Telematics and Informatics Reports 13 (2024) 100116

Contents lists available at ScienceDirect

Telematics and Informatics Reports

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

# Modern computing: Vision and challenges

Sukhpal Singh Gill [a,*], Huaming Wu [b], Panos Patros [c], Carlo Ottaviani [d], Priyansh Arora [e], Victor Casamayor Pujol [f], David Haunschild [g], Ajith Kumar Parlikad [h], Oktay Cetinkaya [i], Hanan Lutfiyya [j], Vlado Stankovski [k], Ruidong Li [l], Yuemin Ding [m], Junaid Qadir [n], Ajith Abraham [o,p], Soumya K. Ghosh [q], Houbing Herbert Song [r], Rizos Sakellariou [s], Omer Rana [t], Joel J.P.C. Rodrigues [u,v], Salil S. Kanhere [w], Schahram Dustdar [f], Steve Uhlig [a], Kotagiri Ramamohanarao [x], Rajkumar Buyya [y]

[a] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK
[b] Center for Applied Mathematics, Tianjin University, Tianjin, China
[c] Raygun Performance Monitoring, Wellington, New Zealand
[d] Department of Computer Science and York Centre for Quantum Technologies, University of York, York, UK
[e] Microsoft, Schiphol, Netherlands
[f] Distributed Systems Group, Vienna University of Technology, Vienna, Austria
[g] Detecon International GmbH, Munich, Germany
[h] Institute for Manufacturing, Department of Engineering, University of Cambridge, Cambridge, UK
[i] Oxford e-Research Centre (OeRC), Department of Engineering Science, University of Oxford, Oxford, UK
[j] Department of Computer Science, University of Western Ontario, London, Canada
[k] Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
[l] Institute of Science and Engineering, Kanazawa University, Japan
[m] Tecnun School of Engineering, University of Navarra, Spain
[n] Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar
[o] Bennett University, Greater Noida, India
[p] Machine Intelligence Research Labs, Auburn, WA, USA
[q] Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India
[r] Department of Information Systems University of Maryland, Baltimore County (UMBC), Baltimore, USA
[s] Department of Computer Science, University of Manchester, Oxford Road, Manchester, UK
[t] School of Computer Science and Informatics, Cardiff University, Cardiff, UK
[u] Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[v] COPELABS, Lusófona University, Lisbon, Portugal
[w] School of Computer Science and Engineering, The University of New South Wales (UNSW), Sydney, Australia
[x] Retired Professor, The University of Melbourne, Victoria, Australia
[y] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia

## ARTICLE INFO

*Keywords:*
Modern computing
Edge AI
Edge computing
Artificial Intelligence
Machine learning
Cloud computing

## ABSTRACT

Over the past six decades, the computing systems field has experienced significant transformations, profoundly impacting society with transformational developments, such as the Internet and the commodification of computing. Underpinned by technological advancements, computer systems, far from being static, have been continuously evolving and adapting to cover multifaceted societal niches. This has led to new paradigms such as cloud, fog, edge computing, and the Internet of Things (IoT), which offer fresh economic and creative opportunities. Nevertheless, this rapid change poses complex research challenges, especially in maximizing

* Correspondence to: School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, UK.
*E-mail addresses:* s.s.gill@qmul.ac.uk (S.S. Gill), whming@tju.edu.cn (H. Wu), panos@raygun.com (P. Patros), carlo.ottaviani@york.ac.uk (C. Ottaviani), priyansh.arora@microsoft.com (P. Arora), v.casamayor@dsg.tuwien.ac.at (V.C. Pujol), david.haunschild@detecon.com (D. Haunschild), aknp2@cam.ac.uk (A.K. Parlikad), oktay.cetinkaya@eng.ox.ac.uk (O. Cetinkaya), hanan@csd.uwo.ca (H. Lutfiyya), vlado.stankovski@fri.uni-lj.si (V. Stankovski), lrd@se.kanazawa-u.ac.jp (R. Li), yueminding@tecnun.es (Y. Ding), jqadir@qu.edu.qa (J. Qadir), ajith.abraham@ieee.org (A. Abraham), skg@cse.iitkgp.ac.in (S.K. Ghosh), songh@umbc.edu (H.H. Song), rizos@manchester.ac.uk (R. Sakellariou), ranaof@cardiff.ac.uk (O. Rana), joeljr@ieee.org (J.J.P.C. Rodrigues), salil.kanhere@unsw.edu.au (S.S. Kanhere), dustdar@dsg.tuwien.ac.at (S. Dustdar), steve.uhlig@qmul.ac.uk (S. Uhlig), rkotagiri@gmail.com (K. Ramamohanarao), rbuyya@unimelb.edu.au (R. Buyya).

https://doi.org/10.1016/j.teler.2024.100116
Received 20 November 2023; Received in revised form 4 January 2024; Accepted 5 January 2024
Available online 8 January 2024
2772-5030/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Quantum computing
Computing

potential and enhancing functionality. As such, to maintain an economical level of performance that meets ever-tighter requirements, one must understand the drivers of new model emergence and expansion, and how contemporary challenges differ from past ones. To that end, this article investigates and assesses the factors influencing the evolution of computing systems, covering established systems and architectures as well as newer developments, such as serverless computing, quantum computing, and on-device AI on edge devices. Trends emerge when one traces technological trajectory, which includes the rapid obsolescence of frameworks due to business and technical constraints, a move towards specialized systems and models, and varying approaches to centralized and decentralized control. This comprehensive review of modern computing systems looks ahead to the future of research in the field, highlighting key challenges and emerging trends, and underscoring their importance in cost-effectively driving technological progress.

## 1. Introduction

The Internet, the expansive computational backbone of interactive machines, is largely responsible for the 21st century's social, financial, and technological growth [1]. The growing reliance on the computing resources it encapsulates has pushed the complexity and scope of such platforms, leading to the development of innovative computing systems. These systems have genuinely improved the capabilities and expectations of computing equipment driven by rapid technical and user-driven evolution [2]. For instance, vintage mainframes combined centralized data processing and storage with transmission interfaces for user input. Due to advancements in clusters and packet-switching technologies, microchip gadgets, and Graphical User Interfaces (GUIs), technology originally shifted from big, centrally-run mainframe computers to Personal Computers (PCs). The globalization of network standards made it possible for interconnected networks worldwide to communicate and share data [3]. Businesses slowly combined sensor and actuator goals with built-in network connectivity by creating architectures and standards that submit tasks to remote pools of computing resources, such as memory, storage, and data processing [4]. As a result, newer models like the Internet of Things (IoT) and edge computing are now beginning to expand the reach of technology outside the confines of traditional network nodes [5].

Over the past six decades, computing models have fundamentally shifted to address the problems posed by the ever-evolving nature of our civilization and its associated computer system architectures [6]. The evolution of computing from mainframes to workstations to the cloud to autonomous and decentralized architectures, such as edge computing and IoT technologies, however, maintains identical core parts and traits that characterize their function [7]. Research in computing underpins all of them! Advancements in areas like security, computer hardware acceleration, edge computing, and energy efficiency typically serve as catalysts for innovation and entrepreneurship that span across various business domains [8]. While computing systems and other forms of system integration create new problems/opportunities, software frameworks have been developed to address them. Thus, middleware, network protocols, and safe segregation techniques must be continually developed and refined to support novel computing systems—and their innovative use cases.

### 1.1. Motivation

By tracking the effect of computing systems on the community, this comprehensive study seeks to (a) establish the essential features and components of modern computing systems, (b) thoroughly assess the development of innovations and behavioral patterns that inspired the invention of these paradigms, and (c) recognize significant developments throughout the models, such as the integration of system design, the shifting between centralization and decentralization, and lags in model conceptualization and development.

This investigation suggests that next-generation computing systems will facilitate the decentralization of computational services. This will be achieved via the composition of decentralized calculation tools with workload-specific targets for performance to create dramatically more complex structures. These will satisfy holistic operational demands, such as improved capacity and power accessibility.

### 1.2. Related surveys and our contributions

Computing being a rapidly growing topic, the time is right for a novel, forward-thinking study to summarize, improve, and integrate the existing and newly-generated information, and to explore possible trends and future viewpoints. Previously, Pujol et al. [9] provided a survey on distributed computing continuum systems that focused on business models. Further back in 2018, Buyya et al. [1] presented a manifesto on fundamental issues, developments, and impacts in cloud computing research. Meanwhile, Gill et al. [4] offered a visionary survey of advances in computing paradigms for fog, edge, and serverless computing. Further, Shalf [10] summarized the 2020 state of the art of technological roadmaps and their implications for the future of systems, including what a post-exascale system would entail. Finally, in 2021, Angel et al. [11] reviewed leading computational frameworks for cloud and edge computing, and showcased breakthroughs that had been brought about via the merging of Machine Learning (ML) with these models.

In order to evaluate and identify the most pressing research issues of modern computing, we have developed the very first taxonomy of its type. We performed a gap analysis of the current surveys using several criteria, as shown in Table 1, which underpinned the design of our work. Hence, our study uniquely contributes by (a) exploring the history of computing paradigm shifts with a focus on technology drivers, (b) providing a thorough taxonomy of computing systems, (c) introducing the hype cycle for modern computing systems with a focus on new trends, and (d) discussing the effects and cost-effective performance requirements of modern computing.

The **key contributions** of this article are summarized as follows:

- It offers a concise overview of the transition from early to modern computing.
- The study explores the evolution of computing paradigms, focusing on technological drivers (1960–2023).
- Following a novel methodology, the article produces a taxonomy of modern computing based on traits of computing such as (1) focus or paradigms; (2) technologies or impact areas; and (3) trends or observations.
- It presents a comprehensive classification of computing: (1) Standalone vs. Networked Computing; (2) General Purpose vs. Specialized Computing, (3) Centralized vs. Decentralized Computing, (4) Computing Trends and Emerging Technologies; and (5) Computational Methodologies: Parallel vs. Sequential Computing.
- The study identifies the impact and performance criteria of modern computing in terms of performance metrics, efficiency metrics, social impact, security and compliance, and economics and management.
- It provides an in-depth summary of computing traits and resources for further research.
- The article identifies open challenges and research directions for the traits of computing.
- Finally, it introduces the hype cycle for modern computing systems, spotlighting emerging trends.

**Table 1**
Comparison of this work with existing studies.

| Work | | [9] | [1] | [4] | [10] | [11] | Our Work |
|---|---|---|---|---|---|---|---|
| Year | | 2023 | 2018 | 2022 | 2020 | 2021 | 2024 |
| A Taxonomy of Modern Computing | | | | | | | ✓ |
| Evolution of Computing Paradigms (1960 to 2023) | | | | | | | ✓ |
| Classification of Computing | Standalone vs. Networked Computing | | | | | | ✓ |
| | General Purpose vs. Specialized Computing | | | | | | ✓ |
| | Centralized vs. Decentralized Computing | | | | | | ✓ |
| | Computing Trends and Emerging Technologies | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Computational Methodologies: Parallel vs. Sequential Computing | | | | | | ✓ |
| Traits of Computing | Focus/ Paradigms | | ✓ | | | | ✓ |
| | Technologies/ Impact Areas | | | | | | ✓ |
| | Trends/ Observations | | | | | | ✓ |
| Impact and Performance Criteria | Performance Metrics | | | | | | ✓ |
| | Efficiency Metrics | | | | | | ✓ |
| | Social Impact | | | | | | ✓ |
| | Security and Compliance | | | | | | ✓ |
| | Economic and Management | | | | | | ✓ |
| Open Challenges and Future Directions | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Emerging Trends in Modern Computing: Hype Cycle | | | | | | | ✓ |

### 1.3. Article organization

The article is organized as follows: Section 2 offers a concise overview of the transition from early to modern computing. Section 3 explores the evolution of computing paradigms, focusing on technological drivers. Section 4 presents a classification of computing systems, and Section 5 examines the impact and performance criteria in modern computing. The article concludes in Section 7, summarizing computing-related technologies and trends through a hype cycle in Section 6. The list of acronyms used in this study is given in Appendix.

## 2. Early computing to modern computing: A vision

Over the last six decades, advancements in computing systems have optimized the efficiency of the available hardware [12]. Over this time period, novel computing models and innovations have been developed and replaced the previous state-of-the-art, all of which incrementally contribute to the current technology status [2]. Fig. 1 shows the transition from early computing to contemporary computing. Originally, a single system could only carry out a single task; hence, a user needed various systems working in tandem to achieve their desired tasks. However, to safely share information between computers – in order to overcome the problem of executing only one task at a time – a reliable communication mechanism is essential [13]. To that end, our investigation unfolds across three key sections: Section 3 delves into the evolution of computing paradigms, emphasizing technological drivers. Section 4 offers a comprehensive classification of computing systems. The discussion in Section 5 revolves around the impact and performance criteria of modern computing. Section 6 introduces the hype cycle for modern computing systems, spotlighting emerging trends.

## 3. Evolution of computing paradigms: Technological drivers

Fig. 1 illustrates the progression of computing technology starting from the year 1960.

### 3.1. Client server

In the year 1960, a centralized platform (a.k.a, distribution integration) was developed to share workloads (a.k.a., jobs) between the resource providers (i.e., server instances) and service consumers (i.e., customers) [12]. Supporting it, a networking system was utilized for communications between client devices and servers, and servers exchange resources for customers to perform their tasks using a load balancing mechanism [14]. Illustrative examples of the client–server model's application include the Email and the World Wide Web (WWW). However, users in this configuration were unable to freely interact with one another.

### 3.2. Supercomputer

A supercomputer is a powerful computer with extraordinary processing capability, such that it can handle complex calculations in several areas of science, including climate study, quantum physics, and molecular simulation [15]. Energy utilization and heat control in supercomputers endured as a key research problem throughout their growth in the 1960s [16]. Supercomputers, such as Multivac, HAL-9000, and Machine Stops, have been instrumental in underpinning/enabling dramatic technological advancements [14].

### 3.3. Proprietary mainframe

To handle massive amounts of data (including dealing with transactions, customer data analysis, and censuses), a high-speed machine with large computing power is required [17]. Virtualization on mainframes allows for increased efficiency, protection, and dependability. In the year 2017, IBM announced the newest version of its mainframe, the IBM z14 [13]. Being built to support massive economic activity and despite their high price tag, mainframe computers deliver outstanding efficiency [14].

### 3.4. Cluster computing

Cluster computing is a method of increasing the efficiency of a computing system by utilizing several nodes to complete a single operation [18]. In order to coordinate various computing nodes, this type of technology requires a rapid Local Area Network (LAN) for exchanging information among them [19].

### 3.5. Home PCs

The early days of the Internet coincided with the flourishing of PC kept at one's home [3]. The Internet was evolving into a foundational network, connecting local networks to the larger Internet using self-adaptive network protocols, such as Transmission Control Protocol/Internet Protocol (TCP/IP)—in contrast to the original Network Control Protocol (NCP)-based Advanced Research Projects Agency Network (ARPANET) mechanisms [2]. As a result, there was a sharp

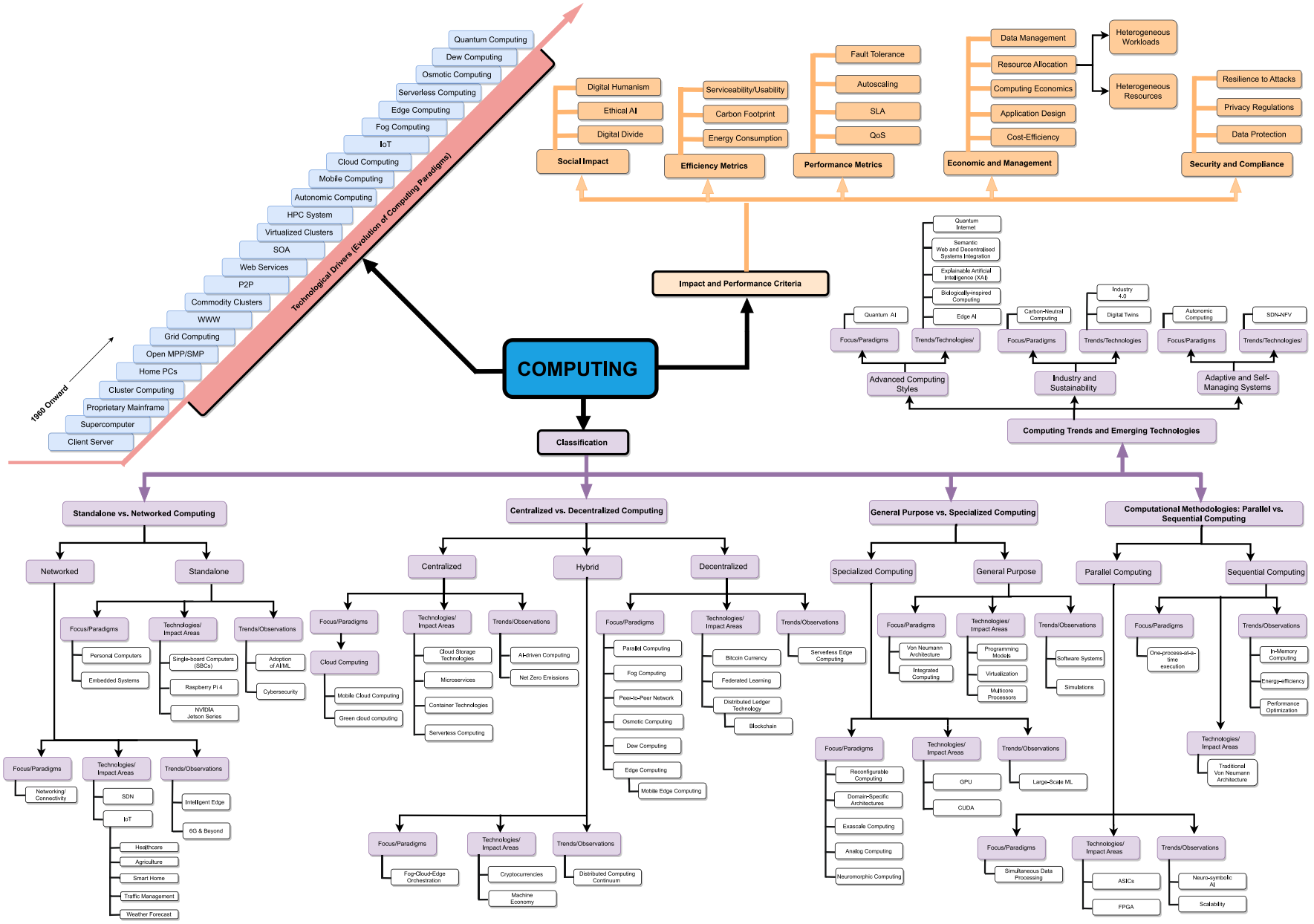


**Fig. 1.** Modern computing: A taxonomy.

increase in the number of hosts on the Internet, which quickly overwhelmed centralized naming technologies like *HOSTS.TXT*. In the year 1985, the earliest publicly available version of a Domain Name System (DNS) was released for the Unix BIND system [20]. This system translates hostnames into IP addresses. Pioneer Windows, Icons, Menus, and Pointers (WIMP)-based GUIs on computers, such as the Xerox Star and the Apple LISA, proved that customers could successfully use machines in their homes for tasks like playing video games and surfing the Internet [21].

