies. This study integrates all the feasible methodologies incorporated in edge intelligence or AI at the edge. This review is to examine the crucial factors that impact the choice of Edge AI infrastructure, such as Cloud, Fog, and Edge computing, and assess their impact on application efficacy and resource utilization. Furthermore, it investigates the influence of the size of an ML model on the efficacy and resource usage of an Edge AI application. A total of 78 studies have been chosen for this evaluation due to their specific focus on the application of AI in edge computing. In order to enhance our comprehension in the context of AI applied to edge computing, these studies were categorized into multiple domains, including infrastructure, resource management, and ML model sizing, among others. Resource supply,

allocation, scheduling, and job deployment are crucial considerations in the field of Edge AI.

9.1 Open challenges

Following facts can be further concluded to improve this survey:

- EdgeAI Infrastructure Optimization: Future studies can examine Edge, Fog and Cloud systems that can be integrated into EdgeAI to create a hybrid model. In this way, scalable infrastructure solutions for EdgeAI systems can be discussed in detail. The research can focus on optimization techniques for resource allocation and latency in systems with varying workloads.
- Security and Privacy in EdgeAI: Considering the heterogeneous structure of the nodes that make up EdgeAI systems, security and privacy issues arise for sensitive data (biometric data). Future research can examine the measures and technical methods taken to ensure the security and privacy of data.
- EdgeAI Applications: Future studies can examine real-world EdgeAI applications such as smart cities and IoT-based healthcare systems. In this way, application challenges and solutions can be provided for researchers.

Author contributions Sukhpal Singh Gill (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Muhammed Golec (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Jianmin Hu (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Minxian Xu (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Junhui Du (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Huaming Wu (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Guneet Kaur Walia (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Software: Lead; Validation: Lead; Writing - original draft: Lead) Subramaniam Subramanian Murugesan (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Babar Ali (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Writing - original draft: Lead) Mohit Kumar (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Kejiang Ye (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Prabal Verma (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Surendra Kumar (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Felix Cuadrado (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead) Steve Uhlig (Conceptualization: Lead; Data curation: Lead; Formal analysis: Lead; Investigation: Lead; Methodology: Lead; Validation: Lead; Writing - original draft: Lead).

Funding M. Golec is supported by the Ministry of Education of the Turkish Republic. B. Ali is supported by the Ph.D. Scholarship at the Queen Mary University of London. H. Wu is supported by the National Natural Science Foundation of China (No. 62071327) and Tianjin Science and Technology Planning Project (No. 22ZYYYJC00020). F. Cuadrado has been supported by the HE ACES project (Grant No. 101093126). M. Xu is supported by the National Natural Science Foundation of China (No. 62102408), Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515010251), Shenzhen Industrial Application Projects of undertaking the National key R & D Program of China (No. CJGJZD20210408091600002).

Data availability Not Available

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no Conflict of interest.

Ethics approval Not Available.

Consent to participate Not Available.

Consent for publication Not Available.

Code availability Not Available.

References

1. Walia, G.K., Kumar, M., Gill, S.S.: Ai-empowered fog/edge resource management for iot applications: a comprehensive review, research challenges, and future perspectives. IEEE Commun. Surv. Tutor. **26**(1), 619–669 (2024)
2. Ding, A.Y., Peltonen, E., Meuser, T., et al.: Roadmap for edge ai: A dagstuhl perspective. ACM SIGCOMM Comput. Commun. Rev. **52**, 28–33 (2022)
3. Golec, M., Gill, S.S.: Computing: Looking back and moving forward. In: Proceedings of the 21st International Conference on Smart Business Technologies (ICSBT 2024), pp. 7–14, (2024)
4. Iftikhar, S., Gill, S.S., Song, C., Xu, M., Aslanpour, M.S., Toosi, A.N., Du, J., Wu, H., Ghosh, S., Chowdhury, D., et al.: Ai-based fog and edge computing: a systematic review, taxonomy and future directions. Internet Things **21**, 100674 (2023)
5. Duan, Q., Huang, J., Hu, S., Deng, R., Lu, Z., Yu, S.: Combining federated learning and edge computing toward ubiquitous intelligence in 6g network: Challenges, recent advances, and future directions. IEEE Commun. Surv. Tutor. (2023)
6. Singh, R., Gill, S.S.: Edge ai: a survey. Internet Things Cyber Phys. Syst. **3**, 71–92 (2023)
7. Shi, Y., Yang, K., Jiang, T., Zhang, J., Letaief, K.B.: Communication-efficient edge ai: algorithms and systems. IEEE Commun. Surv. Tutor. **22**(4), 2167–2191 (2020)