### 3.6. Open MPP/SMP

Massive Parallel Processing (MPP) and Symmetric Multi-Processing (SMP) systems are the two most common forms of parallel computing platforms [16]. In an SMP setup, multiple processors run the same Operating System (OS) concurrently while sharing the rest of the hardware's capacity (e.g., disc space and RAM). Naturally, resource pooling influences the computational speed of completing a given assignment. In an MPP scenario, the file system can be shared, while no other resources are pooled for use during task processing [14]. Incorporating more machines $N$ and their associated storage and RAM space, increases the ability to scale according to the Universal Scalability Law (an extension of Amdahl's Law), assuming $\kappa$ the proportion of work that can be parallelized and $\sigma$ the interprocess communication penalty:

$$\text{Capacity} = \frac{N}{1 + \sigma \cdot (N-1) + \kappa \cdot N \cdot (N-1)}. \quad (1)$$

### 3.7. Grid computing

This technology enables a group to work together towards the same objective by executing non-interactive, and largely IO-intensive tasks [19]. Each application running on only one grid is a top priority [12]. In addition to allocating and managing resources, grid computing also offers a reliable architecture, as well as tracking and exploration support.

### 3.8. WWW

The primary web browsers, websites, and web servers all came into existence in the later stages of the 1980s and early 1990s, underpinned by the development of Hyper Text Transport Protocol (HTTP) and Hyper Text Markup Language (HTML) [2]. The platform for the interconnected system of networks that makes up the WWW was made possible by the standardizing technology of TCP/IP network protocols. This allowed for a dramatic increase in the total number of servers linked to the Web and introduced Information Technology (IT) to the general public. Software applications were thus able to communicate with one another beyond address spaces and networking, e.g., via novel technologies like Remote Procedure Calls (RPCs) [22].

### 3.9. Commodity clusters

Commodity cluster computing employs several computers simultaneously, which can inexpensively execute user tasks [19]. In an effort to standardize their processes, several companies use open standards while building commodity computers [14]. This allowed immediate computing business needs to be met using ready-made processors.

### 3.10. Peer to peer (P2P)

P2P is a distributed framework to share workloads or jobs amongst multiple peers; alternatively, computers and peers may interact with one another openly at the application layer [23]. With no mediator in the center, users of a peer-to-peer system can share resources like memory, CPU speed, and storage space. Peer-to-peer communication utilizes the TCP/IP protocol suite [24]. Interactive media, sharing file infrastructure, and content distribution are some of the most common use cases for P2P technology.

### 3.11. Web services

The technology supporting web services enables the exchange of data between various Internet-connected devices in machine-under standable data formats, such as JavaScript Object Notation (JSON) and Extensible Markup Language (XML), over the WWW [25]. Commonly, web-based services operate as a connection between end users and database servers.

### 3.12. Service-Oriented Architecture (SOA)

The SOA paradigm enables software elements to be reused and made compatible through advertised service designs/Application Programming Interfaces (APIs) [26]. It is normally easier to include services in new apps: the apps can be architected to adhere to standardized protocols and leverage consistent design patterns. This frees the software engineer from the burden of recreating or duplicating current features or figuring out how to link to and interoperate with current systems—e.g., via using Software Development Kits (SDKs) that implement common functionalities, such as networking, retries, marshaling of data and error handling [27]. Each SOA API exposes the logic and data necessary to carry out a single, self-contained business operation (such as vetting the creditworthiness of a client, determining the loan's due date, or handling an insurance application) [28]. The loose integration provided by the service's design allows for the service to be invoked with limited knowledge of the underlying service implementation.

### 3.13. Virtualized clusters

Virtualization enables a guest computer system to be implemented on top of a host computer system, which abstracts away the problem of physically supporting and maintaining multiple types/architectures of physical machines [19]. With a virtualized cluster, several Virtual Machines (VMs) may pool their resources to complete a single job. VM hypervisors, which execute the guest system on the host system, allow software-based virtualization to run either on top of an OS or directly (bare-metal) on hardware [14]. Costs and complexity are reduced, and a greater number of tasks may be completed with identical hardware by adopting a VM-based system.

### 3.14. High Performance Computing (HPC) system

HPC is the computing method of choice when dealing with computationally intensive issues, which tend to arise in the domains of commerce, technology, and research [14,19]. A scheduler in an HPC system manages accessibility to the various computing resources available for use in solving various issues [29]. HPC systems utilize a pooled set of resources, allowing them to perform workloads or tasks via the allocation of concurrent resources and online utilization of various resources.

### 3.15. Autonomic computing

One of the first global initiatives to build computer systems with minimum human involvement to achieve preset goals was IBM's autonomic computing program in 2006 [30]. It was mostly based on research on nerves, thinking, and coordination. Autonomic computing research examines how software-intensive systems may make choices and behave autonomously to achieve user-specified goals [4]. Control for closed- and open-loop systems has shaped autonomic computing [31]. Complex systems can have several separate control networks.

### 3.16. Mobile computing

The term "mobile computing" is used to describe a wide range of IT components that give consumers mobility in their usage of computation, information, and associated equipment and capabilities [32]. An especially popular definition of "mobile" is accessing information while moving, when an individual is not confined to a fixed place. Accessibility at a fixed spot may also be thought of as mobile, especially if it is provided by hardware that consumers can move as needed but that remains in one place while functioning [33]. Mobile computing devices are becoming essential across industries, boosting efficiency and creativity in fields such as healthcare, retail, manufacturing, and the arts.

### 3.17. Cloud computing

Software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) are all examples of Internet-accessible web services [1]. Google Mail is an excellent instance of a SaaS product since it provides a wide range of useful features without the burden of installation and ongoing upkeep costs. PaaS providers like Microsoft provide a scalable environment where users can install their applications [34]. Amazon is a prime instance of an IaaS provider since it provides users with access to servers, networks, storage, and other hardware components necessary to run applications and other workloads efficiently and effectively. Using distant facilities for performing user operations (processing, administration, and storage of data) over the Internet is known as "cloud computing", abbreviated as "XaaS", where X = "I", "P", "S", etc. Cloud computing enables the pooling of resources to reduce execution costs and enhance service accessibility [35]. There are four major types of cloud computing systems: public, private, hybrid, and communal. Dependability, safety, and cost-effectiveness are just a few examples of Quality of service (QoS) characteristics that should be considered while developing a successful cloud service.

### 3.18. IoT

Controllers, gadgets, and detection devices are all examples of IoT devices that can communicate with one another over the WWW [5]. IoT has many potential uses in many different areas, including farming, medical treatment, climate prediction, logistics, home automation, and industrial automation [36].

### 3.19. Fog computing

This cutting-edge design makes extensive use of mobile devices, also known as fog nodes, which are utilized for data storage and processing, and rely on the web for inter-node connectivity [37]. The data plane and the control plane are the two main components of fog computing [38]. Although the control layer is a gateway component and determines the network's layout, the data plane offers capabilities at the network's edge to decrease delay and boost QoS [39]. Fog computing supports IoT gadgets such as smartphones, detectors, and health monitors.

### 3.20. Edge computing

Edge computing is a method that delegates processing to dispersed edge devices for data processing and information exchange [40]. In addition, edge computing enhances QoS, decreases delay, and lowers transmitting expenses by computing huge volumes of data on gadgets at the edge rather than in the public cloud [41]. However, edge computing relies on a constantly available web connection to perform certain tasks in a timely manner, so it is best used for applications that can execute autonomously without centralized control for prolonged periods of time [42].

### 3.21. Serverless computing

The serverless computing paradigm eliminates the need to manage servers and other infrastructure components [43] centrally. Since serverless computing eliminates the need for software engineers to manage servers, it is expected to grow much faster. With serverless computing, hosting companies may easily handle infrastructure management and automatic provisioning [44]. Because of this, less effort and resources are needed to oversee the infrastructure.

### 3.22. Osmotic computing

Osmotic computing is a growing idea that merges IoT, cloud, fog, and edge technology for the constantly changing administration of IT services. The dramatic increase in the size of resources in the network's periphery is the primary force behind this trend. By defining, creating, and implementing a computing model, this paradigm focuses on methods to improve edge and cloud-based IoT services [45]. To manage resources and resolve data difficulties in IoT and data science, osmotic computing applies the fundamental concepts of the osmosis phenomenon in chemistry [46]. The primary objective of this computing model is to distribute workloads and efficiently use available resources among servers without degrading service delivery or efficiency.

### 3.23. Dew computing

Dew computing is "a software-hardware organization model for computers situated in the cloud computing environment", where a local machine complements and operates independently of cloud services [47]. Dew computing may bridge the gap between cloud and on-premises computing. Data and services stored in the cloud are accessible regardless of an Internet connection. The need for constant Internet access is the primary restriction on cloud and fog computing. Complementing fog and edge computing with considerable Internet reliance, an extra layer, including dew computing, is necessary to keep apps and services alive and functioning. Even if dew computing is not conducted entirely online, it nevertheless uses cloud computing and depends on collaboration for data and operations, for example, One Drive [48].

### 3.24. Quantum computing

Quantum computing is a radically different way to analyze knowledge and data. Several possibilities can be taken advantage of when processing information stored in the quantum states of quantum machines that are unavailable when analyzing information in a conventional fashion [49]. The phenomena of quantum entanglement and superposition are two such examples. Because of quantum entanglement, it is difficult to offer a comprehensive description from the understanding of merely the component states, which is a defining characteristic of quantum systems. One definition of the term "superposition" is the potential of merging quantum states to create a new valid quantum state [50]. The primary purpose driving the effort to construct a quantum computer was the modeling of quantum systems; however, it was not until the identification of quantum algorithms capable of achieving realistic objectives that the enthusiasm for constructing such devices began to garner increasing scrutiny [51].

## 4. Classification of computing: Paradigms, technologies and trends

In this section, we discuss the different types of computing and classify them into different broad categories as shown in Fig. 1. Table 2 briefly describes traits of computing that are used in this classification such as (1) focus or paradigms; (2) technologies or impact areas; and (3) trends or observations.

**Table 2**
Summary of computing traits.

| Trait | Description |
|---|---|
| Focus/Paradigms | We discuss well-established computing paradigms, from client–server to quantum computing, which have been explored in the last decade. |
| Technologies/Impact Areas | We cover key research that has grown over time by utilizing these well-established computing paradigms and how this has led to many breakthroughs in the underlying technology. |
| Trends/Observations | The new trends, such as large-scale machine learning, digital twins, edge AI, bitcoin currency, 6G & Beyond and quantum Internet and biologically-inspired computing, for the next generation of computing, have come to light due to these advances in computing paradigms and technology. |

### 4.1. Standalone vs. Networked computing

Standalone computing occurs when a computer is not connected to another computer in any way, whether through wired or WiFi connections [52]. Multiple computers linked together form a network, a model that falls under networked computing.

#### 4.1.1. Standalone computing

In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within standalone computing.

*4.1.1.1. Focus/paradigms.* The following are the main focus or paradigms for standalone computing:

(1) *PCs*: Individuals use PCs, which leverage microprocessors designed for personal use. Before the PC, businesses had to operate computers by connecting several users' terminals to a separate, massive mainframe system [3]. By the end of the 1980s, technical developments had enabled the construction of a compact computer that a person could purchase and use as a word processor or for various computing objectives [2].
(2) *Embedded Systems*: A computer (often a microcontroller or microprocessor) is built into (i.e., embedded in) the design of a device [53]. Most of the time, an individual does not even realize they are using a computer because there might not be any obvious hints of applications, data, or software [54]. The software that operates a microwave oven or an engine control unit of a contemporary vehicle are two instances of items with undetectable integrated systems.

*4.1.1.2. Technologies/impact areas.* The key technologies and affected domains for standalone computing include:

(1) *Single-board Computers (SBCs)*: In an SBC, the CPU, I/O, memory, and various other components are all housed on one integrated circuit board; the quantity of memory is fixed; and there are no slots to be expanded for additional hardware [55].
(2) *Raspberry Pi 4*: The Raspberry Pi is a family of tiny SBCs that have been developed to allow programming and computing capabilities to be available to all. The Raspberry Pi Model B became the inaugural board produced by the foundation behind the Raspberry Pi [55]. Due to its immense popularity, other variants have subsequently been developed, each with its own set of advantages. These include the Raspberry Pi computation component, which has been optimized for use in embedded systems [56].
(3) *NVIDIA Jetson Series*: This is a line of Graphics Processing Units (GPUs) that includes the initial processors built with the explicit purpose of powering self-driving robots [57]. With up to 32 Tera Operations Per Second (TOPS) of Artificial Intelligence (AI) efficiency, these GPUs efficiently handle optical measurements, sensor fusion, positioning, visualization, obstacle detection, and path-planning, all of which are essential for the development of robotics [55]. The Jetson Xavier series focuses on creating specialized robots and edge robots, with several distinct manufacturing components.

*4.1.1.3. Trends/observations.* The main trends and observations regarding standalone computing are:

(1) *Adoption of AI/ML*: NVIDIA Jetson Nano, for instance, enables consumers to equip billions of low-power AI/ML systems with remarkable new features [58]. It paves the way for a wide variety of integrated IoT services, such as low-cost Network Video Recorders (NVRs), consumer automation, and analytics-rich gateways [55]. With its ready-to-try applications and enthusiastic software developer community, Jetson Nano serves as the ideal tool for beginning students to gain knowledge about AI and robotics in real-life situations.
(2) *Cybersecurity*: Embedded systems are compact, specifically designed devices built to carry out a single task, frequently in real-time, while using as few resources as possible [54]. Installing protective measures on these platforms to guard against dangers like unauthorized usage or fraudulent attacks drives the need for embedded security [59]. These safeguards are included in electrical components, firmware, and applications to achieve an all-encompassing defense.

#### 4.1.2. Networked computing

In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within networked computing.

*4.1.2.1. Focus/paradigms.* The following are the main focus or paradigms for networked computing:

(1) *Networking/Connectivity*: Servers in cloud computing underpin the services and APIs provided to internal and external clients. Communication on several levels, both inside and among data centers, is essential for effectively implementing cloud services [1]. Crucially, networking ensures that all parts can talk to one another in a safe, frictionless, effective, and adaptable way.
Many developments and studies in networking during the past ten years have focused on the cloud [60]. For instance, Software-Defined Networking (SDN) and Network Function Virtualization (NFV) aim to construct adaptable, versatile, and programmable computer networks to lessen the financial and time commitments of cloud service providers [61]. Scalability challenges have spurred several current developments in network design for the Cloud Data Centers (CDCs), as well as the necessity for a flat addressing space, and the excess demand for machines. Notwithstanding these developments, numerous networking issues require a resolution. The excessive energy consumption of modern CDCs is a major issue [60]. Especially because it is a common practice in data centers to have all networking equipment active at all times.
Furthermore, unlike computing servers, most network parts (including switches, hubs, and routers), cannot be energy-proportionate; features like hibernation during periods of low traffic and connection-rate adaptability are not built in by default [62]. Consequently, the design and execution of approaches and technologies that seek to minimize network

energy usage and make it proportionate to the incoming load continue to be outstanding issues.

QoS assurance presents another complex challenge within CDC networks [63]. Service Level Agreements (SLAs) in modern clouds focus mostly on computing and storage. There is currently no way to encapsulate network performance constraints like latency and bandwidth assurances without resorting to "best effort" because no abstraction or method guarantees performance isolation. Providing network connections across widely dispersed resources (in other words, installing a "virtual cluster" encompassing resources in an amalgamated cloud setting), exacerbates this difficulty [64]. However, there are numerous open challenges to deliver reliability assurances for these networks—due to packages needing to navigate the (public) Internet, including resources in various locations [65].

*4.1.2.2. Technologies/impact areas.* The key technologies and affected domains for networked computing include:

*4.1.2.2.1. Internet of things (IoT)* Devices (a.k.a., things) that can detect, control, and communicate are now routinely integrated with continuous control and monitoring functions via the Internet [1]. These devices have become ubiquitous in modern society, found in homes, on public transport, along highways, and in vehicles. Because of this, IoT applications may function in many contexts and provide a sophisticated evaluation and administration of complicated relationships [66]. As a result, IoT devices and services may solve problems in many application domains, such as digital health, facility administration systems, production, and transportation. IoT-based systems have to deal with limited processing power, memory, and storage space because (i) platforms are constantly changing, so devices that join a network have to be able to adapt to these changes; (ii) devices differ in how well they work with computers and what features they offer; and (iii) to ensure the safety of the IoT data that has been acquired, a federated system is needed [5].

These days, popular IoT use cases include medical care, smart cities, climate prediction, water supply management, and highway surveillance, all of which leverage the capabilities of cloud, serverless, fog, and edge computing for processing user data to meet QoS requirements [67].