8. Liu, D., Kong, H., Luo, X., Liu, W., Subramaniam, R.: Bringing ai to edge: from deep learning's perspective. Neurocomputing **485**, 297–320 (2022)
9. Rocha, A., Monteiro, M., et al.: Edge ai for internet of medical things: a literature review. Comput. Electr. Eng. **116**, 109202 (2024)
10. Su, W., Li, L., Liu, F., He, M., Liang, X.: Ai on the edge: a comprehensive review. Artif. Intell. Rev. **55**(8), 6125–6183 (2022)
11. Zhang, W., Zeadally, S., Li, W., Zhang, H., Hou, J., Leung, V.C.M.: Edge ai as a service: configurable model deployment and delay-energy optimization with result quality constraints. IEEE Trans. Cloud Comput. **11**(2), 1954–1969 (2023)
12. Qureshi, H.N., Masood, U., Manalastas, M., Zaidi, S.M.A., Farooq, H., Forgeat, J., Bouton, M., Bothe, S., Karlsson, P., Rizwan, A., et al.: Towards addressing training data scarcity challenge in emerging radio access networks: a survey and framework. IEEE Commun. Surv. Tutor. (2023)
13. Golec, M., Iftikhar, S., Prabhakaran, P., Gill, S.S., Uhlig, S.: Qos analysis for serverless computing using machine learning. In: Serverless Computing: Principles and Paradigms, pp. 175–192. Springer, New York (2023)
14. Shahriar, S., Allana, S., Hazratifard, S.M., Dara, R.: A survey of privacy risks and mitigation strategies in the artificial intelligence life cycle. IEEE Access **11**, 61829–61854 (2023)
15. Kumar, M., Walia, G.K., Shingare, H., Singh, S., Gill, S.S.: Ai-based sustainable and intelligent offloading framework for iiot in collaborative cloud-fog environments. IEEE Trans. Consum. Electr. (2023)
16. Hoffpauir, K., Simmons, J., Schmidt, N., Pittala, R., Briggs, I., Makani, S., Jararweh, Y.: A survey on edge intelligence and lightweight machine learning support for future applications and services. ACM J. Data Inf. Qual. **15**(2), 1–30 (2023)
17. Gill, S.S., Buyya, R.: Transforming research with quantum computing. J. Economy Technol. **2**, 1–11 (2024)
18. Huang, N., Dou, C., Wu, Y., Qian, L., Lu, R.: Energy-efficient integrated sensing and communication: a multi-access edge computing design. IEEE Wireless Commun Lett. (2023)
19. Verma, P., Sood, S.K., Kaur, H., Kumar, M., Wu, H., Gill, S.S.: Data driven stochastic game network-based smart home monitoring system using iot-enabled edge computing environments. IEEE Trans. Consum. Electr. (2024)
20. Gill, S.S., Wu, H., Patros, P., Ottaviani, C., Arora, P., Pujol, V.C., Haunschild, D., Parlikad, A.K., Cetinkaya, O., Lutfiyya, H., et al.: Modern computing: vision and challenges. Telemat. Inform. Rep. **13**, 100116 (2024)
21. Velu, S., Gill, S.S., Murugesan, S.S., Wu, H., Li, X.: Cloudaibus: a testbed for ai based cloud computing environments. Cluster Comput. (2024). https://doi.org/10.1007/s10586-024-04562-9
22. Golec, M., Gill, S.S., Bahsoon, R., Rana, O.: Biosec: a biometric authentication framework for secure and private communication among edge devices in iot and industry 4.0. IEEE Consum. Electr. Magazine **11**(2), 51–56 (2020)
23. Golec, M., Gill, S.S., Cuadrado, F., Parlikad, A.K., Xu, M., Wu, H., Uhlig, S.: Atom: Ai-powered sustainable resource management for serverless edge computing environments. IEEE Trans. Sustain. Comput. (2023)
24. Golec, M., Ozturac, R., Pooranian, Z., Gill, S.S., Buyya, R.: Ifaasbus: a security-and privacy-based lightweight framework for serverless computing using iot and machine learning. IEEE Trans. Industr. Inform. **18**(5), 3522–3529 (2021)
25. Golec, M., Gill, S.S., Parlikad, A.K., Uhlig, S.: Healthfaas: Ai based smart healthcare system for heart patients using serverless computing. IEEE Int. Things J. (2023)
26. Peter, N.: Fog computing and its real time applications. Int. J. Emerg. Technol. Adv. Eng **5**(6), 266–269 (2015)
27. Iftikhar, S., Golec, M., Chowdhury, D., Gill, S.S., Uhlig, S., Fog computing based router-distributor application for sustainable smart home. In: IEEE 95th Vehicular Technology Conference:(VTC2022-Spring). IEEE **2022**, pp. 1–5 (2022)
28. Golec, M., Golec, M., Xu, M., Wu, H., Gill, S.S., Uhlig, S.: Priceless: Privacy enhanced ai-driven scalable framework for iot applications in serverless edge computing environments. Int. Technol. Lett., p. e510, (2024)
29. Golec, M., Gill, S.S., Wu, H., Can, T.C., Golec, M., Cetinkaya, O., Cuadrado, F., Parlikad, A. K., Uhlig, S.: Master: Machine learning-based cold start latency prediction framework in serverless edge computing environments for industry 4.0. IEEE J. Select. Areas Sens. (2024)
30. Gill, S.S.: A manifesto for modern fog and edge computing: Vision, new paradigms, opportunities, and future directions. In: Operationalizing Multi-Cloud Environments: Technologies, pp. 237–253. Tools and Use Cases. Springer, New York (2021)
31. Nandhakumar, A.R., Baranwal, A., Choudhary, P., Golec, M., Gill, S.S.: Edgeaisim: a toolkit for simulation and modelling of ai models in edge computing environments. Measurement Sens. **31**, 100939 (2024)
32. Golec, M., Chowdhury, D., Jaglan, S., Gill, S.S., Uhlig, S.: Aiblock: Blockchain based lightweight framework for serverless computing using ai. In: 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid). IEEE **2022**, pp. 886–892 (2022)
33. Lee, C.P., Leng, F.T.J., Habeeb, R.A.A., Amanullah, M.A., ur Rehman, M.H.: Edge computing-enabled secure and energy-efficient smart parking: a review. Microprocess. Microsy. **93**, 104612 (2022)
34. Patrikar, D.R., Parate, M.R.: Anomaly detection using edge computing in video surveillance system. Int. J. Multimed. Inf. Retr. **11**(2), 85–110 (2022)
35. Wu, L., Zhang, R., Zhou, R., Wu, D.: An edge computing based data detection scheme for traffic light at intersections. Comput. Commun. **176**, 91–98 (2021)
36. Liu, Y.: Open university Chinese language and literature teaching model based on nlp technology and mobile edge computing. Mobile Inf. Syst. **2022**(1), 4149492 (2022)
37. Barekar, P.V., Singh, K.R.: Object detection and tracking approach for traffic monitoring. In: International Conference on Smart Computing and Communication. Springer, pp. 25–33 (2024)
38. Golec, M., Gill, S.S., Golec, M., Xu, M., Ghosh, S.K., Kanhere, S.S., Rana, O., Uhlig, S.: Blockfaas: blockchain-enabled serverless computing framework for ai-driven iot healthcare applications. J. Grid Comput. **21**(4), 63 (2023)
39. Cao, K., Liu, Y., Meng, G., Sun, Q.: An overview on edge computing research. IEEE access **8**, 85 714-85 728 (2020)
40. Dolati, M., Rastegar, S.H., Khonsari, A., Ghaderi, M.: Layer-aware containerized service orchestration in edge networks. IEEE Trans. Network Serv. Manage. **20**(2), 1830–1846 (2022)
41. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
42. Larsson, M.: Hands-on Microservices with spring boot and spring cloud: build and deploy Java microservices using spring cloud, Istio, and Kubernetes. Packt Publishing Ltd, Birmingham (2019)
43. Xu, X., Huang, Q., Yin, X., Abbasi, M., Khosravi, M.R., Qi, L.: Intelligent offloading for collaborative smart city services in edge computing. IEEE Int. Things J. **7**(9), 7919–7927 (2020)
44. Huang, H., Peng, K., Xu, X., Collaborative computation offloading for smart cities in mobile edge computing. In: IEEE