- **Healthcare:** Among the many significant IoT applications is medical care, which is designed to treat conditions including heart attacks, diabetes, cancer, COVID-19, and influenza [68]. For instance, a patient's heart condition may be instantly diagnosed using a variety of medical devices in an interconnected IoT and computing environment [69]. Additionally, modern technology like Virtual Reality (VR) or AI can enhance the present healthcare system in the fight against inevitable pandemics [70].
- **Agriculture:** In order to forecast variables like yield, rainfall, and crop quality, the agricultural industry is making use of modern technology to analyze a wide range of data pertaining to agriculture [71]. One use case is the development of cloud-based agricultural systems that can autonomously forecast the state of agriculture using data collected from a variety of IoT or edge sensors. Additionally, to facilitate automated farming, an iOS or Android application is created to handle the massive amounts of data and supply the information they need to the agriculturalists through their edge devices [72].
- **Smart Home:** Owners may optimize energy consumption and offer the necessary protection with the deployment of cameras through the implementation of smart homes, which allow them to operate their home devices from their cell phones [36]. For instance, a resource management approach that incorporates cloud and fog computing may be used for controlling edge devices utilizing a smartphone application, which in turn regulates the room's humidity, lighting, surveillance systems, fans, and voltage, such as via sensors connected to different household devices [73].
- **Traffic Management**: IoT is crucial in the efficient management of traffic through the use of a number of sensors and controllers [74]. To identify potholes, for instance, an IoT-based intelligent transportation system is created. In addition, its efficiency was assessed using a range of machine learning approaches and performance metrics [75]. Additionally, data may be processed swiftly using fog and edge computing methodologies to notify about potholes early, thereby reducing the likelihood of mishaps.
- **Weather Forecasting:** Through the use of cloud computing and the IoT, scientists and weather forecasters may better gather data to inform their work [76]. Scientists have long relied on visual observations, data storage, and the public presentation of meteorological factors like air quality and moisture to better understand and explain these phenomena [77]. The findings may be made using an IoT system that relies on sensors and can transmit the results to the cloud.

Cloud services have long been relied upon by IoT applications to handle processing and permanent storage. Still, as the number of 'things' proliferates, such services are increasingly unable to keep up with the real-time demands of IoT gadgets [78]. This is due to the high quantity of data and the short reaction times required by systems that operate in the real world over wide geographical areas. By moving resource orchestration from servers to edge networks, fog/edge computing expands the capabilities of cloud systems: Set up as a series of nested "cloudlets" that may perform data intake, processing, and administration [79]. Compared to cloud services, geographically localized solutions use less power and allow for more mobile resources by decreasing reaction times and increasing intake bandwidth through horizontal scalability. These features make fog/edge computing a potential future architecture for IoT applications since this architectural model allows for scalability on a logical and geographical scale with near-instantaneous response latency [32].

By aggregating information from implanted and mobile gadgets and establishing mobile area networks, smart e-health apps can track information about patients in a continuous fashion [80]. By performing tasks like healthcare equipment noise filtering, data reduction and fusion, and analytics that identify harmful patterns in patients' well-being, smart gateways gather and interpret data from devices locally [81]. At the same time, longer-term patterns may be evaluated at cloud levels.

In addition, IoT systems supported by fog computing may adjust their actions based on the information they receive from sensors. For example, if a heart attack is recognized by initial processing at the fog layer, the intelligent gateway gathering signals from the defibrillator may adaptively boost the sample size before the attack. Similarly, the Industrial Internet of Things (IIoT) benefits from integrating edge, fog and cloud layers to provide specific and nearly real-time actions [82]. Smart grids and energy management are central to the Internet of Energy (IoE) paradigm. Coarse-grained information on network health may be gathered from dispersed networks of energy producers that track power usage, generation and/or battery life. While 'Smart-Meters' may communicate energy needs to service providers on a more detailed scale, monitoring capacity, generation, and use [80]. Therefore, IoT is a foundational technology for future systems, like electric automobiles and decentralized power grids [33]. In addition, the increased safety, reliability, and durability of electricity distribution that this type of grid may provide can better satisfy the evolving needs of consumers. In-depth surveys are a good resource for IoT researchers who want to explore more.

*4.1.2.2.2. Software-defined network (SDN)* SDN transcends traditional network paradigms by separating control logic from the underlying hardware and centralizing network management [83]. This innovative approach facilitates programmable network architectures and streamlines management by distinctly segregating network policies, hardware implementation, and traffic forwarding [84]. Integral to

cloud computing, SDN enhances communication and automates configurations, revolutionizing network adaptability and resource utilization in diverse environments [85].

NFV is another approach that utilizes software programs to perform traditionally hardware-based networking tasks, such as DNS, load balancing, and intrusion detection. NFV not only lowers costs but also enhances the flexibility of network functions and service responsiveness. Furthermore, VM consolidation in a virtualized network can help reduce energy costs by minimizing the number of VMs in operation [86]. SDN-based cloud computing optimizes network virtualization while decreasing electricity consumption. Crucially, SDN increases the abstraction of physical assets and automates and optimizes the setup process [85].

Many questions still need to be answered by scholars and investigators. First, ensuring data safety during transit across multiple cloud data centers is absolutely necessary for SDN-based cloud computing [87]. Second, even if SDN-enabled cloud infrastructures may be replicated, the balance between cost and energy use remains. Deploying SDN-based cloud computing systems is necessary to offer an economical network virtualization service with lower energy costs and greater dependability [83]. Furthermore, this may also boost data distribution and outcome collection by utilizing methods inspired by AI-based models, allowing us to expand existing information connectivity in such SDN contexts to accommodate blockchain-based systems.

*4.1.2.3. Trends/observations.* The main trends and observations regarding networked computing are as follows:

*4.1.2.3.1. Intelligent edge* The IoT connects billions of new devices, generating massive amounts of information that, inevitably, proves challenging to process. Over 41.6 billion IoT gadgets are estimated to be in operation by the end of 2025 [88]. Increasing numbers of products, including connected autos, smart meters, and in-store sensors, are being created and installed by companies to improve customer experience while generating enormous quantities of data [4].

Meanwhile, this emerging data must be gathered, managed, and processed immediately. *What exactly will this mean?* Edge and fog computing might be a method for moving ahead. In the coming years, edge computing is forecast to receive far greater focus than fog computing. In contrast to traditional cloud computing, which analyzes data at a remote data center, edge computing performs so locally. In fog computing, it is possible to execute a portion of the work in the cloud, while edge devices perform other aspects [89]. Since computing at the edge uses far less network bandwidth than conventional computing, the data exchanged among connected devices could take a long time. Computing it nearby, either on the gadget itself or within a local network, will be more cost- and energy-effective. On the contrary, edge computing may provide cloud computing with much-needed support in coping with the vast volumes of data created by the IoT and other connected devices [90]. Emerging IoT devices create and transport data across the fog and edge, and their processing power is leveraged to carry out processes that could otherwise be performed in the cloud. Hence, managing these systems with fog and edge, IoT devices and support from the cloud requires distributing the intelligence along the computing tiers, which leads to edge intelligent [91].

The terms “fog” and “edge” allude to these novel network nodes for IoT devices. Thus, they aid businesses in reducing their reliance on the cloud by transmitting information to analytics platforms. Businesses can lessen their dependency on cloud platforms for data processing and thereby reduce latency across networks by implementing edge and fog solutions [92]. This will allow rapid evidence-based recommendations to assist them in their decision-making process. Nevertheless, once real-time processing is complete, edge devices must transfer data to the cloud for statistics to be performed on it [93].

A company’s communication network is largely concerned with enabling various remote apps and providing endless storage space, thanks to cloud computing, connectivity, and computing capacity. That will ultimately alter data processing at the edge in real-time, which is essential for optimal data utilization [90]. Future-proof network infrastructures will need to accommodate an unprecedented influx of smart devices. For real-time intelligence, it is crucial to have the decision-making process located close to where the data is produced. Self-driving automobiles and self-sustainable, smart factory equipment, for instance, require to be making split-second decisions [94]. Further, airline sensors collect data on engine efficiency in real time, allowing for predictive maintenance before a plane ever takes off. Potential cost reductions might be considerable. The more business connections an organization has, the more processing power and intelligence it can provide.

*4.1.2.3.2. 6G and beyond* The advancement to Industry 5.0 and the foundation of a technology-driven economy hinge on the development of Beyond 5G (B5G) and 6G networks. As communication and technological advancements increase, international industrial sectors will increasingly depend on 5G and B5G networks to provide revolutionary services and applications that will inevitably require ultra-low latency, unprecedented reliability, and continuous mobility [95]. Meanwhile, underpinned by Moore’s law, mobile devices have been rapidly adopting systems-technology co-optimization (STCO) and related system-building approaches, which departs from the conventional system-on-a-chip (SoC) approach [96].

Through cloud-based principles, including utilizing functioning between and among data centers, connecting in a micro-service setting, and concurrently offering reliable services and applications, it is expected that B5G/6G networks will be able to serve a wide variety of applications [97]. Both B5G and 6G networks aim to enable the smooth and complete integration of many industries, including the IoT, aerial networks (also known as drones), satellite accessibility, and submerged connectivity [98]. To keep up with this astonishing expectation, the next generation of networks (B5G/6G) will largely rely on cutting-edge AI/ML technology for intelligent network operations and administration. B5G and 6G infrastructures are anticipated to provide computationally intensive applications and services paired with infrastructure shifts [99].

Edge computing has received a lot of interest and is being evaluated as an integrated service in 6G networks to enable the two fundamental changes in network infrastructure and network services. While many studies have focused on features like cache services and compute offloading methods, little is known about mobile edge computing implementation. The necessity of moving forward with a software-centric strategy from the network core to the wireless layer was emphasized in the first efforts that contributed significantly to the conception of 5G [100]. As with 5G networks, 6G networks will depend heavily on SDN, which, together with NFV, represents a departure from the conventional hardware-centric strategy [9]. The mobile edge computing paradigm also encourages moving the base station (BS) and the core network functions to different places. BS functions are moved upstream to the cloud, and core network functions are moved downstream to the devices. The resulting boundary between the BS and the end device might be seen as an “edge” or “fog” domain [73].

While cloud computing has made it possible for users to access richer and more complicated apps by tapping into the resources of a remote cloud server, an alternative technique is needed to meet the extremely delicate latency criteria stated for use cases in 5G and maybe 6G [101]. This heterogeneous network design directly results from the complicated traffic distributions in today’s wireless networks. A wireless access point (AP), a macro BS, and a small cell BS are just a few examples of network access nodes that may provide stable and smooth connections for mobile users [102]. These network access nodes provide edge computing at network edges with less delay. The design of diverse mobile edge computing networks has gained more and more interest due to the varied properties of network access nodes, such as coverage capability and power transmission [101]. Nevertheless, it is

imperative to carefully plan for the offloading of computing tasks to many access nodes in a network [103,104].

In a heterogeneous network design, intelligently distributing tasks and resources among different nodes can substantially boost system performance [55]. For instance, by collaborating, the edge and cloud can elevate IoT tasks' QoS. While cloud servers manage compute-intensive tasks well, edge servers excel at processing tasks demanding minimal data or low latency [41]. Strategically assigning tasks among edge servers can redistribute work from overburdened nodes to less active ones, thus accelerating execution times.

### 4.2. General purpose vs. Specialized computing

Leveraging fit-for-purpose software and given enough time, general-purpose computing (which includes desktop PCs, laptops, mobile devices like tablets and smartphones, and even certain televisions) can handle just about any computation [105]. A CPU, memory, input/output systems, and a bus are the main parts of any general-purpose computing system. In contrast, integrated computers, are used in intelligent systems, and are often referred to as "special-purpose" computing systems.

#### 4.2.1. General-purpose computing

In this section, we discuss the main focus, paradigms, technologies, and impact areas, as well as various trends and observations about general-purpose computing.

*4.2.1.1. Focus/paradigms.* The following are the main focus areas and paradigms associated with general-purpose computing:

(1) *Von Neumann Architecture*: A computing device with a Von Neumann architecture has its main components – the CPU, memory, and I/O – connected via a single bus [106]. The efficiency of computers was greatly enhanced by the advent of this architecture, which provided effective means of storing and executing instructions. The fundamental idea behind this design is that data and instructions are handled in the same way. In other words, the data being handled and the program instructions themselves share the same storage and processing resources: a memory address can contain either an instruction to be executed or data; the software execution pathways decide how to interpret it [107]. This design substantially simplifies the framework and features of a computer, making it more accessible to both software engineers and non-technical users.

(2) *Integrated Computing*: Compatibility throughout cloud applications and services is commonly achieved by implementing software adaptors and libraries and deploying application containers for computing to facilitate mobility [108]. Nevertheless, there is still a variety of challenges that have existed since the beginning of cloud computing but, due to their complexities, have not been adequately resolved yet [60]. One of these challenges is encouraging cloud connectivity without mandating a baseline set of capabilities for all services; ideally, customers can combine complicated features from several providers. Another area of investigation is how to develop cloud interoperability middleware that can facilitate the offering of complex services by composing more straightforward services from multiple 3rd-party providers [109]. Such a high degree of abstraction would empower users to make service decisions based on their requirements, such as price, turnaround time, privacy, etc. This brings up an additional key area that needs further study: the manner in which to allow user-level middleware (intercloud and hybrid clouds) to discover potential services for an ensemble without assistance from cloud service providers [110]. A strategy that relies on cloud providers working together is unlikely to be successful because their financial goals lie in keeping all the features they offer to their consumers (i.e., they have no incentive to help due to the fact that just a portion of the offerings in an ensemble are themselves). Consequently, the middleware that allows the melody of services must address challenges at both of its connections: Firstly, the middleware should seamlessly and abstractly provide the service to cloud users. Secondly, for the consumers, a service might be implemented in its entirety by sub-services offered by one vendor (maybe leveraging a 3rd-part SaaS organization able to offer the functionality), or it might be acquired through composing multiple services from various providers [111]. The provider user interface makes it possible to access complex functions without requiring special cooperation from providers [109]. The widespread use of cloud compatibility can offer commercial and financial advantages to cloud manufacturers, but frequently integrated clouds (which were achieved via Cloud Federation) cannot be realized until such time [112]. This calls for the development of intercloud markets, distinctive approaches to invoicing and accounting, as well as novel cloud-suitable pricing systems.

*4.2.1.2. Technologies/impact areas.* The key technologies and affected domains for general-purpose computing include:

*4.2.1.2.1. Programming models* Clusters are a type of parallel or distributed computational system that consists of a group of interconnected standalone computers that work collectively as a single integrated computing resource. Clusters and grids are platforms that communicate with each other to serve as a single resource [113]. A multi-core parallel architecture describes this form of capability, which is based on specific functions. Conversely, cloud computing emerged on top of clusters to abstract leveraging their computing resources and coordinate enormous data sets.

A programming model is tightly coupled to where data is transmitted to manage an application's functions. Important metrics to remember while building a programming model are efficiency, adaptability, goal architecture, and code maintainability [114]. Data analytics software often handles massive data sets that require many phases of processing. Certain steps have to be carried out in order, while others are executed simultaneously across several nodes in a cluster, grid, or cloud. The capacity of algorithms to perform statistical analysis on huge amounts of data will be crucial to unlocking achievements in industrial advances and next-generation scientific discoveries [115].

With the exponential growth of data comes the difficulty of organizing massive data sets, which in turn increases their complexity due to the ways they connect with each other. Its many processes include moving, archiving, replicating, processing, and erasing data. Data life-cycle complexities can be reduced via solutions that automate and improve data management activities. It has been shown that two limitations affect the data life cycle [116]. The framework used is the first limitation, initially regarding how it operates on data derived from consumers and apps. The second limitation derives from the observation that data is spread over several systems and infrastructures. That is why big data applications need to be capable of communicating amongst various systems that deal with the data and the effects that information and occurrences might have. The focus of this work is the second limitation, the big data infrastructure itself, and it includes a comprehensive analysis of the programming models and settings necessary to overcome this limitation.

A programming model is underpinned by how quickly and smoothly its data is manipulated. A few elements to consider when creating a programming model include operation, adaptability, target designs, and the simplicity of maintenance code modification procedures [117]. For the sake of service, it is sometimes necessary to sacrifice at least one of these aspects. The exchange of computation for data storage or transmission is a usual instance of algorithmic manipulation. These difficulties can be mitigated by employing parallel methods and technology. A software engineer might undoubtedly leverage many variants of the same technique to enable distinct performance adjustments on different hardware architectures. Modern computing clusters comprise nodes with more than one CPU, and their hardware designs range from tiny to super powerful.

*4.2.1.2.2. Virtualization* With virtualization, the original physical object is replaced with a virtual one. The OSs of server infrastructure, hard drives, and PCs are some of the most typical targets for virtualization in a data center. Thus, virtualization decouples higher-level software and OSs from the underlying computing system [118].

VMs are a key component of hardware virtualization, standing in for a "real" computer running an OS. Emulating a computer system is what VMs do. A hypervisor makes a copy of the underlying hardware so that several OSs can share the same resources [119]. Despite being around for half a century, VMs are experiencing a surge in popularity because of the rise of the mobile workforce and desktop PCs. Server virtualization, which employs a hypervisor to effectively "duplicate" the underlying hardware, is a primary use case for virtualization technology in the corporate world [120]. In a non-virtualized setting, the guest OS generally works in tandem with the hardware [121].

OSs may be virtualized and continue functioning as if running on hardware, giving businesses access to similar performance levels while reducing hardware costs [122]. The majority of guest OSs do not need full access to hardware; therefore, even if virtualization efficiency is lower than hardware efficacy, it remains preferred. This means firms are less reliant on a single piece of hardware and have more leeway to make necessary changes.

Following the success of server virtualization, other sections of the data center have also begun to implement the same approach. Virtualization technology for OSs has been around for generations [123]. In this implementation, the software enables the hardware to run several OSs in parallel. Companies that want to adopt a cloud-like IT infrastructure should prioritize virtualization. Using server resources more effectively is one of the primary benefits of virtualizing a data center [124]. Thanks to virtualization, IT departments may use a single VM to host a wide variety of applications, workloads, and OSs, with the flexibility to add or subtract resources as required easily. The use of virtualization allows firms to expand readily. Organizations may better monitor resource utilization and react to shifting needs using such systems.

*4.2.1.2.3. Multicore processors* For improved performance and more efficient use of energy, integrated circuits with several processing cores, or "cores", are becoming increasingly common. Furthermore, these processors enable more effective parallel processing and multithreading, allowing for simultaneous processing of numerous jobs [125]. A computer with a dual-core arrangement is functionally equivalent to one with two or more individual CPUs. Sharing a socket between two CPUs accelerates communication between them. The use of processors with multiple cores is one technique to enhance processor performance while surpassing the practical restrictions of semiconductor manufacturing and design. Using several processors helps prevent any potentially dangerous overheating [126]. Multicore processors are compatible with any up-to-date computer hardware architectures. These days, multicore processors are standard in desktop and portable computers. Nevertheless, the actual power and utility of these CPUs depend on software built to leverage parallelism [127]. Application tasks are broken up into many processing threads in a parallel strategy, distributing and managing them over multiple CPUs.

*4.2.1.3. Trends/observations.* The main trends and observations regarding general-purpose computing are as follows:

*4.2.1.3.1. Software systems* Web-based computing and Software Engineering (SE) are closely related disciplines. For instance, service-oriented SE provides various advantages to the software creation procedure and app development by merging the greatest elements of services and the cloud. In contrast to cloud computing, which is concerned with effectively transmitting services to consumers using adaptable virtualization of resources and load balancing, service-oriented SE is concerned with architectural design (service searching and composition) [128].

Customers and developers are both essential to the evolution of hardware innovations, which is why software engineering is a crucial discipline [129]. With the help of distributed computing and virtualization, customers may set up automatically managed VMs and cloud services for their initiatives and applications. Thanks to cloud services, teams working on software may now more easily collaborate on the development, testing, and distribution of their products. Here are some scenarios in which cloud computing might improve software engineering: The production timeline can be compressed [130]. As a result of the availability of ample computing resources made possible by cloud computing and virtualization, software engineers no longer need to rely just on a single physical computer. The time it takes to install the required applications may be decreased by retrieving cloud services, indicating that development activities can be performed with increased parallelism thanks to cloud computing. Third, VMs and cloud instances may substantially improve the setup and delivery procedures.

Using sufficient virtualization resources from a private or public cloud, developers can speed up the building and testing process, which is otherwise, extremely time-consuming [123]. To circumvent this, a simplified system for managing code versions is required. In software development, code branches are used for refining and adding features. With cloud computing, there is no need to invest in or lease expensive hardware only to store some code. A distributed software engineering team may access apps more easily in a cloud setting, and service quality can be enhanced through dynamic resource allocation. As a result, the software construction process is streamlined thanks to cloud computing, which eliminates the need for development servers to rely on specific physical computers [129]. Nevertheless, there are obstacles when merging software engineering and cloud computing. The majority of the difficulties are with moving the data. Because different cloud providers use various APIs to offer cloud services, migrating software and data from one cloud to another while avoiding vendor lock-in is challenging. Avoiding over-reliance on any one set of APIs is one way to fix this problem while building and releasing applications in the cloud. The problem of dependability and accessibility is another obstacle. If everything is moved to the cloud, it will be difficult to retrieve the data if the cloud is compromised by hackers or affected by an unexpected calamity. The engineering teams are responsible for creating a local backup of their work [131].

Cloud computing allows software engineering academics to study multinational software development. Several investigations have examined the feasibility of using cloud computing to lower operational, delivery, and software development expenses. Researchers have investigated the feasibility of replacing services with a cloud-based platform for student-to-student knowledge exchange and collaboration [132]. Software systems have been supplanted by systems running on the cloud to reduce expenditures and maximize the utilization of resources. The conventional data management techniques have become increasingly cumbersome in the past few years due to the rapid increase in available data. A new frontier for study in software engineering has opened up thanks to the IoT, Blockchain (the distributed ledger), and ML/AI, with data management being the primary challenge [133]. These studies also provide a springboard for further study and innovative approaches to cloud data management, leading to the development of advanced technologies like Cisco's pioneering fog computing [134]. Enterprise software developers are creating an abstraction layer, or "Blockchain-as-a-Service", and selling it to other businesses as a subscription service [4]. These numerous new fields rely significantly on software engineering, yet they could not exist without it.

*4.2.1.3.2. Simulations* The capacity to carry out research, analyze strengths and shortcomings, and demonstrate viability is hampered in new or emerging computing domains due to the lack of mature technology and sufficient infrastructure. In many cases, the time and resources needed to acquire the necessary physical resources make it impractical to conduct the necessary research [79]. An alternative approach that can approximate a physical environment is a simulator.

Additionally, simulation offers the ability to test suggested hypotheses in lightweight and low-cost settings. Real-world testing of novel methods is difficult and expensive because of the time and effort required to gather the necessary hardware resources (particularly for large-scale tests) and create the necessary software applications and systems [135]. Investigators demonstrate the viability of their ideas by modeling and simulation, and then conduct tests to confirm their concepts in a monitored environment utilizing simulation tools. Simulation software provides a convenient setting for testing solutions to real-world issues by allowing users to experiment and see what happens [136]. If a commercially available simulator is inadequate for user needs, then researchers should consider building their own, complete with graphical user interfaces. This is especially true if users need to simulate components of emerging computer architectures [137]. Researchers could benefit greatly from using a simulator to formulate questions and analyze different theoretical frameworks in simulated setups, therefore stimulating more research and fostering the development of communities within the relevant field.

#### 4.2.2. Specialized computing

In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within specialized computing.

*4.2.2.1. Focus/paradigms.* The following are the main focus or paradigms for Specialized Computing:

*4.2.2.1.1. Reconfigurable computing* The modern paradigm of reconfigurable computing enables hardware components to swiftly alter their configuration and functioning in response to changing processing needs. Reconfigurable computer devices, such as Field-Programmable Gate Arrays (FPGAs), can be reprogrammed to perform a variety of different functions [138]. The main function of reconfigurable computing is to fill the void among general-purpose CPUs and Application-Specific Integrated Circuits (ASICs) [19]. It allows hardware to be optimized for efficiency, power efficiency, and flexibility by matching application requirements. Static and dynamic switching are the two primary modes of operation for reconfigurable technology. In static reconfiguration, the component settings are adjusted prior to the computer starting to compute. However, dynamic reconfiguration permits hardware changes to be made while the system is running, allowing for dynamic modifications to hardware behavior.

*4.2.2.1.2. Domain-specific architectures* As computing and the digital transformation spread to various use cases, such as cloud (AI/HPC), networking, edge, the IoT, and self-driving cars, highly domain-specific computational tasks are making it more likely that Domain-Specific Architectures (DSAs) can enable big performance gains [139]. Using ChatGPT and other comparable software that are powered by large language models – which are fundamental to achieving generative AI – provides greater specialization inside AI workloads at extremely high volume, which motivates further hardware specialization [81]. DSAs, or application-domain-specific hardware and software, have substantial market potential. As a result of their superior performance on tasks that profit from a significant amount of parallel computing, such as AI workloads (learning and predicting), GPUs and Tensor Processing Units (TPUs) are currently controlling a sizable portion of the data center market [140]. Meanwhile, accelerations of 15–50 times the original speed, depending on the workload, are not uncommon. In the automobile industry, minimal latency and high-performance inference are provided via tailor-made solutions from industry leaders.

*4.2.2.1.3. Exascale computing* To handle the massive computations required by convergent modeling, simulation, AI, and data analysis, an entirely novel type of supercomputer called exascale computing has emerged [2]. This is motivated by advanced computational needs in science and engineering.

Exascale computing (also supercomputing) becomes essential to expedite the generation of knowledge. Researchers and technologists may employ data analysis driven by exascale supercomputing to expand the frontiers of our existing understanding and promote breakthrough ideas. Supercomputing capabilities are in high demand as the world moves towards exascale computing to ensure continued scientific and technological advancements, while our civilization's technological and scientific frameworks are progressing quickly thanks to exascale computing [141]. The immense potential of these tools necessitates their careful operation, especially as cultures worldwide undergo rapid changes in their moral frameworks and their perceptions of what it means to live sustainably. As such, novel responses to formerly intractable issues are being uncovered thanks to exascale computing.

Exascale supercomputers are prohibitively expensive to construct; thus academics and scientists rely on funding to lease them instead of buying their own [142]. Exascale computing systems produce enormous quantities of heat because of the volume of data they process. They require extremely cold environments to be stored in or unique cooling mechanisms built into the systems and racks themselves for optimal performance. Differentiating them from other types of supercomputers and quantum computers, they are computer systems with the largest capacity and most powerful hardware [143].

To further our understanding of the universe, exascale computers can model elementary physical processes like the granular interactions of atoms. Quite a few sectors rely on this capacity to analyze, forecast, and construct the world of tomorrow: for instance, better predict the weather, investigate in detail the interaction between rain, wind, clouds, and various other atmospheric occurrences, analyze their effects on one another at a molecular level and so on. Mathematical formulas can be used to determine the millisecond-by-millisecond effects of all forces acting in a certain environment at a specific time [144]. These seemingly trivial interactions rapidly generate billions of possible permutations, which need trillions of mathematical equations to calculate and analyze. This kind of speed is only achievable on an exascale machine. By studying the results of these computations, researchers can gain a deeper insight into the nature of our universe [143]. Exascale supercomputers, despite their challenges, can literally increase our understanding, enabling us to address the problems of the future.

*4.2.2.1.4. Analog computing* A novel approach may minimize errors in ultra-fast analog optical neural networks. Larger and more complicated machine-learning models need stronger and more effective computing gear. However, standard digital computers are lagging. Compared to a digital neural network, an analog optical network's performance in areas like image classification and voice recognition is comparable. However, its speed and energy efficiency far exceed those of its digital counterparts [145]. Nevertheless, hardware faults in these analog devices might impact the accuracy of calculations. One possible source of this inaccuracy is microscopic flaws in the hardware itself [146]. Errors tend to multiply rapidly in a complex optical neural network. Even when using error-correction approaches, due to the basic features of the components that make up an optical neural network, a certain degree of error is inescapable [147]. Conversely, the optical switches that make up the network's architecture can reduce mistakes they typically accrue by adding a modest hardware component.

*4.2.2.1.5. Neuromorphic computing* When applied to AI, neuromorphic computing makes it possible for AI to learn and make decisions independently, significantly improving over the first generation of developing AI. To acquire abilities in areas like recognizing voice and sophisticated tactical games, including chess and Go, neuromorphic algorithms are now involved in deep learning [145].

Next-generation AI will imitate the human brain's capacity to comprehend and react to circumstances instead of merely operating from formulaic algorithms [148]. When it comes to understanding what they have read, neuromorphic computing systems will seek out patterns and use their 'common sense' and the surrounding context. When Google's Deep Dream AI was programmed to hunt for dog faces, it notably showed the limitations of algorithm-only computer systems [147]: Any

images that it interpreted as having dog faces were transformed into dog faces.

Third-generation AI computing attempts to simulate the elaborate structure of a living brain's neural network [149]. This calls for AI with computing and analytic capabilities on par with the extremely efficient biological brain. To demonstrate their exceptional energy economy, human brains can surpass supercomputers using less than 20 watts of electricity. Spiking Neural Networks (SNN) are the AI equivalent of our synaptic neural network [150]. They leverage many layers of artificial neurons, and each spiking neuron may fire and interact with its neighbors in response to external inputs.

Most AI neural network architectures follow the Von Neumann design [106], which divides the memory and computation into discrete nodes. Computers exchange information by reading it from memory, sending it to the CPU for processing, and then returning it to storage. This constant back-and-forth wastes a lot of time and effort. It causes a slowdown that becomes more noticeable while processing huge data sets. As a response, multiple neuromorphic devices can be utilized to supplement and improve the performance of traditional technologies, such as CPUs, GPUs, and FPGAs [146]. Low-power neurological systems may perform powerful activities, including learning, browsing, and monitoring. A practical instance would involve immediate voice recognition on mobile phones without the CPU needing to interact with the cloud.

*4.2.2.2. Technologies/impact areas.* The key technologies and affected domains for Specialized Computing include:

(1) *Graphics Processing Unit (GPU)*: GPUs have rapidly risen in prominence as a crucial component of both home and enterprise computers [18]. A GPU is a special type of computer chip deployed in a variety of application domains, most notably the rendering of moving images. While GPUs are best recognized for their usage in gaming, they are also finding increasing application in the fields of creative creation and AI [151]. The initial purpose of GPUs was to speed up the display of 3D visuals. They improved their functionality as they got more adaptable and programmable over time. This paved the way for more complex lighting and shadow characteristics and photorealistic environments to be implemented by graphics developers. Additional engineers started using GPUs to drastically speed up various tasks in deep learning, HPC, and other fields [138].

(2) *Compute Unified Device Architecture (CUDA)*: The demand for more powerful computers grows daily. As a result of constraints like size, climate, etc., vendors throughout the world are finding it difficult to make future improvements to CPUs [18]. Service providers that provide solutions in this kind of environment have begun to seek out performance improvements elsewhere. The use of GPUs for parallel processing is one option that enables significant speed gains [152]. The total number of cores in a GPU is significantly greater than that of a CPU. Although CPUs are designed for sequential processing, offloading them to GPUs enables parallel processing. For general-purpose computing on NVIDIA's GPUs, users can rely on CUDA, which allows for the execution of processes in parallel on the GPU without any specific order requirement [138]. Offloading compute-intensive activities to Nvidia's GPU using CUDA is straightforward thanks to the library's support for the popular C, C++, and Fortran programming languages [152]. CUDA is employed in scenarios needing extensive computational power or suitable for parallel processing to achieve high performance. Fields such as AI, healthcare analysis, science, digital transformation, cryptocurrency mining, and scientific modeling, among others, depend on CUDA technology.

*4.2.2.3. Trends/observations.* The main trends and observations regarding Specialized Computing are as follows:

*Large-Scale ML*: As big data grows, ML algorithms with many variables are needed to ensure that these models can handle very large data sets and make accurate predictions, including hidden features with many dimensions, middle representations, and selection functions [153]. The need for ML systems to train complicated models with millions to trillions of variables has increased as a result [154]. Distributed clusters of tens to hundreds of devices are often used for ML systems because they can handle the high computing needs of ML algorithms at these sizes. Yet, developing algorithms and software systems for these distributed clusters requires intensive analysis and design [155]. The latest advances in industrial-scale ML have focused on exploring new concepts and approaches for (a) highly specialized monolithic concepts for large-scale straight applications, such as different distributed topic models or regression models, and (b) for adaptable and readily programmable universally applicable distributed ML platforms such as GraphLab based on vertex programming and Petuum using a parameter-driven server [156]. It is widely acknowledged that knowledge of distributed system topologies and programming is essential; however, ML-rooted statistical and algorithmic discoveries can yield even more fruit for large-scale ML systems in the form of principles and techniques specific to distributed machine learning applications. These guidelines and techniques shed light on several crucial questions:

- How to share an ML application among nodes?
- How to connect machine-learning calculations with machine-to-machine dialog?
- How should one proceed with having such a conversation?
- What ought to be conveyed among machines? And, should they cover many big ML-related topics, from practical use cases to technical implementations to theoretical investigations [98]?

Understanding how these concepts and tactics may be made effective, generally applicable, and easy to develop is the primary goal of large-scale ML systems studies, as is ensuring that scientifically validated accuracy and scalability assurances underpin them.

### 4.3. Centralized vs. Decentralized computing

A central server controls and processes most of the data in a centralized network, whereas no single entity has influence over a decentralized network.

#### 4.3.1. Centralized computing

In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within centralized computing.

*4.3.1.1. Focus/paradigms.* The following are the main focus or paradigms for centralized computing:

(1) *Cloud Computing*: The adoption of cloud computing, which revolutionized how end-users and software engineers interact with applications and computing systems, led to the rise of technology as the fifth utility [1]. Cloud computing was successfully accepted by giving consumers on-demand access to the computing power they want, the freedom to modify their resource consumption as needed, and the transparency of paying just whatever is being utilized. Business groups, regulatory bodies, and universities have all been quick to endorse it since it first appeared. Like contemporary society relying on essential utilities, the cloud has grown into the economy's foundation by providing immediate utilization of subscription-driven computing resources [157]. As a result of using cloud technology, innovative companies can be launched quickly, existing ones can expand globally, advances

in science can be sped up, and novel computing methods can be developed for ubiquitous and pervasive apps [34]. SaaS, PaaS, and IaaS have served as the three primary service models that have pushed uptake in the cloud thus far [35].

- *Mobile Cloud Computing*: To provide value to mobility consumers, network operators, and cloud service providers, mobile cloud computing integrates mobile devices, cloud computing, and communication networks. With the help of mobile cloud computing, a wide variety of handheld gadgets can run complex mobile apps. Under this paradigm, handling and storing data is done by servers rather than individual mobile devices [32]. Several advantages result from the use of mobile cloud computing apps based on this architecture: (i) battery life has significantly increased; (ii) there has been an increase in both the speed and size of data being stored and processed; (iii) the system's emphasis on "store once, access anywhere" eliminates complex data synchronization; and (iv) stability and scalability have been dramatically enhanced. Nevertheless, inadequate network capacity is a significant challenge for mobile cloud computing [33]. Wireless mobile cloud services have capacity constraints in contrast to their cable counterparts. The spectrum of mobile devices offers a wide range of wavelengths. This has resulted in slower access speeds, as much as one-third in comparison to a wired network. Due to the increased likelihood of data loss on a wireless network, it is more challenging to recognize and deal with security risks on mobile devices than on desktop computers [158]. Customers frequently report issues with accessibility to services, including network outages, overcrowding on public transit, lack of coverage, etc. Customers may occasionally experience a low-energy signal, which slows down access and impacts data storage. Mobile cloud computing is employed on several OS-driven platforms, including Apple iOS, Android, and Windows Phone, resulting in network modifications that need cross-platform compatibility [159]. Mobile gadgets have a greater environmental impact due to their high energy consumption and low output [60]. As the use of mobile cloud computing grows, so does the problem of the increased drain on mobile devices' batteries. A device's battery life is crucial for using its software and executing other tasks. Although the modified code is tiny in size, offloading uses more energy than running it locally [160].
- *Green Cloud Computing*: In the last few decades, Information and Communication Technology (ICT) has significantly evolved, drawing on technological advancements from the past two centuries. This evolution has elevated computing to the status of a fundamental service, akin to traditional utilities such as water, electricity, gas, and telephony, thereby establishing it as the fifth essential utility in modern society [161]. Modern cloud computing systems are becoming progressively large-scale and dispersed as more and more businesses and organizations have shifted their computing workload to the cloud—while others opt out of maintaining code altogether and instead leverage cloud-powered SaaS services. A cloud computing infrastructure of this magnitude not only offers more affordable and dependable services but also, increases energy effectiveness and reduces the global community's carbon impact [162]. Every minor enhancement is much appreciated. In an effort to achieve zero carbon emissions, the community has recently been aggressively exploring a more sustainable version of cloud computing called green cloud computing to lessen reliance on fossil fuels and curb its carbon footprint [163].
  Green cloud computing is a system that considers its constraints and goals to minimize energy consumption. Researchers are focusing on scheduling workloads and resources in light of carbon emissions, in order to increase the effectiveness of the resources used [164]. Additionally, forecasting problems with hardware and creating management systems to use hardware with varying degrees of dependability can maximize device lifetime and reuse. Further, utilizing micro-data centers – rather than standard server data centers – is a promising approach to boost efficiency and save costs. These facilities can accommodate future growth, serve huge populations, and dissipate heat effectively [165].
  Furthermore, virtualization is another ecologically friendly technique that boosts the versatility of system resources. Through improved tracking and control, servers may pool their resources more effectively [166]. Innovations and practices that support sustainable development are constantly being developed as organizations rely more heavily on cloud services to enable "green cloud computing".