13th International conference on cloud computing (CLOUD). IEEE **2020**, 176–183 (2020)
45. Li, C., Wang, H., Song, R.: Intelligent offloading for noma-assisted mec via dual connectivity. IEEE Int. Things J. **8**(4), 2802–2813 (2020)
46. Zhang, Y., Liu, X., Xu, J., Yuan, D., Li, X., A novel adaptive computation offloading strategy for collaborative dnn inference over edge devices. In: IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE **2022**, pp. 378–385 (2022)
47. Fresa, A., Champati, J.P.V.: An offloading algorithm for maximizing inference accuracy on edge device in an edge intelligence system. In: Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems, pp. 15–23 (2022)
48. Khan, I., Raza, S., Rehman, W.U., Khan, R., Nahida, K., Tao, X.: A deep learning-based algorithm for energy and performance optimization of computational offloading in mobile edge computing. Wireless Commun. Mobile Comput. **2023**(1), 1357343 (2023)
49. Du, M., Wang, Y., Ye, K., Xu, C.: Algorithmics of cost-driven computation offloading in the edge-cloud environment. IEEE Trans. Comput. **69**(10), 1519–1532 (2020)
50. Choudhury, A.S., Halder, T., Basak, A., Chakravarty, D.: Implementation of artificial intelligence (ai) in smart manufacturing: a status review. In: International Conference on Computational Intelligence and Smart Communication, pp. 73–85. Springer, New York (2022)
51. Plathottam, S.J., Rzonca, A., Lakhnori, R., Iloeje, C.O.: A review of artificial intelligence applications in manufacturing operations. J. Adv. Manufact. Process. **5**(3), e10159 (2023)
52. Yang, C., Lan, S., Wang, L., Shen, W., Huang, G.G.: Big data driven edge-cloud collaboration architecture for cloud manufacturing: a software defined perspective. IEEE Access **8**, 45938–45950 (2020)
53. Moon, J., Jeong, J., Smart manufacturing scheduling system: Dqn based on cooperative edge computing. In: 15th international conference on ubiquitous information management and communication (IMCOM). IEEE **2021**, pp. 1–8 (2021)
54. Mishra, I., Kumar, S., Gupta, N.: Smart manufacturing: opportunities and challenges overcome by industry 4.0. Soft Computing in Materials Development and its Sustainability in the Manufacturing Sector, pp. 179–182 (2022)
55. Zhang, Y., Tang, D., Zhu, H., Zhou, S., Zhao, Z.: An efficient iiot gateway for cloud-edge collaboration in cloud manufacturing. Machines **10**(10), 850 (2022)
56. Mohanram, P., Gilerson, A., Schmitt, R., et al.: Architecture for edge-based predictive maintenance of machines using federated learning and multi sensor platforms. (2023)
57. Rizk, H., Chaibet, A., Kribèche, A.: Model-based control and model-free control techniques for autonomous vehicles: a technical survey. Appl. Sci. **13**(11), 6700 (2023)
58. Ning, Z., Hu, H., Wang, X., Guo, L., Guo, S., Wang, G., Gao, X.: Mobile edge computing and machine learning in the internet of unmanned aerial vehicles: a survey. ACM Comput. Surv. **56**(1), 1–31 (2023)
59. Ahmed, M., Mirza, M.A., Raza, S., Ahmad, H., Xu, F., Khan, W.U., Lin, Q., Han, Z.: Vehicular communication network enabled cav data offloading: a review. IEEE Trans. Intell. Trans. Syst. (2023)
60. Xue, D., Guo, Y., Li, N., Song, X., Zhang, L.: Cross-domain coordination of resource allocation and route planning for the edge computing-enabled multi-connected vehicles. J. Cloud Comput. **12**(1), 33 (2023)
61. Ming, G.: Exploration of the intelligent control system of autonomous vehicles based on edge computing. PLoS One **18**(2), e0281294 (2023)
62. Firdaus, M., Rhee, K.-H.: A joint framework to privacy-preserving edge intelligence in vehicular networks. In: Firdaus, M. (ed.) Int. Conf. Inform. Security Appl., pp. 156–167. Springer, New York (2022)
63. Atan, B., Basaran, M., Calik, N., Basaran, S.T., Akkuzu, G., Durak-Ata, L.: Ai-empowered fast task execution decision for delay-sensitive iot applications in edge computing networks. IEEE Access **11**, 1324–1334 (2022)
64. Anees, T., Habib, Q., Al-Shamayleh, A.S., Khalil, W., Obaidat, M.A., Akhunzada, A.: The integration of wot and edge computing: issues and challenges. Sustainability **15**(7), 5983 (2023)
65. Ajibuwa, O., Hamdaoui, B., Yavuz, A.A.: A survey on ai/ml-driven intrusion and misbehavior detection in networked autonomous systems: techniques, challenges and opportunities. arXiv preprint arXiv:2305.05040, (2023)
66. Verma, P., Sood, S.K.: Fog assisted-iot enabled patient health monitoring in smart homes. IEEE Int. Things J. **5**(3), 1789–1796 (2018)
67. Shaikh, T.A., Rasool, T., Verma, P.: Machine intelligence and medical cyber-physical system architectures for smart healthcare: taxonomy, challenges, opportunities, and possible solutions. Artif. Intell. Med. **146**, 102692 (2023)
68. Ahmed, S.T., Basha, S.M., Ramachandran, M., Daneshmand, M., Gandomi, A.H.: An edge-ai enabled autonomous connected ambulance route resource recommendation protocol (aca-r3) for ehealth in smart cities. IEEE Int. Things J. (2023)
69. Misra, S., Pal, S., Deb, P.K., Gupta, E.: Kedge: Fuzzy-based multi-ai model coalescence solution for mobile healthcare system. IEEE Syst. J. (2023)
70. Chakraborty, C., Nagarajan, S.M., Devarajan, G.G., Ramana, T., Mohanty, R.: Intelligent ai-based healthcare cyber security system using multi-source transfer learning method. ACM Trans. Sens. Netw. (2023)
71. Dvijotham, K., Winkens, J., Barsbey, M., Ghaisas, S., Stanforth, R., Pawlowski, N., Strachan, P., Ahmed, Z., Azizi, S., Bachrach, Y., et al.: Enhancing the reliability and accuracy of ai-enabled diagnosis via complementarity-driven deferral to clinicians. Nat. Med. **29**(7), 1814–1820 (2023)
72. Keele, S., et al.: Guidelines for performing systematic literature reviews in software engineering. Tech. Rep, Citeseer (2007)
73. Kitchenham, B.A.: Systematic review in software engineering: where we are and where we should be going. In: Proceedings of the 2nd international workshop on Evidential assessment of software technologies, pp. 1–2 (2012)
74. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. J. Syst. Softw. **80**(4), 571–583 (2007)
75. Tawfik, G.M., Dila, K.A.S., Mohamed, M.Y.F., Tam, D.N.H., Kien, N.D., Ahmed, A.M., Huy, N.T.: A step by step guide for conducting a systematic review and meta-analysis with simulation data. Trop Med Health **47**, 1–9 (2019)
76. Singh, S.P., Sharma, A., Kumar, R.: Design and exploration of load balancers for fog computing using fuzzy logic. Simul Model Pract Theory **101**, 102017 (2020)
77. Gos, K., Zabierowski, W.: The comparison of microservice and monolithic architecture. In: 2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). IEEE, pp. 150–153, (2020)
78. Errasti-Alcala, B., Fernandez-Recio, R.: Meta-heuristic approach for single-snapshot 2d-doa and frequency estimation: Array topologies and performance analysis [wireless corner]. IEEE Antenn Propag Magazine **55**(1), 222–238 (2013)