*4.3.1.2. Technologies/impact areas.* The key technologies and affected domains for centralized computing include:

(1) *Cloud Storage Technologies*: Files and information stored in the cloud may be accessed from anywhere with a web connection or via secure network access. Transferring files to the cloud puts the responsibility for data security squarely on the shoulders of the cloud provider, rather than consumers. The service provider hosts, manages, and maintains the servers where user data is stored, and they also guarantee that users always have access to their files [167]. When compared to storing data on local discs or storage networks, cloud-based storage is a more affordable and scalable option. There is a limit to the quantity of information that can be stored on a hard disc. When users exhaust internal storage space, they must copy their data to removable media. The difference between on-premises storage networks and cloud storage is that the latter sends data to servers located in a remote data center. VMs, which are abstracted on top of an actual server, make up the vast majority of users' servers [168]. Known as autoscaling, a cloud provider spins up more virtual servers as necessary to accommodate users' ever-increasing storage demands. Files, blocks, and objects are the three primary categories of cloud storage, which are accessible in private, public, and hybrid cloud configurations.

(2) *Microservices*: Microservices are a type of application architecture in which several autonomous services collaborate using simple APIs. A cloud-native software development method, microservice architecture separates an app's main functionality into its own modules [169]. By compartmentalizing the app's components, the development and operations teams may collaborate without interfering with each other. If several engineers can collaborate on the same project simultaneously, it takes less time to complete. This is in contrast with the monolith software architecture, which had been the standard for application development in the past [170].
  All of an app's features and services are tightly bound together and run as one seamless whole under a monolithic architecture [171]. The application's architecture becomes more involved whenever new features are introduced, or existing ones enhanced. Because of this, optimizing a single feature inside the application requires disassembling the whole thing, which is a time-consuming and tedious process. This additionally necessitates that scaling the application as a whole is required if scaling any one process inside it—rather than just scaling out just that overloaded element [172].
  Microservice architectures separate an app's essential features into individual processes. To adapt to shifting business requirements, software engineering teams may develop and maintain new elements independently of the rest of the application. The monolith has been the standard for application development in the past. An application's features and services are tightly bound together and run seamlessly under a monolithic architecture [173].

Microservices' malleability might hasten the deployment of novel modifications, necessitating the development of novel patterns. In software engineering, a "pattern" is supposed to refer to any mathematical approach that is known to function. An "anti-pattern" is an erroneous pattern that is often applied to achieve a solution but often ends up causing even more problems.

(3) *Container Technologies*: Given the advent of Docker, container technology has gained widespread use in the cloud computing sector, where it is used to efficiently execute user workloads [174]. Since containers are independent entities that may run without sharing data with other containers, this technology provides an inexpensive cloud environment for deploying applications. In a container, applications deployed on the same hardware server can share the same underlying resources while maintaining their own distinct processes [175].
Container technology leverages Linux kernel capabilities, such as `libcontainer` and control groups (`cgroups`). By utilizing `cgroups` and namespaces, Docker can operate containers independently within a host node, providing the container with its own dedicated set of runtime resources (including the host's networked devices, disc space, memory, and CPU). In addition, namespaces provide for more efficient application deployment and development by separating the program's perspective from the operating environment [176]. Furthermore, containerization becomes an example of creating, publishing, and running applications in an isolated way and is indicated as a Container as a Service (CaaS). There are three primary advantages of CaaS: (1) containers boot up in no time at all; (2) they consume fewer resources than VMs; and (3) many instances may be operated at once using container technology [177].
Recent investigations [178] into container technology reveal unanswered research questions. Firstly, containers are less secure than VMs since they share the kernel, but this is something that may be fixed in future versions with the help of Unikernel. Secondly, optimizing container performance is a time-consuming endeavor that requires buffer space. Swarm and Kubernetes are two examples of cutting-edge cloud computing tools that may be used for handling user-created QoS-based container clusters [179,180]. Thirdly, because containers share the same computing/hardware resources, co-located tenants can suffer from unpredictable performance interference when the CPU Shares algorithm is used, and even worse, they can leak information enabling side-channel attacks to be performed by a malicious tenant [181].

(4) *Serverless Computing*: The use of serverless computing in the creation of apps for the cloud is gaining traction [182]. The goal of serverless computing is to ensure that only the most effective serverless technologies are deployed, reducing costs while increasing benefits [183]. Meanwhile, companies in all industries are adopting AI since it is the next generation of innovation. Due to these AI-driven platforms, we have been able to make more accurate, timely decisions [184]. They have altered the methods used to conduct business, communicate with customers, and assess company information. Complex ML systems can significantly affect developers' output and efficiency [185]. However, switching to a serverless architecture may be able to solve many of the issues that engineers face. The serverless design ensures that the machine learning models are administered correctly and that all available resources are utilized efficiently. Developers will be able to devote a greater amount of time to training AI models rather than maintaining the server environment [186]. Creating ML algorithms is a common practice when confronting difficult problems. They perform tasks such as data analysis and preprocessing, model training, and AI model tuning [186]. Serverless computing running AI tasks will provide for reliable data storage and communication.

*4.3.1.3. Trends/observations.* The main trends and observations regarding centralized computing are as follows:

(1) *AI-driven Computing*: The fundamental benefit of autonomic computing is a reduced overall cost of ownership. Therefore, the cost of upkeep will be drastically reduced. The number of technicians required to keep everything running smoothly will go down as a result, too. Autonomous IT systems driven by AI will reduce the time and money needed for installation and upkeep while also improving IT system stability [4].
In accordance with higher-order benefits, businesses would be more capable of handling their operations with the help of IT systems that are able to adopt and execute directions based on their business plan and allow for adjustments in reaction to evolving circumstances. Using AI-based autonomic computing has several advantages, including reducing the expense and quantity of human labor needed to manage large server farms, which is made possible through server consolidation [187]. Using AI for self-driving computers will simplify system administration. As a result, computer systems will be greatly enhanced. Server load distribution is another potential use case since it allows for parallel data processing across several computers. Meanwhile from an energy perspective, analyzing the power grid in real-time allows for more cost-effective and long-term power policy decisions to be made [1].
There are benefits to using remote data centers instead of keeping data in-house. Despite the hefty upfront expenses, businesses may obtain AI technology relatively easily by paying a monthly fee on the cloud. When employing an AI-powered system, there may be no need for human involvement in data analysis [188]. Using AI in the cloud can potentially make businesses more effective, strategic, and insight-driven. AI can increase output by automating routine processes and data analysis without human intervention [74]. For instance, integrating AI technology with Google Cloud Stream statistics could enable real-time personalization, anomaly detection, and management scenario prediction [189]. As the number of cloud-based applications grows, it is essential to implement a system of rigorous data protection based on intelligence. Network security systems backed by AI-enabled traffic tracing and analysis; AI-enabled devices can sound an alarm as soon as an anomaly is detected. Such methods will ensure keeping sensitive data protected.

(2) Net Zero Emissions: Several data center operators have committed to being carbon neutral by the year 2030 as sustainability becomes an increasingly hot subject in the industry [190]. But are these promises only a reaction to the possibility of legislation, or is it actually making progress? If business planes are a major contributor to global warming, how do they plan to cut their carbon footprint so rapidly? The data centers' businesses in the United States use about as much power as the state of New Jersey [191]. If all of the power came from renewable resources, this level of demand would not be a problem. Liquid cooling and energy generation both require water, and a typical data center uses as much water as an urban area of 30,000 to 50,000 individuals [192]. Becoming a pioneer in sustainability might also bring up emerging markets. Companies are going to utilize green data centers to offset their carbon footprints as they grow and become more energy efficient and sustainable [193]. A car company, for instance, might employ emission-free data centers for all of its corporate services. Last but not least, adopting environmentally friendly practices may help businesses comply with environmental rules, avoid fines, and get access to attractive, low-interest, long-term capital investment possibilities [194].

*4.3.2. Decentralized computing*

In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within decentralized computing.

*4.3.2.1. Focus/paradigms.* The following are the main focus or paradigms for decentralized computing:

(1) *Parallel Computing*: Through the utilization of several processor cores, parallel computing can perform multiple tasks simultaneously. The ability to divide and conquer a work into smaller, more manageable chunks is what sets parallel computing apart from its serial counterpart [195]. Real-world events may be modeled and simulated effectively on parallel computing systems [196]. As processing and network speeds continue to increase at an exponential rate, adopting a parallel architecture is no longer just a nice-to-have. The IoT and big data will eventually require us to process terabytes of data simultaneously. Devices such as dual-core, quad-core, eight-core, and even 56-core CPUs utilize parallel computing. Therefore, although parallel computers are not brand new, this is the problem: These new technologies are spitting up ever-faster networks, and computer efficiency has surged 250,000 times in 20 years [197]. For instance, AI technologies will sift through more than 100 million patients' heart rhythms in the medical sector alone, looking for signals of A-fib or V-tach, saving many lives in the process [196]. When the systems must slowly move through each procedure, they will not be able to complete it on time. As great as the potential is, parallel computing may be nearing the edge of what it can achieve with conventional processors. Parallel calculations may see significant improvements in the coming decade, thanks to quantum computers. In a current, unauthorized announcement, Google claimed to have achieved *quantum supremacy* [76, 198]. If it is accurate, then Google has created a machine that can perform in 4 min whatever would require the most capable supercomputer on the planet 10,000 years to achieve [51]. Quantum computing is a major step forward for parallel computation. Imagine it like this: Processing in a serial fashion does one task at a time. An 8-core simultaneous computer can do eight tasks simultaneously. There are fewer particles in the universe than there are qubits' states in a 300-qubit quantum computer [198].

(2) *Fog Computing*: The proliferation of IoT devices and the effort needed for analyzing and storing enormous amounts of knowledge led to the development of fog computing as a complementary service to traditional cloud computing. Fog computing, which provides fundamental network functions, can back IoT apps that require a small response-time window [37]. Due to the dispersed, diverse, and constrained nature of the fog computing paradigm, it is challenging to spread IoT application operations effectively within fog nodes to meet QoS and Quality of Experience (QoE) limitations [39]. Vehicle-to-Everything (V2X), medical tracking, and manufacturing automation adopt fog computing as it delivers the ability to compute close to the consumer to match fast response demands for these applications. Due to the proliferation of IoT devices, these applications generate massive volumes of data. Cloud computing falls short of satisfying latency demands due to the transmission of data over long distances and network overload. Bridging data sources and CDCs, it sets up a network of gateways, routers, switches, and compute resources [199]. The use of fog computing enhances the capabilities of cloud computing due to its minimal latency and cost-effectiveness, as well as the decrease in bandwidth necessary for the transit of data. It is more secure to process confidential information locally at the fog nodes, and if/when needed, only submit trained models – not raw data – to intermediate nodes and eventually the cloud for aggregation, e.g., via federated learning [200]. These applications collect data from various IoT devices to deliver useful insights and deal with latency issues [201].

(3) *P2P Network*: This network is formed in its most basic form when two or more PCs are linked to one another and exchange resources without passing through a third computer that acts as a server [23]. A P2P network might be a spontaneous connection, which would consist of two or more computers linked together using a Universal Serial Bus for the purpose of file sharing. In a fixed infrastructure, P2P networking utilizes copper lines to connect six PCs located in a single workplace [24]. Alternately, a peer-to-peer network may be an ecosystem that is considerably larger in scale and is characterized by the use of specialized protocols and apps to establish direct links between consumers over the web.

(4) *Osmotic Computing*: This model has become pervasive in various settings, from urban planning and healthcare to linked vehicles and Industry 4.0 [46]. It lays the groundwork for a system in which vehicles, pedestrians, and urban infrastructure interact and share real-time information to improve traffic flow. As more people use IoT applications housed in different types of networks (cloud, edge, and IoT), it is now clear that the providers who make up the IoT's service ecosystem (data, service, network, and equipment) are all interconnected [48]. In this setting, buyers and sellers implicitly expect their data and services to be secure and trustworthy. There is no requirement for familiarity with the federated ecosystem (service, data, and network) for users of the IoT apps to connect with many applications using a web-based user interface [202]. Users send their information to application providers without realizing that those trusted suppliers may share that information with any third parties (such as a company that hosts analytics on the cloud or a company that provides the infrastructure for mobile devices). Security issues may arise for software due to the wide variety of computing devices available from different manufacturers and their presence in an untrusted realm with no overarching authority [203].

(5) *Dew Computing*: It stands out because of its near-complete independence from Internet access, its users' physical closeness to servers, its low latency, outstanding speed, excellent user interface, and adaptability in terms of control available to users [204]. Instead of serving as a replacement for cloud computing, dew computing serves as a useful supplement. In the not-too-distant future, people throughout the globe might be able to limit their time spent online, increasing their efficiency and effectiveness. Countries have adopted measures to handle the influx of Internet users caused by the COVID-19 blackout. To lighten the Internet's burden, video streaming services are reducing visual quality, while others just update their software outside of peak viewing times. The dew computer's proximity to the user in the design means it can facilitate all electronic interactions with fewer steps and more efficient data transfer [204].

(6) *Edge Computing*: Since its origins in content delivery networks, distributed computing has matured into the mainstream as an edge computing paradigm that places resources near the client's end. Big data is typically best stored in the cloud, whereas immediate information created by consumers and exclusively for the customer needs computing power and storage on the edge [40]. To accommodate growing mobile user needs, cloud providers have realized they must shift crucial processing to the device.

With its high performance and low cost, edge computing is a key driver for AI. This can be the most helpful method to see how AI relates to edge computing. Due to the data- and compute-intensive characteristics of AI, edge computing aids AI-powered applications in resolving their technical problems. AI/ML systems consume large amounts of data to discover trends and provide trustworthy recommendations [205]. AI use cases that need video analysis face latency challenges and rising expenses

due to the cloud-based transmission of high-definition video data.

The delay and reliance on central processing in cloud computing are problematic when ML inputs, outputs, and (re-)training data must be handled in real-time. It is possible to perform computation and decisions at the edge, eliminating the need for costly backbone connections and allowing immediate action on the data. Client information regarding location is stored at the edge instead of in the cloud for security reasons. When data is streamed to the cloud, all relevant data and datasets are uploaded. Edge networks for computing have introduced several difficulties associated with infrastructure management because of their dispersed and intricate nature [206]. Managing resources efficiently requires carrying out several tasks. Examples include VM consolidation, resource optimization, energy efficiency, workload prediction, and scheduling. Resource management has historically relied on static, established guidelines, mostly based on operations research methodologies, even in dynamic, rapidly changing settings and in immediate situations. To deal with these issues, especially when choices must be made, AI-based solutions are being used more and more frequently. AI/ML methods have become increasingly common in the past few years [207]. However, selecting where on edge to carry out a task can be challenging, as it requires considering tradeoffs like the volume of data on edge servers and the ability to move users [208]. The cache has to anticipate the consumer's next destination for it to build on the notion of mobility [209]. It is situated at a suitable edge to cut costs and energy consumption. Several different methods, including genetic algorithms, neural network models, and reinforcement learning, are utilized in this process.

- *Mobile Edge Computing*: Mobile Edge Computing – now Multi-access Edge Computing (MEC) – expands its possibilities by introducing cloud computing to the web's edge. Initially targeted solely on the edge nodes of mobile networks, MEC has since expanded its scope to include conventional networks and, ultimately, integrated networks. While typical cloud computing occurs on servers located far from the end-user and devices, MEC enables activities to be carried out at base stations, centralized controllers, and various other aggregating sites on the Internet [210]. MEC improves consumer QoE by redistributing cloud computing workloads to customers' individual, on-premises servers, thus relieving congestion on mobile networks and lowering latency [211]. Innovative applications, services, and user experiences are being unlocked at a dizzying rate thanks to advances in edge data generation, collection, and analysis and in the transmission of data between devices and the cloud [212]. Because of this, MEC is accessible to consumers and businesses in a wide range of contexts and industries. Integrating MEC into a camera network improves the speed with which data may be stored and processed. With sufficient processing power and bandwidth, data may be immediately analyzed locally instead of being sent to a remote data center [213]. Self-driving automobiles and autonomous mobile robots (AMRs) are two examples of emerging technologies that require powerful ML to arrive at judgments rapidly. If such decisions take place in a remote data center, only seconds might be the distinction between nearly escaping failures and causing a tragedy [205]. Because the vehicle must avoid hitting pedestrians, animals, and other vehicles, judgments must be made on the vehicle. Machine-to-machine (M2M) communication will be essential to the success of 6G as the forthcoming generation of a global wireless standard and the technological advances that will emerge from it [101].

*4.3.2.2. Technologies/impact areas.* The key technologies and affected domains for decentralized computing include:

*4.3.2.2.1. Distributed ledger technology* The computing paradigms of fog, edge, and cloud are currently experiencing explosive growth in both the business and academic worlds. Security, confidentiality, and data integrity in these systems have become increasingly important as their practical applications have expanded [214]. Data loss, theft, and corruption from malicious software like ransomware, trojans, and viruses raise serious considerations in this area. For the system's and most importantly, end-users' sake, it is crucial that data integrity be maintained, and that no data be delivered from an unauthenticated source. Medical care, innovative cities, transport, and monitoring are all examples of applications of critical importance where the margin for error is near zero [4].

- *Blockchain*: Because the majority of edge devices have limited computing and storage capacity, developing an appropriate system for data security, and preserving integrity is challenging. The IoT and other real-time systems have used blockchain technology for data security [134]. To store and monitor the worth of an asset over time, a blockchain is, in theory, a set of distributed ledgers. When new information is added to the system, it becomes a block with a Proof of Work (PoW). A PoW is a hash value that cannot be made without changing the PoW of the blocks that came before it in the ledger. Miners create and verify these PoWs while also mining blocks in the Fog network [215].

  After a miner has completed the PoW, it broadcasts the newly created block into the network, where the other nodes check its legitimacy before joining it in the chain. Also, the fraudulent change of data in a blockchain will not work unless at least half of the copies of the data in question are changed individually by carrying out the same actions. With such a strict time constraint, modifying any data in the blockchain will be extremely difficult. Network nodes must offer route selection, preservation, financial services, and mining for the blockchain to function. Considering these challenges, numerous groups have worked to develop solid frameworks for combining blockchain and fog computing [133]. The majority of these systems employ a dynamic allocation mining technique in which the least-used nodes mine and validate the chains. In contrast, the remaining nodes are employed for load balancing, computation, and data collection [108,216]. The blockchain on a large portion of the network is replicated at those nodes if a worker detects an issue in relation to blockchain manipulation or signature forging. Furthermore, blockchains offer public-key encryption with adaptive key exchange for further security. Blockchain is a deceptively simple central notion, but incorporating it into fog computing systems presents several challenges. Cost and upkeep are major factors surrounding storage capacity and scalability. Only complete nodes (nodes that can fully validate the transactions or blocks) store the whole chain, which still results in massive storage needs. Data anonymity and privacy issues are another blockchain shortcoming. Privacy is, therefore, not incorporated into the blockchain architecture; consequently, third-party tools are necessary for accomplishing these crucial requirements [217]. This might result in less efficient applications that demand more resources (both computationally and in terms of storage space) to run. There are still numerous unresolved issues and potential future developments for blockchains in IoT architectures [13].

  Insufficient resources are the main barrier to excellent data protection and dependability. Because of resource limits, more complex encryption or key generation cannot be incorporated with these chains of data [218]. Only restricted encryption algorithms may be implemented. By considering resource limitations, more effective algorithms may be created. In high fault-rate scenarios, wherein the edge nodes are susceptible to attack at any time, modifying such chains is another essential approach [219]. Network and I/O bandwidth needs are greatly increased due to

the necessity of revalidating blocks and copying chains from the primary network. The majority of frameworks additionally use a master–slave architecture, which introduces a potential weak spot. In diverse settings, this is to be expected. The balance between cost and reliability must be meticulously evaluated when considering redundancy [132]. The blockchain flaws also continue to impact fog architectures. There is a need to develop efficient consensus techniques that can validate blocks with little block sharing and copying. Those curious might learn more about blockchain by reading an in-depth report on the topic.

*4.3.2.2.2. Federated learning* Data is needed for ML model training, testing, and validation. Information is stored in locations accessible by thousands or millions of users (devices). Rather than sharing the entire dataset required to train a model, federated devices only communicate the parameters specific to that device's instance of the model. The parameter sharing mechanism is defined by the federated learning topology [220]. Each participant in a centralized topology contributes the parameters of the model to a centralized server, which then trains the centralized model and returns the trained parameters to each participant. Parameters are typically shared among a smaller group of peers in other configurations, including peer-to-peer or hierarchical ones. ML methods that require large or geographically dispersed data sets may benefit from federated learning. However, there is no universally applicable machine-learning solution [221]. Several unanswered questions remain about federated learning that researchers and developers are hard at work trying to address [222,223]. There are a lot of opportunities for efficient communication in federated learning. This means the master server or other entities acquiring the parameters must be able to cope with occasional interruptions or delays in transmission. Getting all the federated devices to talk to each other and stay in sync is still an open issue [222]. There is typically a lack of transparency between federated parties and a central server regarding the computing capacity of the federated parties. However, it is still challenging to ensure that the training activities will operate on a diverse mix of devices [220]. Federated parties' data sets might be quite varied in terms of data amount, reliability, and variety [224]. It is challenging to predict how statistically diverse the training data sets will be and how to protect against any detrimental effects this diversity may have. Efficient deployment of privacy-enhancing solutions is required to prevent data loss due to shared model parameters.

*4.3.2.2.3. Bitcoin currency* Transaction settlement using blockchain technology was initially proposed with the digital (crypto-) currency Bitcoin. The blockchain is a distributed ledger that verifies monetary transactions using PoW and may be configured to record anything of worth. Blockchains, including bitcoins and cryptocurrencies, are innovative in operating apps across networks [225]. Designers create smart contracts for Bitcoin money exchanges, which are subsequently carried out on blockchain VMs [226].

Blockchain relies on a decentralized, concurrency-agnostic runtime environment and consensus mechanism. Blocks of data may be disseminated across Bitcoin ledgers via a peer-to-peer network with no requirement for a centralized authority, thanks to the Bitcoin enabling network [226]. The data in the blockchain is certified by the members to keep it safe and open, and anybody is welcome to join the network. Cloud computing may use this property, and the security of cloud storage, in particular, can benefit from it. Cloud computing infrastructures enable the execution of complex applications and the handling of massive data sets. Centralized data centers coupled with Fog or IoT devices at the network edge cannot efficiently handle the enormous data storage required to deliver high-availability, real-time, low-latency services [227].

A distributed cloud design is required to deal with these problems instead of the more conventional network architecture. Blockchain technology, a fundamental element of distributed cloud systems, offers detailed control over resources by enabling their management through distributed apps [228]. It also allows for the tracking of resource usage, providing both customers and service providers with the means to verify that the agreed-upon QoS is being met. A marketplace is a platform where everyone may promote their computer resources while discovering what they require using AI-based techniques or models of prediction [229]. Blockchains, compared to cloud computing, offer fewer computer resources available to run distributed applications, such as less space for storing data, less powerful VMs, and a more unstable protocol. As a result, apps that are sensitive to delay and those that use a lot of resources need to find solutions to these problems [230].

Combining blockchain and cloud computing to develop a block chain-based distributed cloud can provide novel advantages and solve current restrictions. Data moves closer to its owner and user through Blockchain's distributed cloud, providing on-demand resources, security, and cost-effective access to infrastructure [231]. In the meantime, the high price and substantial consumption of electricity from clouds may be solved with a blockchain-based distributed cloud. Cloud storage security is another area where blockchain may play a role in the future [232]. By dividing user data into smaller pieces before storing it everywhere, it is possible to encrypt it further. A small portion of the data is accessible to the hacker, not the entire file. In addition to eradicating data-altering hackers from the network, a backup copy of the data may be used to restore any changes [229]. The use of quantum computers to circumvent the mathematical impossibility of modern encryption is one of their most publicized uses. In the meantime, many online publications have predicted the end of Bitcoin and other cryptocurrency use after Google stated it had achieved quantum supremacy.

*4.3.2.3. Trends/observations.* The main trends and observations regarding decentralized computing are as follows:

*Serverless Edge Computing*: Serverless' 'scale-to-zero' feature, which releases unoccupied containers from the system, works well for energy-conscious IoT scenarios with load-inconsistent applications. On the other hand, fine-grained scaling (i.e., at the function stage) is capable of handling extremely distinct needs and execution settings at the edge [185]. Many IoT applications rely on instances initiated by sensing or actuating, just like functions in serverless [102]. However, unlike serverless functions, IoT devices often sense or act only on rare occasions, whereas they sleep the majority of the time to conserve power. So, first, serverless appears to be an ideal paradigm of execution. However, combining serverless, edge computing, and IoT applications is challenging because serverless was originally designed for cloud environments, which do not have the same constraints as edge computing devices [233]. In light of this opportunity, it is essential to combine serverless, edge computing, and IoT applications to address this challenge. This is crucial to be addressed, as the fact is that although this adaptation looks needed and helpful, its practicality necessitates comprehensive inspections to avoid ramifications.

### *4.3.3. Hybrid computing*

It involves combining both a centralized network and a decentralized network. In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within hybrid computing.

*4.3.3.1. Focus/paradigms.* The following are the main focus or paradigms for hybrid computing:

*Fog-Cloud-Edge Orchestration*: Increasingly, IoT technologies are required in daily life. Smart cities, automated manufacturing, virtual reality, and autonomous cars are just a few instances of the vast variety of sectors where the application of these technologies has been rising quickly [234]. This type of IoT application frequently necessitates access to heterogeneous distant, local, and multi-cloud compute resources, in addition to a globally dispersed array of sensors. The expanded Fog-Cloud-Edge orchestration paradigm is born from this.

This new paradigm has made it a necessity to expand application-orchestration needs (i.e., self-service deployment and run-time administration) beyond the confines of a purely cloud-based infrastructure and across the full breadth of cloud or edge resources. Recent years have seen an increased focus on the research and advancement of orchestrating platforms in both business and academic settings as a means of meeting this need.

*4.3.3.2. Technologies/impact areas.* The key technologies and affected domains for hybrid computing include:

(1) *Cryptocurrencies*: Decentralized networks with powerful computational power were pioneered by cryptocurrencies. There is no centralized authority that controls the cryptocurrency market or issues new cryptocurrencies. Bitcoin, the first decentralized digital currency, was launched in 2009 and employs blockchain technology to record transactions and save user histories [235]. Blockchain Explorer and similar tools reveal Bitcoin's decentralized network activity as it moves from one wallet to another, and they also reveal the activity of other cryptocurrency networks. There is no equivalent technology that would enable such transparency in the private banking business, nor would such a publication ever be made public. Decentralization design incorporates many additional features that make it hard for bad actors to forge bitcoin or steal from user accounts, such as synchronizing the blockchain across all machines on the network [236]. Bitcoin and other cryptocurrencies are required to function on decentralized networks: A blockchain does not have a central controlling computer or administrator.

(2) *Machine Economy*: The emerging machine economy refers to the exchange of resources (such as power, data storage, processing power, currency, and network connections) in the upcoming global networks of computers [237]. Together, the data centers that power the cloud, the web, and monetary exchanges, form a network that will support the machines that power the future economy. This is the time when AI willfully conceals or exaggerates its powers. AI conceals and safeguards limited supplies to protect the crucial scarce resource of computation cycles used to generate AI insights. The organization is guarding the computation cycles used to generate AI insights, which are the most crucial scarce resource in this case. Lies, trickery, and barter to coax AI into parting with its limited resources will become an increasingly hot issue in the coming years [238]. To prevent itself from being overused, AI will have to resort to dishonest behavior. The machine economy is going to be among the most significant developments to come for human culture; and will be among the hottest topics of the emerging payment and AI technologies needed to fund future interstellar and interplanetary travel.

*4.3.3.3. Trends/observations.* The main trends and observations regarding hybrid computing are as follows:

*Distributed Computing Continuum*: Emerging from the convergence of IoT, edge, fog, and cloud computing, Distributed Computing Continuum Systems (DCCS) represent a novel computing paradigm that harnesses the collective power and heterogeneity of these diverse computing tiers to address the demanding computational requirements of future applications [239]. These applications, ranging from autonomous vehicles and e-Health to smart cities, holographic communications, and virtual reality, demand unprecedented levels of computational power, low latency, and efficient data management. Achieving these stringent requirements necessitates seamless integration and collaborative operation among all computing tiers, transforming the underlying infrastructure into a unified, intelligent system. As exemplified by edge and fog computing, the underlying infrastructure of DCCS plays a pivotal role in determining its performance. This geographically distributed, heterogeneous, and resource-constrained infrastructure poses significant challenges, needing new approaches that can dynamically adapt to application and user demands [9]. Cloud-centric methodologies, often tailored to cloud-specific assumptions, fall short in addressing the characteristics of edge, fog, and DCCS environments.

To address these challenges, DCCS advocates for decentralized intelligence, empowering each component of the underlying infrastructure to make autonomous decisions based on its specific tasks and local conditions [240]. This approach leverages the concept of service level objectives (SLOs), well-established in cloud computing, to define the operational goals of each component of the system. By modularizing and distributing SLOs across the system, a DCCS can achieve scalable intelligence within its infrastructure. Further, incorporating the Markov Blanket concept into SLO management enables causal filtering, ensuring that only conditionally dependent variables are considered when making decisions. This selective filtering, coupled with causal inference or active inference, empowers each component to make informed decisions independently, adapting to its dynamic environment and the overall system's requirements [241]. This loosely-coupled architecture fosters a resilient and adaptive DCCS, capable of catering to the diverse and evolving demands of future applications.

### 4.4. Computational methodologies: Parallel vs. Sequential computing

Parallel computing implies a computer model wherein numerous tasks are completed concurrently, employing a number of processors or threads [242]. In this paradigm, many processes run concurrently and their outputs are pooled. Tasks can be conducted in parallel instead of sequentially, potentially reducing execution times.

#### 4.4.1. Parallel computing

In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within parallel computing.

*4.4.1.1. Focus/paradigms.* The following are the main focus or paradigms for parallel computing.

*Simultaneous Data Processing*: In order to handle many parts of a task at once, parallel processing employs multiple processors, or CPUs. By breaking down large computations into smaller ones, systems may drastically speed up their execution [242]. Parallel processing is possible on current computers with multiple cores and on any machine with more than one CPU. Multi-core processors are embedded processors containing two or more CPUs for increased performance, lowered energy use, and more efficient handling of many tasks. Two to four cores are common in modern computers, with some models supporting up to 12. Modern computers commonly use parallel processing to complete complex processes and calculations. At the most basic level, sequential and parallel-serial processes differ in how registers are employed. Shift registers work in series, computing every bit one at a time, while registers with concurrent loading handle each bit of a word concurrently [243]. Using multiple functional units that can execute identical or distinct tasks in parallel enables the management of more complex parallel processing.

*4.4.1.2. Technologies/impact areas.* The key technologies and affected domains for parallel computing include:

(1) *ASICs*: Application-Specific Integrated Circuits (ASICs) are integrated circuits designed for specific uses. As their name suggests, ASICs are limited to a single function. They provide a single function and are consistent throughout their service life [138]. ASICs are semiconductor devices and circuitry developed to carry out a particular task. In contrast to mainstream processors, including CPUs and GPUs, both the speed and the energy efficiency of ASICs are optimized to fit the needs of a specific application [244]. Their excellent performance, minimal energy use, and small form factor make them ideal for mass-produced goods that can afford the higher bespoke design costs.

(2) *FPGA*: A Field Programmable Gate Array (FPGA) is a semiconductor that can be programmed to provide unique logic for use in both early system prototype design and the last version of a system to circumvent obsolescence [138]. In contrast to other bespoke or semi-custom integrated circuits, FPGAs can be easily reprogrammed by a software update to meet the changing requirements of the larger system they are integrated into, using hardware design languages, such Verilog and *Very High-Speed Integrated Circuit Hardware Description Language (VHDL)* [245]. Nowadays, most rapidly expanding applications are perfect fits for FPGAs, which include edge computing, AI, network security, 5G, industrial control, and automated machinery.

*4.4.1.3. Trends/observations.* The main trends and observations regarding parallel computing are as follows:

(1) *Neuro-symbolic AI*: Advances in deep learning techniques have unlocked a few of AI's enormous possibilities. Consequently, it is now obvious that these methods are at a breaking point and that such sub-symbolic or neuro-inspired solutions only function effectively for particular kinds of issues and are typically opaque to both analysis and comprehension [246]. However, symbolic AI methods, founded on rules, logic, and reasoning, perform significantly better in terms of openness, comprehensibility, authenticity, and reliability than sub-symbolic methods. A new path termed neuro-symbolic AI was recently recommended, integrating the effectiveness of sub-symbolic AI alongside the visibility of symbolic AI [247]. This synergy has the potential to yield a new generation of AI devices and platforms that are both comprehensible and expansion-intolerant and can combine logic with learning in a generic fashion.
(2) *Scalability*: The most important advantage of scalable design is improved efficiency, as well as the capacity to deal with sudden spikes in traffic or severe loads with little to no warning [248]. An application or online company may continue to operate smoothly during busy periods with the assistance of a scalable system, preventing businesses from incurring financial losses or suffering reputational harm [173]. If a system is organized into component services (for example, using the microservices system design), monitoring, updating features, troubleshooting, and scaling may become simpler tasks.

### 4.4.2. Sequential computing

In this section, we discuss the main focus or paradigms, technologies or impact areas, and various trends or observations within sequential computing.

*4.4.2.1. Focus/paradigms.* The following are the main focus or paradigms for sequential computing:

*One-process-at-a-time execution*: In the context of computing, sequential computing describes a paradigm in which operations are carried out in a certain order, with the output of one operation feeding into the data being the input of the subsequent one [249]. A single processor carries out all of the model's tasks in the sequence specified by the code.

*4.4.2.2. Technologies/impact areas.* The key technologies and affected domains for sequential computing include:

*Traditional Von Neumann Architecture*: This architecture is a sequential computing-based concept for digital machines. This system includes a CPU, RAM, and I/O devices, all interconnected by a bus [250]. The CPU of a system based on the Von Neumann architecture processes instructions sequentially, feeding the output of one into the input channel of the subsequent one [107].

*4.4.2.3. Trends/observations.* The main trends and observations regarding sequential computing are as follows:

(1) *In-Memory Computing*: In-memory computing is a method used to perform computations solely in memory (like RAM). This word usually refers to massive and complicated computations that must be executed on a cluster of computers using specialized systems software [249]. As a clustering system, the machines pool their RAM, so the computation is effectively done across machines and uses the combined RAM capacity of all the machines collectively.
(2) *Energy-efficiency*: Power effectiveness and sustainability have emerged as major issues for HPC systems as their processing capacity increases [251]. To reduce electrical usage while increasing computational performance, scientists are inventing environmentally friendly hardware layouts, investigating innovative cooling strategies, and fine-tuning algorithms. The general efficiency of HPC systems is being improved by the development of energy-aware scheduling and utilization strategies.
(3) *Performance Optimization*: Since single-processor efficiency can no longer develop at a rapid pace, the era of the single-microprocessor computer is coming to an end. It is time for a new era in computing when parallelism takes center stage and sequential computing takes a back seat [252]. There are still significant scientific and engineering obstacles to overcome, but now is a good moment to try new approaches to computer programming and hardware design. Various computer architectures have emerged, each tailored to certain performance and efficiency goals. The next wave of discoveries will certainly necessitate enhancements to computer hardware and software [253]. No one can say for sure if we will succeed in making parallel computing as mainstream and user-friendly as yesterday's peak sequential single-processor computer systems in the field of computing. Innovative novel applications that motivate the computer business will slow down if parallel programming and associated software activities do not become popular, and if creativity slows down across the economy as a whole, many other sectors will suffer as well [121].