79. Himeur, Y., Sayed, A., Alsalemi, A., Bensaali, F., Amira, A.: Edge ai for internet of energy: Challenges and perspectives. ArXiv, vol. arXiv:abs/2311.16851, (2023)
80. Gill, S.S.: Quantum and blockchain based serverless edge computing: a vision, model, new trends and future directions. Int. Technol. Lett. **7**(1), e275 (2024)
81. Gill, S.S., Xu, M., Ottaviani, C., Patros, P., Bahsoon, R., Shaghaghi, A., Golec, M., Stankovski, V., Wu, H., Abraham, A., et al.: Ai for next generation computing: Emerging trends and future directions. Internet Things **19**, 100514 (2022)
82. Sharif, Z., Jung, L.T., Ayaz, M., Yahya, M., Pitafi, S.: Priority-based task scheduling and resource allocation in edge computing for health monitoring system. J. King Saud Univ. Comput. Inform. Sci. **35**(2), 544–559 (2023)
83. Zhuang, Z., Li, Y., Sun, Y., Qin, W., Sun, Z.-H.: Network-based dynamic dispatching rule generation mechanism for real-time production scheduling problems with dynamic job arrivals. Robot. Comput. Integr. Manuf. **73**, 102261 (2022)
84. Singh, H., Tyagi, S., Kumar, P., Gill, S.S., Buyya, R.: Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: analysis, performance evaluation, and future directions. Simul. Model. Pract. Theory **111**, 102353 (2021)
85. Desai, F., Chowdhury, D., Kaur, R., Peeters, M., Arya, R.C., Wander, G.S., Gill, S.S., Buyya, R.: Healthcloud: a system for monitoring health status of heart patients using machine learning and cloud computing. Internet Things **17**, 100485 (2022)
86. Sheng, S., Chen, P., Chen, Z., Wu, L., Yao, Y.: Deep reinforcement learning-based task scheduling in iot edge computing. Sensors **21**(5), 1666 (2021)
87. Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A.A., Yogamani, S., Pérez, P.: Deep reinforcement learning for autonomous driving: a survey. IEEE Trans. Intell. Trans. Syst. **23**(6), 4909–4926 (2021)
88. Zhong, Z., Xu, M., Rodriguez, M.A., Xu, C., Buyya, R.: Machine learning-based orchestration of containers: a taxonomy and future directions. ACM Comput. Surv. (CSUR) **54**(10s), 1–35 (2022)
89. Iftikhar, S., Raj, U., Tuli, S., Golec, M., Chowdhury, D., Gill, S.S., Uhlig, S., Tesco: Multiple simulations based ai-augmented fog computing for qos optimization. In: IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta). IEEE **2022**, 2092–2099 (2022)
90. Nayeri, Z.M., Ghafarian, T., Javadi, B.: Application placement in fog computing with ai approach: taxonomy and a state of the art survey. J. Netw. Comput. Appl. **185**, 103078 (2021)
91. Carvalho, O., Garcia, M., Roloff, E., Carreño, E.D., Navaux, P.O.: Iot workload distribution impact between edge and cloud computing in a smart grid application. In: High Performance Computing: 4th Latin American Conference, CARLA,: Buenos Aires, Argentina, and Colonia del Sacramento, Uruguay, September 20–22, 2017, Revised Selected Papers 4. Springer **2018**, 203–217 (2017)
92. Nguyen, C., Klein, C., Elmroth, E.: Multivariate lstm-based location-aware workload prediction for edge data centers. In: 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE **2019**, 341–350 (2019)
93. Winters, P.R.: Forecasting sales by exponentially weighted moving averages. Manag. Sci. **6**(3), 324–342 (1960)
94. Qiao, Y., Munir, M.S., Adhikary, A., Raha, A.D., Hong, S.H., Hong, C.S.: A framework for multi-prototype based federated learning: Towards the edge intelligence. In: 2023 International Conference on Information Networking (ICOIN). IEEE, pp. 134–139 (2023)
95. Briouya, A., Briouya, H., Choukri, A.: Overview of the progression of state-of-the-art language models. TELKOMNIKA (Telecommunication Computing Electronics and Control) **22**(4), 897–909 (2024)
96. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708 (2017)
97. Tan, M., Le, Q.: Efficientnet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR, pp. 6105–6114 (2019)
98. Tan, M., Le, Q.: Efficientnetv2: smaller models and faster training. In: International conference on machine learning. PMLR, pp. 10096–10106 (2021)
99. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, (2017)
100. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6848–6856 (2018)
101. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Adv. Neural Inf. Process. Syst. **33**, 1877–1901 (2020)
102. Xiong, Y., Sun, Y., Xing, L., Huang, Y.: Extend cloud to edge with kubeedge. In: IEEE/ACM Symposium On Edge Computing (SEC). IEEE **2018**, pp. 373–377 (2018)
103. Singh, P., Kaur, A., Gill, S.S.: Machine learning for cloud, fog, edge and serverless computing environments: comparisons, performance evaluation benchmark and future directions. Int. J. Grid Utility Comput. **13**(4), 447–457 (2022)
104. You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J., Keutzer, K.: Fast deep neural network training on distributed systems and cloud tpus. IEEE Trans. Parallel Distribut. Syst. **30**(11), 2449–2462 (2019)
105. Sun, Y., Kist, A.M.: Deep learning on edge tpus. arXiv preprint arXiv:2108.13732, (2021)
106. Casalicchio, E., Iannucci, S.: The state-of-the-art in container technologies: application, orchestration and security. Concurr. Comput. Pract. Exp. **32**(17), e5668 (2020)
107. Zhang, J., Lu, C., Cheng, G., Guo, T., Kang, J., Zhang, X., Yuan, X., Yan, X.: A blockchain-based trusted edge platform in edge computing environment. Sensors **21**(6), 2126 (2021)
108. Wang, T., Zhang, G., Liu, A., Bhuiyan, M.Z.A., Jin, Q.: A secure iot service architecture with an efficient balance dynamics based on cloud and edge computing. IEEE Int. Things J. **6**(3), 4831–4843 (2018)
109. Gharaibeh, A., Salahuddin, M.A., Hussini, S.J., Khreishah, A., Khalil, I., Guizani, M., Al-Fuqaha, A.: Smart cities: a survey on data management, security, and enabling technologies. IEEE Commun. Surv. Tutor. **19**(4), 2456–2501 (2017)
110. Krebs, B.: Krebsonsecurity hit with record ddos. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-recordddos/ (2016). Accessed 15 Jun 2024 [Online]
111. Dyn, Dyn analysis summary of friday october 21 attack. Available: http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21attack/ (2016). Accessed 15 Jun 2024 [Online]
112. Bhardwaj, K., Miranda, J.C., Gavrilovska, A.: Towards IoT-DDoS prevention using edge computing. In: USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). Boston, MA: USENIX Association. [Online]. Available: https://www.usenix.org/conference/hotedge18/presentation/bhardwaj(2018)
113. Oleghe, O.: Container placement and migration in edge computing: concept and scheduling models. IEEE Access **9**, 68028–68043 (2021)

114. Li, Z., Yang, Z., Xie, S., Chen, W., Liu, K.: Credit-based payments for fast computing resource trading in edge-assisted internet of things. IEEE Int. Things J. **6**(4), 6606–6617 (2019)
115. Zhang, X., Zhong, Y., Liu, P., Zhou, F., Wang, Y.: Resource allocation for a uav-enabled mobile-edge computing system: computation efficiency maximization. IEEE Access **7**, 113 345-113 354 (2019)
116. Tran, T.X., Pompili, D.: Joint task offloading and resource allocation for multi-server mobile-edge computing networks. IEEE Trans. Vehicular Technol. **68**(1), 856–868 (2018)
117. Wei, Y., Pan, L., Liu, S., Wu, L., Meng, X.: Drl-scheduling: an intelligent qos-aware job scheduling framework for applications in clouds. IEEE Access **6**, 55112–55125 (2018)
118. Carrión, C.: Kubernetes scheduling: taxonomy, ongoing issues and challenges. ACM Comput. Surv. **55**(7), 1–37 (2022)
119. Junior, P.S., Miorandi, D., Pierre, G.: Stateful container migration in geo-distributed environments. In: IEEE international conference on cloud computing technology and science (CloudCom). IEEE **2020**, pp. 49–56 (2020)
120. Murugesan, S.S., Velu, S., Golec, M., Wu, H., Gill, S.S.: Neural networks based smart e-health application for the prediction of tuberculosis using serverless computing. IEEE J. Biomed. Health Inform., pp. 1–12, (2024)
121. Amin, R., Vadlamudi, S., Rahaman, M.M.: Opportunities and challenges of data migration in cloud. Eng. Int. **9**(1), 41–50 (2021)
122. Chiang, M., Zhang, T.: Fog and iot: an overview of research opportunities. IEEE Int. Things J. **3**(6), 854–864 (2016)
123. Iftikhar, S., Golec, M., Chowdhury, D., Gill, S.S., Uhlig, S.: Fogdlearner: a deep learning-based cardiac health diagnosis framework using fog computing. In: Proceedings of the 2022 Australasian Computer Science Week, pp. 136–144 (2022)
124. Imdoukh, M., Ahmad, I., Alfailakawi, M.G.: Machine learning-based auto-scaling for containerized applications. Neural Comput. Appl. **32**(13), 9745–9760 (2020)
125. The Prometheus Authors, Prometheus - monitoring system & time series database. https://prometheus.io/ (2024). Accessed 20 Jun 2024
126. Klinaku, F., Frank, M., Becker, S.: Caus: an elasticity controller for a containerized microservice. In: Companion of the. ACM/SPEC International Conference on Performance Engineering **2018**, pp. 93–98 (2018)
127. Ali, A.H.: A survey on vertical and horizontal scaling platforms for big data analytics. Int. J. Integr. Eng. **11**(6), 138–150 (2019)
128. Nguyen, T.-T., Yeom, Y.-J., Kim, T., Park, D.-H., Kim, S.: Horizontal pod autoscaling in kubernetes for elastic container orchestration. Sensors **20**(16), 4621 (2020)
129. Rossi, F., Nardelli, M., Cardellini, V.: Horizontal and vertical scaling of container-based applications using reinforcement learning. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). IEEE, pp. 329–338 (2019)
130. Hu, H., Jiang, C.: Edge intelligence: Challenges and opportunities. In: 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), pp. 1–5 (2020)
131. Dai, C., Song, Q.: Heuristic computing methods for contact plan design in the spatial-node-based internet of everything. China Commun. **16**(3), 53–68 (2019)
132. Mattmann, C.: Machine learning with tensorflow. Simon and Schuster (2020)
133. Kaloev, M., Krastev, G.: Experiments focused on exploration in deep reinforcement learning. In: 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 351–355 (2021)
134. Hong, C.-H., Varghese, B.: Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. ACM Comput. Surv. (CSUR) **52**(5), 1–37 (2019)
135. Ebrahimi, M., Veith, A.d.S., Gabel, M., de Lara, E.: Combining dnn partitioning and early exit. In: Proceedings of the 5th International Workshop on Edge Systems, Analytics and Networking, pp. 25–30 (2022)
136. Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., Tang, L.: Neurosurgeon: collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Comput. Arch. News **45**(1), 615–629 (2017)
137. Liang, C., Zuo, S., Chen, M., Jiang, H., Liu, X., He, P., Zhao, T., Chen, W.: Super tickets in pre-trained language models: from model compression to improving generalization. arXiv preprint arXiv:2105.12002, (2021)
138. Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R.: Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun. Surv. Tutor. **16**(1), 369–392 (2013)
139. Marin, G.: Network security basics. IEEE Secur. Priv. **3**(6), 68–72 (2005)
140. Puliafito, C., Vallati, C., Mingozzi, E., Merlino, G., Longo, F., Puliafito, A.: Container migration in the fog: a performance evaluation. Sensors **19**(7), 1488 (2019)
141. Singh, G., Singh, P.: A taxonomy and survey on container migration techniques in cloud computing. Sustain. Dev. Through Eng. Innov. Select Proc. SDEI **2020**, 419–429 (2021)
142. Kristiani, E., Yang, C.-T., Nguyen, K.L.P.: Optimization of deep learning inference on edge devices. 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), pp. 264–267, (2020)
143. Saha, S., Banerjee, K., Ghosh, S., Mitra, S., Pal, D.: Ai-driven edge computing for iot: a comprehensive survey and future directions. Int J Adv Res Sci Commun Technol (2023)
144. Liang, Q., Hanafy, W.A., Ali-Eldin, A., Shenoy, P.: Model-driven cluster resource management for ai workloads in edge clouds. ACM Trans. Auton. Adap. Syst. **18**, 1–26 (2022)
145. Shi, Y., Yang, K., Jiang, T., Zhang, J., Letaief, K.: Communication-efficient edge ai: Algorithms and systems. IEEE Commun. Surv. Tutor. **22**, 2167–2191 (2020)
146. Zou, X., Li, K., Zhou, J.T., Wei, W., Chen, C.: Robust edge ai for real-time industry 4.0 applications in 5g environment. IEEE Commun. Stand. Magazine **7**(2), 64–70 (2023)
147. Du, J., Xu, M., Gill, S.S., Wu, H.: Computation energy efficiency maximization for intelligent reflective surface-aided wireless powered mobile edge computing. IEEE Trans. Sustain. Comput. (2023)
148. Johnson, D., Chen, G., Lu, Y.: Multi-agent reinforcement learning for real-time dynamic production scheduling in a robot assembly cell. IEEE Robot. Autom. Lett. **7**, 7684–7691 (2022)
149. Dong, T., Xue, F., Xiao, C., Li, J.: Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. Concurr. Comput. Pract. Exp. **32**, e5654 (2020)
150. Chhabra, A., Singh, G., Kahlon, K.: Multi-criteria hpc task scheduling on iaas cloud infrastructures using meta-heuristics. Cluster Comput. **24**, 885–918 (2020)
151. Roeder, J., Rouxel, B., Altmeyer, S., Grelck, C.: Energy-aware scheduling of multi-version tasks on heterogeneous real-time systems. Proceedings of the 36th Annual ACM Symposium on Applied Computing (2021)
152. Yousef, L.A., Yousef, H., Rocha-Meneses, L.: Artificial intelligence for management of variable renewable energy systems: a review of current status and future directions. Energies **16**(24), 8057 (2023)
153. Chowdhury, D., Das, A., Dey, A., Banerjee, S., Golec, M., Kollias, D., Kumar, M., Kaur, G., Kaur, R., Arya, R.C., et al.: Covidetector: a transfer learning-based semi supervised approach to detect covid-19 using cxr images. BenchCouncil Trans. Benchmarks Stand. Eval. **3**(2), 100119 (2023)