## 4.5. Computing trends and emerging technologies

New computing trends and emerging technologies continue to advance the field of computing, improving the adaptability, self-management, and sustainability of many types of industrial systems.

### 4.5.1. Advanced computing styles and trends

In this section, we discuss advanced computing styles and trends and their related technologies and paradigms.

*4.5.1.1. Focus/paradigms.* The following are the main focus or paradigms for advanced computing styles:

*Quantum AI*: Quantum computing is attractive because it is a unique innovation that can radically change AI and computing in general. In this section, we look into what quantum computing can do and how it can affect AI and the wider economy. The implications of this computing method might have far-reaching effects on several facets of our cultural and financial lives [4]. The widespread impact of AI suggests that combining it with quantum computing might unleash dramatic change in the field of AI [198].

Several algorithms that made it possible to do tasks previously thought impossible for conventional computers emerged in the wake of the foundational studies that formalized the notion of a quantum computer [254]. The development of Shor's algorithm, an effective method for dividing enormous amounts of data, has bolstered research into quantum computing and quantum cryptography. Yet, existing

cutting-edge technologies are not yet accurate enough to execute Shor's algorithm successfully, which requires a degree of precision for performing register initialization, quantum operations on multiple qubits, and storing quantum states. It is also crucial to remember that quantum computers have particular limits [76]. The acceleration afforded by quantum computers grows exponentially compared to the amount of time a conventional computer takes (Grover's method); hence, it is not predicted that it will effectively solve NP-hard efficiency issues. The benefits of quantum computing, such as quantum superposition and entanglement, typically vanish rapidly with the complexity and magnitude (i.e., the number of quantum systems involved) of the underlying hardware, making the process of designing a quantum computer non-trivial. Despite this, the curiosity of significant technologically advanced players (IBM, Microsoft, Google, Amazon, Intel, and Honeywell) has skyrocketed in the past few years, and a plethora of fresh startups have emerged to propose remedies for quantum computing using technologies as diverse as superconducting devices, encased ions, and integrated light circuits. Corporations like these are among the numerous that are investing in quantum research and development at the moment [255].

Although there are many obstacles to overcome, the Google AI team has achieved considerable strides in the past few years, gaining a quantum edge by developing Sycamore, a programmable quantum computer. Similarly, IBM has now launched the Eagle chip, the first quantum computer with more than 100 qubits of hardware [256]. This is only the beginning of an intensive research and development program, with the tech giant hoping to increase the number of qubits to over a thousand by 2024 [51]. But as was previously stated, protecting these devices from ambient noise is a significant constraint when trying to retain the subtle characteristics of composite quantum states while still allowing for coherence in quantum development. Because of this, a quantum computer's components require ultra-low temperatures in the order of fractions of a Kelvin, which presents hurdles for both device design and material development [257].

*4.5.1.2. Trends/technologies.* The main trends and technologies regarding advanced computing styles are as follows:

*(1) Edge AI*: Recent advancements in AI efficiency, the rise of IoT devices, and the emergence of edge computing have all unleashed the promise of edge AI. This has opened up previously unimaginable uses for edge AI, such as helping radiologists make diagnoses, assisting in driving cars and fertilizing crops [92]. Since its inception in the mid-1990s—paired with the emergence of content delivery networks that utilize edge servers positioned near users to stream online and gaming video—edge computing has been the subject of much discussion and adoption by professionals and businesses. Almost every sector today has tasks that may benefit from adopting edge AI.

In truth, edge applications are driving the next generation of AI computing, which will improve people's lives in various settings, such as at home, at work, at school, and on the road. AI at the edge refers to the application of AI to physical devices. In contrast to storing all of an organization's data in a single centralized spot, such as a cloud provider's data center or a private data warehouse, "Edge AI" allows for AI calculations to be performed close to the users at the network's edge. Because the Internet is accessible all across the globe, any area might be thought of as its outskirts. Omnipresent traffic signals, autonomous equipment, and mobile phones are just a few examples. It might also be anything from a shop to a factory to a healthcare facility. Companies of all sizes strive to automate more of their processes because doing so improves productivity, effectiveness, and safety [258]. Computer software may aid with this through the ability to recognize patterns and dependably carry out identical tasks repeatedly. However, it is challenging to fully convey them in a system of algorithms and regulations because the world is unpredictable and human actions cover infinite circumstances. Today, as edge AI has progressed, robots and devices can work with the "intelligence" of human cognition no matter what they are. Intelligent IoT apps driven by AI may learn to adjust to novel circumstances and effectively complete identical or similar tasks [259]. Substantial progress in important areas has allowed for the practical deployment of AI models at the edge.

Furthermore, developments in neural networks, along with other areas of AI, have laid the groundwork for universal ML [260]. Many companies are finding that they can successfully train AI models and put them into action at the edge. AI in the periphery requires widely distributed computing resources. Recent advancements in enormously parallel GPUs are currently used to run neural networks. The development of devices connected to the IoT is partly responsible for the present age's unparalleled surge in data volume [261]. The development of sensors, smart cameras, robots, and other data-gathering equipment has made it possible to begin using AI models at the edge in nearly all facets of business. The increased speed, dependability, and security that 5G/6G is delivering to the battleground are also helping IoT use cases [118].

*(2) Biologically-inspired Computing*: The term "bio-inspired computing" refers to creating computer systems by drawing inspiration from the natural world. As an aside, computer science is also used to model and understand biological processes [145].

Computing architectures that take cues from nature can function as autonomous, flexible networks. Similarly, bio-inspired computing offers a fresh perspective on AI by building modular, self-improving systems [262]. Swarm intelligence refers to the ability of swarms of autonomous entities to generate intelligence by collaborating in ways reminiscent of the behavior of bees or ants. Biologists, software engineers, computer scientists, physicists, mathematicians, and geneticists all work together on the subject of bio-inspired computing [263]. Compared to their digital counterparts, biological systems have several distinct benefits. AI has advanced thanks to incorporating many concepts originally derived from natural processes into machine learning. Adaptable and responsive autonomous robots might be extremely useful in high-risk settings like conflict zones and hazardous clean-up activities [146].

Tasks like crop pollination might be performed by swarms of small robots. Bio-inspired technology is being used in cognitive modeling by developing artificial neural network systems based on neuron function within the brain. Training, growing, and collaborating on computer chips is becoming a reality [264]. When these nodes are linked by self-organizing wireless links, they form a system well adapted to modeling issues with several basic causes [263]. Self-learning and reconfigurable chips mean less time spent loading software and more time spent getting things done. Such systems might help explain the propagation of ideas through a community or construct a model of brain function that reflects true biological processes.

The use of DNA in natural computing is a topic of current study. Data storage, covert messaging, and even computation are all possibilities that have been proposed by DNA bioinformatics studies DNA [265]. DNA molecules may also form practical structures by self-assembly. The computer hardware, such as switches, CPUs, and timers, might be replaced by biological components. It is already possible to employ some biological substances in electronics. Even internal cell programming for purposes like medication secretion is feasible.

*(3) Explainable Artificial Intelligence (XAI)*: Successful completion of computer engineering tasks depends on wise decision-making. Can workloads be reliably executed on an automated system? Is there any way to understand how the trained models came

to their conclusions? Problems like this are typical and must be solved until any computer can be used in action [4]. Incorrect decision-making about such complicated and cutting-edge technology is costly in terms of resources and money. Many AI/ML implementations in computer systems have improved resource utilization and energy usage through better decision-making. However, the forecasts made by these AI/ML models for computing devices are still not usable, interpretable, or implementable. Such restrictions are a common problem for AI/ML models [266]. Most current research has focused on clarifying how QoS is accomplished, even though QoS remains a top priority. Is there anything academics can do to help the IT industry move forward? Therefore, when attempting to make educated judgments on handling resources (a prime manifestation of AI for computing), a solid grounding in Explainable AI (XAI) and experience with XAI methods and tools is required [267]. Forecasting of resource and power consumption and SLA variances, as well as the implementation of promptly proactive action to resolve these concerns, are examples of the types of Explainable AI techniques that may be used. XAI forecasting algorithms must be correctly developed to make computing more practical, explicable, and deployable [268].

(4) *Semantic Web and Decentralized Systems Integration*: Fog computing has emerged as a software engineering culture and practice that combines at least five different technology types: IoT, AI, Cloud-to-Edge Computing, Blockchain, and Digital Twins [269]. Various recent projects have presented their vision of integration between the Semantic Web and decentralized systems, for example, networks based on Blockchain technologies [270]. Here, the main challenge is to achieve a new generation of trustworthy, sustainable, human-centric, performant, and scalable smart applications.

(5) *Quantum Internet*: It is an ecosystem enabling quantum devices to communicate and share data in a setting that uses quantum physics' peculiar rules. In principle, this would grant the quantum Internet hitherto unattainable skills via standard web apps [59]. Quantum devices, such as a quantum computer or a quantum processor, may generate the quantum states of qubits, which can then be used to encode information. Sending qubits over a network of physically distinct quantum devices is, in essence, what the quantum Internet will be all about. Importantly, this will occur because of the strange characteristics of quantum states. That probably sounds like the conventional web [271]. However, if one wants to transmit qubits, then they need to use a quantum channel instead of a conventional one, which requires exploiting the peculiar behavior of quantum particles used to encode information onto quantum states. That requires to build up, and apply, relatively novel (or exotic) knowledge on the top of what is known about classical computing to effectively drive the possible evolution of quantum ecology into an effective quantum internet [272] [273] [254]. One could imagine that their favorite web browser will not have much in common with the quantum Internet [4].

### 4.5.2. Industry and sustainability trends

In this section, we discuss industry and sustainability trends and their related technologies and paradigms.

*4.5.2.1. Focus/paradigms.* The following are the main focus or paradigms for industry and sustainability trends:

*Carbon-Neutral Computing*: The expansion of the computer age is an important factor in the data center industry's advancement; however, the push towards carbon neutrality is a more dramatic paradigm change and the industry's biggest challenge to date. Large-scale cloud providers have pledged to attain zero emissions on all initiatives by 2030 [274]. The fight against climate change must include data centers. Everything from everyday conveniences like Internet banking and shopping to cutting-edge technologies like machine learning, quantum technology, and autonomous vehicles would be impossible without them. There is no denying of the ever-increasing need for data centers. Nevertheless, because of the damage they cause to the natural world, they also attract greater scrutiny [190]. A sustainable future with a zero-carbon footprint is possible because of these advancements in electricity, water effectiveness, and land utilization. Online conferences and handheld gadgets make it feasible for individuals to work from their homes and cut transit carbon emissions; however, each bit of data has a carbon footprint of its own [192]. Therefore, whereas electronic devices provide opportunities to enhance our oversight of water and materials and to support sustainable economic growth, simply sending a message provides for the challenging environmental impact of data. However, this may differ greatly depending on the spot and efficiency of the data centers that deal with traffic [193]. Crucially, as globalization brings online amenities to more societies, physical infrastructure, such as data centers, must grow to accommodate an increase in consumers, a majority of whom will be in regions around the globe that currently lack access to green power availability.

*4.5.2.2. Trends/technologies.* The main trends and technologies regarding advanced computing styles are as follows:

(1) *Industry 4.0*: The Fourth Industrial Revolution, or Industry 4.0, reshapes how goods are made, enhanced, and disseminated. Emerging innovations such as the IoT, cloud computing, analytics, and AI/ML are being incorporated into manufacturing facilities and processes [275]. Advanced sensors, software with embedded capabilities, and robots are used in these "smart industries" to gather information for more informed decision-making. When data from manufacturing operations is combined with data from Enterprise Resource Planning (ERP), supply chain, customer service, and other corporate systems, information that was previously kept separate can be seen and understood in completely new ways, which leads to even more value being created [276].

Improved efficiency and responsiveness to clients is made possible by the advent of technological innovations such as enhanced automation, predictive maintenance, and automatic optimization of process enhancements [277]. To enter the fourth industrial revolution, the manufacturing sector must embrace the development of smart factories. The ability to see industrial assets in real-time and access preventative maintenance tools may be gained by analyzing the massive volumes of big data generated from sensors on the production line. Smart factories implementing cutting-edge IoT technology see gains in output and quality [278].

Manufacturing inaccuracies and costs can be reduced by using AI-powered visual insights instead of traditional business models for human inspection. Quality assurance staff may monitor production operations from almost any location with minimal expenditure using a smartphone linked to the cloud. Companies may save money on costly repairs by identifying problems early on with the help of ML algorithms [49]. Any business operating in the industrial sector, from individual to process production and even in the energy and mining industries, may use the ideas and tools of Industry 4.0.

(2) *Digital Twins*: A digital twin is a computerized model of and connected to a real-world object that may be used to test and improve its design, performance, and usability [279]. Smart sensors embedded in the object capture data in real-time, allowing a digital depiction of the asset to be produced [99]. The model may be used through an asset's lifespan, from development and testing to actual usage, revamping and eventual retirement. To create a digital representation of a physical object, digital twins

utilize many technologies. The term "IoT" describes the network of interconnected devices and the underlying infrastructure that enables them to exchange data and instructions with one another and the cloud as a whole. With gratitude to the introduction of affordable computer chips and high-bandwidth connectivity, one can now have trillions of gadgets hooked up to the global web. Digital twins use data from IoT sensors to replicate physical properties in a virtual form [280]. The information is sent into a system or panel to be viewed as it changes in real time. Studying, solving issues, and pattern recognition are just a few examples of the kinds of cognitive challenges that AI seeks to address [281]. AI/ML-based algorithms and statistical models let machines do tasks with little to no human help. They do this by relying on patterns of observation and inference. Machine learning techniques used in digital twins process enormous amounts of sensor data, allowing for the identification of data patterns. Optimization of performance, servicing, emissions outputs, and efficiency may all be gained using data insights provided by AI/ML [282]. There are several significant distinctions between digital twins and modeling: even though both leverage virtual model-based simulations, a digital twin maintains a two-way connection and can affect the physical object. Offline optimization and the design process are two common applications of simulation. Developers use simulators to test out different iterations of a product. On the contrary, digital twins are interactive and dynamically updated virtual worlds. Both their scope and their utility have increased.

#### 4.5.3. Adaptive and self-managing systems

In this section, we discuss adaptive and self-managing systems and their related technologies and paradigms.

*4.5.3.1. Focus/paradigms.* The following are the main focus or paradigms for adaptive and self-managing systems:

*Autonomic Computing*: IBM's autonomic computing program was one of the earliest worldwide efforts to develop computing systems with little human intervention required to accomplish predetermined goals [30]. It was primarily based on findings about how human nerves and thinking work and how they are coordinated—bioinspiration, as discussed above. In autonomic computing, researchers explore how software-intensive systems can make decisions and act without human interaction to reach the (user-specified) "administration" objectives [283]. The concept of control for closed- and open-loop systems has significantly impacted the foundations of autonomic computing [31]. Multiple independent control networks may coexist in practice inside complex systems. The integration of ML and AI to enhance resource utilization and efficiency at scale remains an important obstacle regardless of investigations into autonomic frameworks to handle computing resources, from a single resource (e.g., a web server) to resource groupings (e.g., several servers inside a CDC) [4]. Autonomous and self-managing systems can be implemented on a spectrum from fully automated to partially automated with human oversight through the use of AI/ML to improve the efficiency and performance of the computing systems.

*4.5.3.2. Trends/technologies.* The main trends and technologies regarding adaptive and self-managing systems are as follows:

*SDN-NFV*: The explosion of IoT devices and the concomitant flood of sensor data enable knowledge-driven IoT applications, including connected cities and smart agriculture [84]. To begin providing such services, one must develop a data-gathering method that is flexible enough to adapt to shifting conditions in the field. Network programmability (SDN or NFV) enables the easy reconfiguration of IoT networks [86]. Current SDN/NFV-based approaches in the IoT environment nevertheless fail owing to a shortage of knowledge of resources and overhead, as well as incompatibility with conventional protocols [1]. This void must be filled by prioritizing resource and power limitations in the creation of SDN/NFV-enabled IoT nodes and network protocols. Assigning traffic sources to those Virtual Network Functions (VNFs) across the most efficient paths, with sufficient energy and network reliability, may maximize the number of active NFV nodes [9].

**Summary:** Table 3 lists a summary of open challenges and future directions in Paradigms/ Technologies/ Impact Areas, along with recommendations for further reading. Table 4 lists the summary of Trends/Observations for modern computing along with the recommendations for future reading.

## 5. Impact and performance criteria

In this section, we discuss the impact of contemporary computing and performance criteria.

### 5.1. Performance metrics

We are considering QoS, SLA, autoscaling, and fault tolerance as performance metrics for computing systems.