154. Wang, Y., Dong, S., Fan, W.: Task scheduling mechanism based on reinforcement learning in cloud computing. Mathematics **11**(15), 3364 (2023)
155. Zhu, S., Ota, K., Dong, M.: Energy-efficient artificial intelligence of things with intelligent edge. IEEE Int. Things J. **9**(10), 7525–7532 (2022)
156. Doyle, J., Golec, M., Gill, S.S.: Blockchainbus: a lightweight framework for secure virtual machine migration in cloud federations using blockchain. Secur. Priv. **5**(2), e197 (2022)
157. Singh, R., Gill, S.S.: Next generation edge computing: a roadmap to net zero emissions. J. Econ. Technol. **1**, 208–221 (2023)
158. Vermesan, O., Coppola, M., Bahr, R., Bellmann, R.O., Martinsen, J.E., Kristoffersen, A., Hjertaker, T., Breiland, J., Andersen, K., Sand, H.-E., Lindberg, D.: An intelligent real-time edge processing maintenance system for industrial manufacturing, control, and diagnostic. [Online]. Available: https://api.semanticscholar.org/CorpusID:250150289 (2022)
159. Kovalenko, I., Barton, K., Moyne, J., Tilbury, D.M.: Opportunities and challenges to integrate artificial intelligence into manufacturing systems: thoughts from a panel discussion [opinion]. IEEE Robot. Autom. Magazine **30**(2), 109–112 (2023)
160. Cinar, Z., Nuhu, A.A., Zeeshan, Q., Korhan, O., Asmael, M., Safaei, B.: Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability **12**(19), 8211 (2020)
161. Chien, C., Dauzére-Pérés, S., Huh, W.T., Jang, Y., Morrison, J.R.: Artificial intelligence in manufacturing and logistics systems: algorithms, applications, and case studies. Int. J. Prod. Res. **58**, 2730–2731 (2020)
162. Ying, J., Hsieh, J., Hou, D., Hou, J., Liu, T., Zhang, X., Wang, Y., Pan, Y.-T.: Edge-enabled cloud computing management platform for smart manufacturing. 2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0 &IoT), pp. 682–686, (2021)
163. Nain, G., Pattanaik, K., Sharma, G.: Towards edge computing in intelligent manufacturing: past, present and future. J. Manuf. Syst. **62**, 588–611 (2022)
164. Ringler, N., Knittel, D., Ponsart, J., Nouari, M., Yakob, A., Romani, D.: Machine learning based real time predictive maintenance at the edge for manufacturing systems: a practical example. 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), pp. 1–6, (2023)
165. Pule, M., Matsebe, O., Samikannu, R.: (2022) Application of pca and svm in fault detection and diagnosis of bearings with varying speed. Math. Probl. Eng. **1**, 5266054 (2022)
166. Yu, W., Dillon, T., Mostafa, F., Rahayu, W., Liu, Y.: A global manufacturing big data ecosystem for fault detection in predictive maintenance. IEEE Trans. Industr. Inform. **16**, 183–192 (2020)
167. Thalluri, L.N., Venkat, S.N., Prasad, C.V.V.D., Kumar, D.V., Kumar, K.P., Sarma, A.V.N., Adapa, S.D.: Artificial intelligence enabled smart city iot system using edge computing. In: 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC). IEEE, pp. 12–20, (2021)
168. Cojbasic, S., Dmitrasinovic, S., Kostic, M., Sekulić, M.T., Radonić, J., Dodig, A., Stojkovic, M.: Application of machine learning in river water quality management: a review. Water Sci. Technol. J. Int. Assoc. Water Pollut. Res. **88**(9), 2297–2308 (2023)
169. Wang, H., Yuan, Y., Yang, X., Zhao, T., Liu, Y.: Deep q learning-based traffic signal control algorithms: model development and evaluation with field data. J. Intell. Trans. Syst. **27**, 314–334 (2022)
170. Wang, S., Wang, S.: A novel multi-agent deep rl approach for traffic signal control. 2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 15–20, (2023)
171. Ferraz, O., Araujo, H., Silva, V., Fernandes, G.F.P.: Benchmarking convolutional neural network inference on low-power edge devices. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5, (2023)
172. Marini, R., Mikhaylov, K., Pasolini, G., Buratti, C.: Low-power wide-area networks: comparison of lorawan and nb-iot performance. IEEE Int. Things J. **9**, 21051–21063 (2022)
173. Nguyen, D.C., Ding, M., Pham, V.Q., Pathirana, P., Bao, L., Aruna, L., Seneviratne, J., Li, D., Niyato, F.I.H.V., Poor, L.F.I., Le, L., Li, J., Niyato, D.: Federated learning meets blockchain in edge computing: opportunities and challenges. IEEE Int. Things J. **8**, 12 806-12 825 (2021)
174. Adiwijaya, J., Tanaya, V.R., Anderies, Chowanda, A.: Federated learning and differential privacy in ai-based surveillance systems model. 2023 14th International Conference on Information & Communication Technology and System (ICTS), pp. 283–288, (2023)
175. Mutalemwa, L., Shin, S.: A classification of the enabling techniques for low latency and reliable communications in 5g and beyond: Ai-enabled edge caching. IEEE Access **8**, 205 502-205 533 (2020)
176. Antevski, K., Girletti, L., Bernardos, C., de la Oliva, A., Baranda, J., Mangues-Bafalluy, J.: A 5g-based ehealth monitoring and emergency response system: experience and lessons learned. IEEE Access **9**, 131420–131429 (2021)
177. Adhikari, M., Hazra, A.: 6g-enabled ultra-reliable low-latency communication in edge networks. IEEE Commun. Stand. Magazine **6**, 67–74 (2022)
178. Guo, X., Hong, X.: Dqn for smart transportation supporting v2v mobile edge computing. 2023 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 204–206, (2023)
179. Liu, X.-Y., Zhu, M., Borst, S., Elwalid, A.: Deep reinforcement learning for traffic light control in intelligent transportation systems. ArXiv, vol. arXiv:abs/2302.03669, (2023)
180. Guo, X., Hong, X.: Dqn for smart transportation supporting v2v mobile edge computing. In: 2023 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, pp. 204–206 (2023)
181. Cheng, W., Liu, X., Wang, X., Nie, G.: Task offloading and resource allocation for industrial internet of things: a double-dueling deep q-network approach. IEEE Access **10**, 103 111-103 120 (2022)
182. Seid, A.M., Boateng, G.O., Mareri, B., Sun, G., Jiang, W.: Multi-agent drl for task offloading and resource allocation in multi-uav enabled iot edge network. IEEE Trans. Netw. Serv. Manage. **18**, 4531–4547 (2021)
183. Peng, X., Gao, H., Han, G., Wang, H., Zhang, M.: Joint optimization of traffic signal control and vehicle routing in signalized road networks using multi-agent deep reinforcement learning. ArXiv, vol. arXiv:abs/2310.10856, (2023)
184. Gao, L., Gu, Z., Qiu, C., Lei, L., Li, S., Zheng, S., Jing, W., Chen, J.: Cola-hrl: Continuous-lattice hierarchical reinforcement learning for autonomous driving. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 13143–13150, (2022)
185. Dai, J., Gluzman, M.O.: Queueing network controls via deep reinforcement learning. ArXiv, vol. arXiv:abs/2008.01644, (2020)
186. Baller, S.P., Jindal, A., Chadha, M., Gerndt, M.: Deepedgebench: benchmarking deep neural networks on edge devices. 2021 IEEE International Conference on Cloud Engineering (IC2E), pp. 20–30, (2021)

187. Pan, Q., Wu, J., Nebhen, J., Bashir, A., Su, Y., Li, J.: Artificial intelligence-based energy efficient communication system for intelligent reflecting surface-driven vanets. IEEE Trans. Intell. Trans. Syst. **23**, 19 714-19 726 (2022)
188. Golec, M., Walia, G.K., Kumar, M., Cuadrado, F., Gill, S.S., Uhlig, S.: Cold start latency in serverless computing: a systematic review, taxonomy, and future directions. arXiv preprint arXiv:2310.08437, (2023)
189. Eapen, B., Sartipi, K., Archer, N.: Serverless on fhir: deploying machine learning models for healthcare on the cloud. ArXiv, vol. arXiv:abs/2006.04748, (2020)
190. Solaiman, K., Adnan, M.A.: Wlec: A not so cold architecture to mitigate cold start problem in serverless computing. 2020 IEEE International Conference on Cloud Engineering (IC2E), pp. 144–153, (2020)
191. Senjab, K., Abbas, S., Ahmed, N., Khan, A.U.R.: A survey of kubernetes scheduling algorithms. J. Cloud Comput. **12**(1), 87 (2023)
192. Golec, M., Gill, S.S., Wu, H., Can, T.C., Golec, M., Cetinkaya, O., Cuadrado, F., Parlikad, A.K., Uhlig, S.: Master: machine learning-based cold start latency prediction framework in serverless edge computing environments for industry 4.0. IEEE J. Select. Areas Sens. **1**, 36–48 (2024)
193. Wei, Y., Wang, X., Bian, S., Zhao, W., Jin, Y.: The-v: Verifiable privacy-preserving neural network via trusted homomorphic execution. 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD), pp. 1–9, (2023)
194. Jaffe, C., Vreeman, D., Kaminker, D., Nguyen, V.: Implementing hl7 fhir. J. Healthcare Manage. Stand. (2023)
195. Gill, S.S., et al.: Quantum computing: vision and challenges. Technical Report, arXiv preprint arXiv:2403.02240, pp. 1–11, (2024)
196. Dupont, M., Didier, N., Hodson, M., Moore, J., Reagor, M.: Entanglement perspective on the quantum approximate optimization algorithm. Phys. Rev. A **106**(2), 022423 (2022)
197. Zhang, Y., Zhang, R., Potter, A.: Qed driven qaoa for network-flow optimization. Quantum **5**, 510 (2020)
198. Kirby, W.M., Love, P.: Variational quantum eigensolvers for sparse hamiltonians. Phys Rev Lett **127**(11), 110503 (2020)
199. Hatay, E.S., Golec, M., Golec, M., et al.: Quantum cloud computing: trends and challenges. J. Economy Technol. **1**, 1–11 (2024)
200. Du, Y., Hsieh, M.-H., Liu, T., You, S., Tao, D.: On the learnability of quantum neural networks. arXiv: Quantum Physics, (2020)
201. Zhao, C., Gao, X.-S.: Qdnn: deep neural networks with quantum layers. Quant. Mach. Intell. **3**, 1–9 (2021)
202. Nawaz, S.J., Sharma, S.K., Wyne, S., Patwary, M.N., Asaduzzaman, M.: Quantum machine learning for 6g communication networks: State-of-the-art and vision for the future. IEEE Access **7**, 46 317-46 350 (2019)
203. Bravyi, S., Kliesch, A., Koenig, R., Tang, E.: Hybrid quantum-classical algorithms for approximate graph coloring. Quantum **6**, 678 (2020)
204. Kalapothas, S., Flamis, G., Kitsos, P.: Efficient edge-ai application deployment for fpgas. Information **13**, 279 (2022)
205. Hu, Y., Liu, Y., Liu, Z.: A survey on convolutional neural network accelerators: Gpu, fpga and asic. 2022 14th International Conference on Computer Research and Development (ICCRD), pp. 100–107, (2022)
206. Vitale, A., Donati, E., Germann, R., Magno, M.: Neuromorphic edge computing for biomedical applications: gesture classification using emg signals. IEEE Sens. J. **22**, 19 490-19 499 (2022)
207. Li, Q., Ren, J., Pan, X., Zhou, Y., Zhang, Y.: Enigma: low-latency and privacy-preserving edge inference on heterogeneous neural network accelerators. 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), pp. 458–469, (2022)
208. Qiao, Y., Munir, M.S., Adhikary, A., Le, H.Q., Raha, A.D., Zhang, C., Hong, C.-S.: Mp-fedcl: Multi-prototype federated contrastive learning for edge intelligence. ArXiv, vol. arXiv:abs/2304.01950, (2023)
209. Zhong, J., Wu, Y., Ma, W., Deng, S., Zhou, H.: Optimizing multi-objective federated learning on non-iid data with improved nsga-iii and hierarchical clustering. Symmetry **14**, 1070 (2022)
210. Mu, X., Shen, Y., Cheng, K., Geng, X., Fu, J., Zhang, T., Zhang, Z.: Fedproc: prototypical contrastive federated learning on non-iid data. Future Gener. Comput. Syst. **143**, 93–104 (2021)
211. Jiao, L., Denoeux, T., Liu, Z., Pan, Q.: Egmm: an evidential version of the gaussian mixture model for clustering. ArXiv, vol. arXiv:abs/2010.01333, (2020)
212. Hummelholm, A.: Ai-based quantum-safe cybersecurity automation and orchestration for edge intelligence in future networks. European Conference on Cyber Warfare and Security, (2023). [Online]. Available: https://api.semanticscholar.org/CorpusID:259453663
213. Samriya, J.K., Kumar, M., Gill, S.S.: Secured data offloading using reinforcement learning and markov decision process in mobile edge computing. Int. J. Netw. Manage. **33**(5), e2243 (2023)
214. S.P. C, Jain, K., Krishnan, P.: Analysis of post-quantum cryptography for internet of things. 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 387–394, (2022)
215. Wang, L.-J., Zhang, K., Wang, J.-Y., Cheng, J., Yang, Y.-H., Tang, S.-B., Yan, D., tang, yan-lin, Liu, Z., Yu, Y., Zhang, Q., Pan, J.-W.: Experimental authentication of quantum key distribution with post-quantum cryptography. Npj Quant. Inform., vol. 7, (2020)
216. Yang, J., Baker, T., Gill, S.S., Yang, X., Han, W., Li, Y.: A federated learning attack method based on edge collaboration via cloud. Pract. Exp. Softw. **54**(7), 1257–1274 (2020)
217. Yigit, Y., Bal, B., Karameseoglu, A., Duong, T., Canberk, B.: Digital twin-enabled intelligent ddos detection mechanism for autonomous core networks. IEEE Commun. Stand. Magazine **6**, 38–44 (2022)
218. Ahmad, S.F., Ferjani, M.Y., Kasliwal, K.: Enhancing security in the industrial iot sector using quantum computing. 2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 1–5, (2021)
219. Badidi, E.: Edge ai for early detection of chronic diseases and the spread of infectious diseases: opportunities, challenges, and future directions. Future Int. **15**(11), 370 (2023)

Sukhpal Singh Gill (FHEA) is an Assistant Professor of Cloud Computing at the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. Dr. Gill is serving as an Editor-in-Chief for IGI Global IJAEC and Area Editor for Springer Cluster Computing Journal, also serving as an Associate Editor in IEEE IoT, Elsevier IoT, Wiley SPE, Wiley ETT and IET Networks Journals. He has co-authored 200+ peer-reviewed papers (with Citations 10100+ and H-index 51) and has published in prominent international journals and conferences such as IEEE TCC, IEEE TSC, IEEE TSUSC, IEEE COMST, IEEE TCE, ACM TOIT, IEEE TII, IEEE TNSM, IEEE IoT Journal, Elsevier JSS/FGCS, IEEE/ACM UCC and IEEE CCGRID. He has received several awards, including the Queen Mary University Education Excellence Award 2023, Outstanding Reviewer Award from IEEE IT Professional Magazine 2024, Elsevier Internet of Things Editor's Choice Award 2024, Elsevier Best Paper Award 2023, Distinguished Reviewer Award from SPE (Wiley), Best Paper Award AusPDC at ACSW 2021. He has edited and authored research various books for Elsevier, Springer and CRC Press. His research interests include Cloud Computing, Edge Computing, IoT and Energy Efficiency. For further information, please visit: http://www.ssgill.me.

Muhammed Golec is a PhD student in Computer Science at Queen Mary University of London (QMUL). Earlier, he graduated from QMUL in MSC Computer Science (Distinction) through the Ministry of Education Scholarship. He has published articles in prominent journals and conferences such as IEEE TII and Elsevier IoT, IEEE Consumer Electronics Magazine, and IEEE CCGRID. His research interests include Cloud Computing, Serverless Computing, AI, and Security and Privacy.

Jianmin Hu is currently a master's student at Shenzhen Advanced Technology Research Institute. He obtained a Bachelor's degree in Software Engineering from Harbin Institute of Technology in 2023. His current research direction is cloud native and system for large language model.

Minxian Xu (Senior Member, IEEE) is currently an Associate Professor at Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. He received the BSc degree in 2012 and the MSc degree in 2015, both in software engineering from University of Electronic Science and Technology of China. He obtained his Ph.D. degree from the University of Melbourne in 2019. His research interests include resource scheduling and optimization in cloud computing. He has co-authored 70+ peer-reviewed papers published in prominent international journals and conferences, such as ACM CSUR, IEEE TSC, IEEE TMC, ACM TOIT, ACM TAAS and ICSOC. He was awarded the 2023 IEEE TCSC Award for Excellence (Early Career Award). More information can be found at: minxianxu.info.

Junhui Du received the BSc degree in mathematics from the Nanjing University of Information Science Technology, China, in 2021. He is currently working toward the master's degree in mathematics with the Center for Applied Mathematics, Tianjin University, Tianjin, China. His research interests include Internet of Things, deep learning, and mobile edge computing.

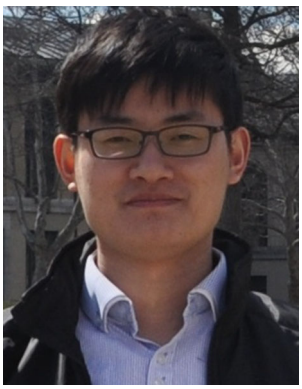
Huaming Wu (Senior Member, IEEE) received the B.E. and M.S. degrees from Harbin Institute of Technology, China in 2009 and 2011, respectively, both in electrical engineering. He received the Ph.D. degree with the highest honor in computer science at Freie Universität Berlin, Germany in 2015. He is currently a professor at the Center for Applied Mathematics, Tianjin University, China. His research interests include mobile cloud computing, edge computing, Internet of Things, deep learning, complex networks, and DNA storage.

Guneet Kaur Walia is a Ph.D scholar at the Department of Information Technology, Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, Punjab, India. She successfully completed her Masters in Computer Science Engineering at Punjab Agricultural University, Ludhiana, Punjab, in 2016. Her dedication and enthusiasm for exploring various cutting-edge technologies make her a passionate researcher. She has published in prominent international journals including IEEE TCE and IEEE COMST. Her research interests include Cloud Computing, Edge Computing, Internet of Things (IoT), Resource Management in Edge Computing, and Artificial Intelligence (AI).

Subramaniam Subramanian Murugesan is a PhD student in Electronic Engineering at Queen Mary University of London (QMUL), funded by a UKRI EPSRC Doctoral Training Partnership (DTP) studentship. He holds a master's degree in Big Data Science from QMUL. His research focuses on RF emission detection, identification, and localization using Software-Defined Radio (SDR), advanced antennas, AI/ML/DL applications, Cloud & IoT, software engineering, and edge AI technologies. He has published work in journals such as the IEEE Journal of Biomedical and Health Informatics (JBHI), Cluster Computing (Springer).

Babar Ali is a PhD student in the School of Electronic Engineering and Computer Science at Queen Mary University of London (QMUL). He has published PhD findings in journals such as Wiley International Journal of Network Management and Elsevier Internet of Things. His research interests include Cloud Computing, Fog Computing, IoT, Edge Computing, and Wireless Sensor Networks.

Mohit Kumar is Assistant Professor in the Department of Information Technology at Dr. B R Ambedkar National Institute of Technology, Jalandhar, India. He received his Ph.D. degree from Indian Institute of Technology Roorkee in the field of Cloud Computing, 2018, and M. Tech degree in Computer Science and Engineering from ABV-Indian Institute of Information Technology Gwalior, India in 2013. His research topics cover the areas of Cloud computing, Fog/ Edge Computing, Internet of Things, federated learning, Blockchain, and Artificial Intelligence. He has published more than 40 research articles in reputed journals, IEEE Transactions and international conferences. He has been Session chair and keynotes Speaker of many International conferences, webinars, FDP, STC in India. He has guided six M. Tech Thesis and supervising Ph.D. Scholar. He is an active reviewer of several reputed journals and international conferences. He is a member of the IEEE.



Kejiang Ye is currently a Professor and the Director of the Research Center for Cloud Computing, Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences (CAS). He received his B.S and Ph.D degree both from Zhejiang University and was a Post Doctoral Research Associate at Carnegie Mellon University (CMU). His research interests include Digital Technology and Systems (e.g., Cloud Computing, Big Data and Industrial Internet). He is a Senior Member of IEEE and a Distinguished Member of China Computer Federation (CCF).

Prabal Verma is currently working as Assistant Professor in Information Technology Department, National Institute of Technology (NIT), Srinagar, Jammu and Kashmir India. He also worked as Assistant Professor in the department of Computer Science and Engineering at Thapar Institute of Technology, Patiala . He did his Doctoral degree in Computer Science and Engineering from Guru Nanak Dev University, Amritsar. His work is published in highly reputed publishers like IEEE, Elsevier, Wiley, Taylor and Francis, and Springer. His current working research areas include Internet of Things (IoT) in Healthcare, Big Data and Fog-Cloud computing. He is also IEEE Member.

Surendra Kumar is an Assistant Professor at GLA University in Mathura, Uttar Pradesh, India. He has a strong academic background, having received his Master of Computer Application (M. C. A.) from Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, Uttar Pradesh, India in 2013. He continued his academic pursuits at the same university, earning his Doctor of Philosophy in the Department of Computer Science. He was recently recognized with an International Distinguished Young Researcher Award for 2021-22 from the International Institute of Organized Research. Dr. Kumar has also published several research articles in reputed journals, conferences, and book chapters. Dr. Kumar's academic achievements and research contributions demonstrate his dedication to his field of study and his commitment to advancing knowledge and innovation in the areas of resource management, cloud security, cryptography, and distributed computing, blockchain technology.

Felix Cuadrado received a Ph.D. degree in telecommunications engineering from the Universidad Politécnica de Madrid (UPM), Spain, in 2009. He is currently a Senior Distinguished Fellow (Beatriz Galindo scheme) with the Universidad Politécnica de Madrid, a Visiting Reader at the Queen Mary University of London, and a fellow of the Alan Turing Institute. He has numerous publications in top-tier journals and conferences, including IEEE TSC, IEEE TCC, Elsevier JSS, Elsevier FCGS, IEEE ICDCS, and WWW. His research explores the challenges arising from large-scale data-intensive applications through a combination of software engineering, distributed systems, and mathematical approaches.

Steve Uhlig obtained a Ph.D. degree in Applied Sciences from the University of Louvain, Belgium, in 2004. Prior to joining Queen Mary, he was a Senior Research Scientist with Technische Universität Berlin/Deutsche Telekom Laboratories, Berlin, Germany. Starting in January 2012, he is the Professor of Networks and Head of the Networks Research group at Queen Mary, University of London. Between 2012 and 2016, he was a guest professor at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Current Research interests: Internet measurements, software-defined networking, content delivery.

Authors and Affiliations

Sukhpal Singh Gill[1] · Muhammed Golec[1,2] · Jianmin Hu[3] · Minxian Xu[3] · Junhui Du[4] · Huaming Wu[4] · Guneet Kaur Walia[5] · Subramaniam Subramanian Murugesan[1] · Babar Ali[1] · Mohit Kumar[5] · Kejiang Ye[3] · Prabal Verma[8] · Surendra Kumar[6] · Felix Cuadrado[7] · Steve Uhlig[1]

✉ Muhammed Golec
m.golec@qmul.ac.uk

Sukhpal Singh Gill
s.s.gill@qmul.ac.uk

Jianmin Hu
jm.hu@siat.ac.cn

Minxian Xu
mx.xu@siat.ac.cn

Junhui Du
dujunhui_0325@tju.edu.cn

Huaming Wu
whming@tju.edu.cn

Guneet Kaur Walia
guneetkw.it.22@nitj.ac.in

Subramaniam Subramanian Murugesan
s.subramanianmurugesan@qmul.ac.uk

Babar Ali
b.ali@qmul.ac.uk

Mohit Kumar
kumarmohit@nitj.ac.in

Kejiang Ye
kj.ye@siat.ac.cn

Prabal Verma
prabal.verma@nitsri.ac.in

Surendra Kumar
surendra.kumar@gla.ac.in

Felix Cuadrado
felix.cuadrado@upm.es

Steve Uhlig
steve.uhlig@qmul.ac.uk

[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

[2] Abdullah Gul University, Kayseri, Turkey

[3] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

[4] Center for Applied Mathematics, Tianjin University, Tianjin, China

[5] Department of Information Technology, Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, India

[6] Department of Computer Engineering and Applications, GLA University, Mathura, India

[7] Technical University of Madrid (UPM), Madrid, Spain

[8] Department of Information Technology, National Institute of Technology, Srinagar, India