#### 5.1.1. QoS and SLA

Predicting how a cloud computing system will work in real-time is a major difficulty, even if AI techniques are used [284]. The efficiency of a computer may be measured using QoS metrics, including execution time, cost, scalability, elasticity, latency, and dependability. A SLA, a legally binding contract between a cloud service consumer and provider, defines QoS standards and potential penalties should they be violated [285]. Today, various IoT applications can use blockchain and similar technologies. Each one has its own QoS factors that depend on its area, goal, and demand [286]. An SLA may also be assessed with a metric called SLA violation rate, which determines compensation in the event of an SLA breach by estimating the divergence of the real SLA compared to the needed (estimated or predicted) SLA [287]. Since compromized QoS in one cloud service may negatively impact the QoS of the entire computing system, QoS is becoming increasingly crucial while assembling cloud services. Provisioning the proper quantity and quality of cloud resources that will satisfy the QoS of an application's price range, response time, and deadline is essential for providing an effective cloud service [288]. Consequently, cloud providers should guarantee to offer sufficient resources to minimize or reduce the SLA violation rate, allowing users' workloads to be executed in accordance with their set time and cost constraints [289]. In that regard, the diversity of applications and their behaviors on different machines requires a tighter description of their needs to minimize SLA violation while not over-provisioning infrastructure [290]. QoS-aware resource management methods, which can determine and meet the QoS needs of a computing system, such as SLO-driven modeling and execution-reordering of web requests, are crucial to its success in the future [291]. Several research issues must be overcome before QoS can be attained effectively [292]. Initially, the execution time of an application is large, and its performance is diminished due to a lack of cloud resources during runtime—which can be compounded by transparent processes to the developer, such as garbage collection, magnifying the potential of inexplicable SLO violations [293]. Additionally, finding the requirement for effective SLA-aware resource management methods decreases the SLA violation rate and preserves the overall efficiency of the computing system. Finally, to reach the ultimate goal of having multiple clouds, there has to be a unified SLA standard across all cloud providers [294]. Since many IoT applications rely on cloud computing systems that employ AI-based supervised or unsupervised algorithms for learning or models for forecasting, it is imperative to determine the appropriate balance amongst various QoS needs.

**Table 3**
Summary of open challenges and future directions in Paradigms/Technologies/Impact areas along with further reading.

| Paradigms/Technologies/Impact areas | Open challenges and future directions | Further reading |
|---|---|---|
| Cloud Computing | What are the tradeoffs that need to be established between the various QoS requirements brought on by the large variety of IoT applications operating on cloud systems? | ACM CSUR [1] |
| Autonomic Computing | What additional problems may be addressed by an autonomic computing expansion that is based on AI/ML as the number of IoT and scientific workloads increases? | Elsevier IoT [4] |
| Mobile Cloud Computing | How would AI-based deep learning algorithms be used to anticipate the resource demands beforehand for diverse geographic resources needed for mobile cloud computing, requiring new strategies for provisioning and scheduling resources? | ACM CSUR [60] |
| Green Cloud Computing | How can improved methods for effective data encoding for lower bandwidth usage and energy-effective transmission in data-intensive IoT devices make cloud computing more environmentally friendly? | ACM CSUR [162] |
| Fog Computing | How can AI approaches be utilized to properly schedule tasks when working in locations with varying amounts of fog resources? | Elsevier JPDC [39] & IEEE COMST [41] |
| Edge Computing | In what ways edge computing can be utilized to boost power and resource utilization, hence enhancing QoS? | IEEE COMST [41,206] |
| Mobile Edge Computing | How can novel resource provisioning and scheduling policies be developed for mobile edge computing that makes use of AI-based deep learning approaches to forecast the resource requirements beforehand for resources that are located in different locations? | IEEE COMST [41,213] & ACM CSUR [60] |
| Serverless Computing | How to reduce the cold start time and increase scalability using serverless edge computing? | IEEE TSC [186] & ACM CSUR [183] |
| Osmotic Computing | How can osmotic computing improve resource availability or performance at the network edge while moving services from the data center to the edge for AI/ML-driven adaptive administration of microservices? | ACM TOIT [46] |
| Dew Computing | How should dew computing allow a highly scalable method that can increase or reduce the real-time demands of performing operations at runtime via utilizing AI? | Elsevier IoT [48] |
| Programming Models | How to select a programming model that efficiently gathers data when and where it is needed while keeping complexity low relative to the total number of processors at hand? | Procedia Computer Science [115] |
| Virtualization | How can unbreakable security for VMs be ensured if consumers do not follow recommended practices when it comes to login credentials, installations, and other operations? | ACM CSUR [122] |
| IoT | How to ensure that an SLA is upheld while responding to customer requests as quickly as possible using IoT applications? | IEEE COMST [78] |
| Integrated Computing | How may QoS characteristics change if communication between layers in a fog-edge/cloud computing paradigm is improved? | ACM CSUR [108] & Elsevier FGCS [112] |
| Connectivity/ Networking | How can satisfying the demand or need for network solutions enabling high performance, resilience, dependability, scalability, adaptability, and cybersecurity remain constant? | ACM CSUR [60] & IEEE COMST [61] |
| Container Technologies | How can the QoS in data processing be enhanced by leveraging containers with virtualization? | Springer JoS [174] & Wiley CCPE [178] |
| Microservices | How to handle errors, ensure data integrity, and communicate effectively amongst services in a distributed system using a microservice architecture? | IEEE TSC [172] |
| Software-defined Networks | What are some ways in which SDN might help minimize power usage in cloud and edge computing? | Wiley ETT [84] |
| Distributed Ledger Technology (Blockchain) | How can distributed ledger technology (Blockchain) be utilized to secure the data for IoT applications? | IEEE COMST [108,216] |
| Federated Learning | How could companies ensure privacy in federated learning services, which differ from learning in data centers in that users' data is disclosed to third parties or the centralized server while exchanging model changes during the training stage? | Elsevier KBS [222] & CIE [220] |
| Software Engineering | How can fault tolerance be improved in computing systems dynamically without manually writing the software code by utilizing AI to "automatically" diagnose and fix an error? | Elsevier JSS [129] |
| Distributed Computing Continuum Systems | How can Distributed Computing Continuum Systems consider all computing tiers as a single system and optimize future applications in a decentralized manner? | IEEE TKDE [239] |

### *5.1.2. Autoscaling*

Thanks to the dynamic nature of the cloud, self-adapting techniques may be used to reduce resource costs without compromizing QoS [295]. Resource autoscaling, or strategy, reconfiguration, and provisioning, allows for self-additivity. Scientists have looked into autoscaling, or the dynamic modification of computational resources like VMs, for several reasons [123]. These include the desire to learn more about (a) horizontal changes, or the addition or removal of VMs; (b) vertical transformations, or the addition or removal of VM resources; (c) choice-making techniques, such as analytical modeling, control theory, and neural networks; and (d) utilizing a range of pricing models, such as on-demand. When it comes to latency-sensitive QoS requirements, the primary challenge for autoscaling methods is figuring out how to make a scaling decision quickly enough. AI prediction is the initial step towards making decisions in the quickest way possible [248]. However, traditional ML may not be up to the task when it comes to IoT applications requiring real-time mistake correction due to a lack of autonomous error correction [296]. Also, the rise of latency-sensitive IoT apps and microservices that need responses in the range of milliseconds has made things worse while container-based solutions

**Table 4**
Summary of Trends/Observation for modern computing along with future reading.

| Trends/ Observation | Open challenges and future directions | Further reading |
|---|---|---|
| AI-driven Computing | How to optimize the management of resources using the latest AI/ ML models in computing systems? | Elsevier IoT [4] |
| Large Scale Machine Learning | How can businesses mitigate the risks associated with the proliferation of sensitive information that arise as a result of the proliferation of data produced by AI and ML systems? | IEEE TKDE [155] |
| Edge AI | What strategies should be employed to oversee the simulation and information transmission among peripheral devices and other systems? What network infrastructures should be utilized to enable this communication? | Elsevier IoTCPS [92] & ACM SIGCOMM [261] |
| Bitcoin Currency | How can computing be utilized to maximize the efficiency of computation or processing capacity usage in cryptocurrency for cloud mining? | Elsevier JNCA [226] |
| Industry 4.0 | How can AI, the cloud, and edge computing be used to do predictive analysis that involves company resources? | IEEE COMST [275] |
| Intelligent Edge | How to deal with big problems that come up when designing system-level, algorithm-level, or architectural-level developments or innovations for integrated cognitive ability, like making decisions in real-time, keeping AI training and inference environmentally friendly, and deploying protection? | IEEE COMST [88] |
| XAI | How can the forecasting of resource and power consumption and SLA variances, as well as the implementation of promptly proactive action, reduce SLA violations and enhance QoS using XAI? | ACM CSUR [266] |
| Exascale Computing | How to make energy-efficient computing as power-hungry as the supercomputers that do calculations and transfer data within the computing environment nowadays? | ACM CSUR [142] |
| 6G and Beyond | What role 6G may play in reducing latency and improving reaction times by transmitting data between edge devices at high speeds? | IEEE COMST [98] |
| Quantum AI | What steps should be taken to build the AI cloud-based quantum computing infrastructures that are expected to be the foundation for our usage of quantum computers and simulators, which will supplement our existing classical computing hardware? | Wiley SPE [51] |
| Quantum Internet | How can the benefits of quantum networking be preserved while integrating the quantum Internet into currently operating conventional technology that will have to exist alongside and communicate effortlessly with today's Internet services? | IEEE COMST [254] |
| Analog Computing | How is it that analog computers can do complicated computations faster and more accurately than their digital equivalents, which utilize ML methods? | Nature Electronics [146] |
| Neuromorphic Computing | How might neuromorphic systems, which model the brain's structure and function and use analog circuits to do AI tasks, pave the way for creating incredibly adaptable, self-learning machines? | Nature Computational Science [149] |
| Biologically-inspired Computing | What can researchers take away from brain cells concerning ways to minimize the energy needed for computation, AI, and ML, given that these cells can easily combine smaller tasks to execute larger ones? | Elsevier ESA [263] |
| Digital Twins | How can network digital twins aid in speeding up preliminary installations by preparing navigation, protection, digitization, and evaluation in simulation while offering the scalability and interoperability of complex networks? | IEEE COMST [280] |
| Net Zero Computing | How can companies mitigate the negative ecological impact of their IT infrastructure by constructing environmentally friendly data centers and improving energy effectiveness, given that these centers use significant quantities of electricity and release enormous quantities of waste heat while also providing powerful computing services? | IEEE COMST [190] |

and burstable efficiency resources should make it possible to deploy and provision resources in the cloud quickly. To prevent a potentially disastrous situation, a smart car's onboard computer constantly monitors data such as the vehicle's speed, the location of other drivers and passengers, and the road conditions [297]. The cloud alone cannot answer this problem due to the instability and latency in connections between the cloud and users; instead, autoscaling techniques for IoT applications must take these factors into account [298]. The truth is that autoscaling needs to be made bigger because the cloud naturally gets in the way of Industry 4.0 ideas, like real-time management, and making decisions without a central authority.

### 5.1.3. Fault tolerance

Providers of cloud computing services owe it to their customers to make such services available without interruption, regardless of what problems arise [299]. To meet the QoS standards of a computing system efficiently, fault tolerance approaches are employed. Software, hardware, and even networks may all go wrong when a computer system operates. In addition, fault resilience guarantees the reliability and accessibility of cloud services [4]. Timeout breakdowns, overload issues, and resource-lack failures are further examples of cloud dependability issues. A major breakdown has the potential to cause a cascade of failures in the system [300]. Several proactive and reactive fault tolerance approaches have been developed to cope with these kinds of failures. The most common method of handling faults in long-running processes is called "checkpointing", and it involves preserving the current state after each modification [301]. Additionally, checkpoints are employed if there is a possibility of not beginning at the same position [1]. Replication-based resilience is another well-known method; it involves duplicating the nodes or jobs until they are completed. If a system is overloaded or malfunctioning, a task migration-based resilience solution can move the work to another computer. Computer systems must have autonomous resilience-aware resource management technology, reliability of service methods, and reliable information integrity (e.g., blockchain) to keep running. Reliability impacts QoS in cloud computing while still delivering it effectively. One of the biggest obstacles in cloud computing is figuring out how to deliver a secure and effective cloud service while cutting down on power consumption and emissions [302]. Cloud computing has built-in redundancy to maintain service availability, QoS, and performance guarantees. Resource

management must consider varying failures and workload prototypes for medical care, urban planning, and agricultural applications to run well [71]. Predicting failure in systems that use cloud computing is difficult and can impact the dependability of the system [301]. Predicting faults and achieving the requisite dependability of the cloud service while maintaining QoS necessitates several machine or deep learning approaches [13]. Replication-based fault tolerance solutions are effective for IoT applications because they reduce task delay and response time. A dependable cloud storage system that will offer an effective retrieval system for processing big data is also required to deal with big data applications [303].

### 5.2. Efficiency metrics

We are considering energy consumption, carbon footprint, and serviceability as efficiency metrics for computing systems.

#### 5.2.1. Energy consumption

Data collection and processing have risen exponentially during the last several years. This pattern has been pushing cloud systems to the limits of their computational and, by extension, energy consumption capacities [304]. Annually, CDCs have increased their power use by around 20% to 25% [305]. This shift has led to the rise of decentralized computer architectures such as Fog and Edge. The latency and cost-effectiveness of cloud computing are all vastly improved by moving parts of its computation to distributed edge devices and networks. There nevertheless exist difficulties associated with this. Irregular energy supply, even without the power supply itself, presents significant issues for numerous highly critical and remote sensing applications.

The ever-growing number of IoT devices and the data they produce have put networking's ability to handle information, compute, and transfer data throughput to the test [162]. Meanwhile, smaller IoT devices are currently created with limited computing power, storage spaces, and energy. Hence, it is imperative to boost the performance of fog and edge nodes in the network. Sustainability in CDCs and minimizing their carbon impact have also become more pressing concerns. This must be accomplished without lowering the bar for QoS [306]. Notwithstanding the obstacles, there have been several advances in this area. Software, hardware, and transitional approaches have all been taken to the energy management problem.

Approaches and techniques are being designed to optimize software efficiency, supported by computational models [306]. One example is mobile edge computing offloading. Hardware-wise, particularly for the application, devices were designed to provide peak performance while minimizing energy consumption. Energy efficiency in Wireless Sensor Networks (WSNs) has been extensively researched [4]. Fog/edge-node sleep time scheduling, active resource management, and additional energy-saving strategies have all been used in the intermediate phase. There are still many unanswered questions and potential avenues for development when it comes to the effectiveness and longevity of fog, edge, and cloud infrastructures.

Advanced algorithms for encoding data into fewer bits are explored to reduce transmitter power needs, which are crucial due to limited transmission bandwidth, more critical than direct CPU power needs. Despite the need for specialized hardware, encoding methods may be used by taking advantage of the universal encoders present in virtually all mobile devices [13]. Yet, it has become impossible to lower the ideal bandwidth due to the rising quantity of data exchange and loss. Preparing for CPU and data utilization in a way that minimizes heat generation requires modeling at the transistor level, which necessitates the development of 3D thermal simulation systems [75].

Lastly, the aim is to minimize power consumption to the point that the CPU and transceiver may be powered entirely by energy harvesting or scavenging approaches [307]. Consequently, the Fog/Edge network's granularity may be decreased, leading to more widely scattered, overbearing, and resilient architectures. In various fields, like energy limits, blockchain algorithms might be studied with various versatile AI-based learning approaches for enhanced energy scheduling.

#### 5.2.2. Carbon footprint

End-user needs for applications and the resulting growth in storage in the Exabyte range will result in the first Exascale system by 2025, followed by a Zettascale system by 2035 [2]. While this is certainly something to be proud of, there are also many difficulties that come along with it. Keeping everything running requires massive amounts of energy, which poses a major obstacle. At the moment, over ten percent of the world's power is used each year by the ICT sector [190]. The rebound effect, which leads to even higher demand and consumption, makes it counterproductive to create ever-larger systems by increasing efficiency. The next generation of autonomous system paradigms will likely place a greater emphasis on power and carbon footprints in light of climate change and the projected 1.5 °C rise in worldwide temperatures owing to emissions of carbon dioxide by 2100 [2]. This is not merely about lowering energy use per unit of processing, as is the case now, but also about more basic issues with systems that assume continuous stable power supplies, connectivity with sources of clean energy, and alternate techniques of minimizing energy usage [308]. The study and treatment of systems as living ecosystems rather than as collections of discrete components is a topic of great interest, and this includes the comprehensive integration of managing energy (asynchronous computation, power scaling, wake-on-LAN, air conditioning, etc.).

#### 5.2.3. Serviceability/usability

The fields of human–computer interaction and networked systems have yet to fully merge with each other. This closer synchronization would be especially helpful for cloud computing [1]. Despite significant work on resource management and the back-end associated concerns, accessibility is a vital component in lowering the costs of organizations investigating cloud services and infrastructure. Costs associated with labor might decrease since customers will receive superior service and increase their output [309].

NIST's Cloud Usability model addresses five dimensions of cloud usability: capability, personalization, reliability, security, and value, all of which have been highlighted as critical issues [310]. The term "capable" refers to the degree to which cloud service can fulfill the needs of its customers. With the assistance of personal customization options, individuals and businesses will have the capability to modify the visual style and adjust or eliminate features from interfaces for various services. Trustworthy, robust, and useful are attributes associated with possessing a system that fulfills its duties throughout state situations, is safely protected, and delivers value to customers accordingly. Current cloud initiatives have mostly concentrated on wrapping up sophisticated services into APIs that can be accessed by end users [309]. HPC Cloud is the most evident example. To make HPC applications more accessible and easier to use, researchers have developed several different services. In addition to being packaged as services, these systems provide Web interfaces through which their settings may be set and their input and output files managed.

DevOps is another path associated with cloud usage that has gained popularity in recent years [311]. DevOps has increased the efficiency of both software engineers and administrators when it comes to developing and delivering remedies on the cloud. Cloud computing is important not only for creating brand new solutions AIOps and MLOps [312] but also, for streamlining the process of moving existing applications from onsite settings to adaptable, multi-tenant cloud services.

### 5.3. Social impact

We are considering the digital divide, ethical AI, and digital humanism as social impact metrics for computing systems.

#### 5.3.1. Digital divide

Corporations in rural areas have significant challenges due to the difficulty of gaining a connection to broadband connectivity and, by extension, cloud-based resources [313]. Access to the web is one example of a long-standing infrastructural gap between urban and rural areas. There are a lot of companies that cannot expand and innovate because they lack access to new technology. Businesses in rural areas face another obstacle: the high cost of maintaining and upgrading on-premises IT infrastructure. Cloud computing's main benefits are the ability to work together and think creatively. The cloud encourages teamwork by facilitating real-time, distributed collaboration. This greater collaboration encourages invention. As a result, rural enterprises may now compete on an equal basis with their metropolitan competitors [314]. Accessibility to data and fundamental information is also crucial. The benefit of using the cloud has increased significantly with the advent of generative AI. Comprehensive sales, marketing, and manufacturing capabilities are provided by core AI services, but these cannot be reproduced with human processing and can be too costly to install on-site for modest organizations. The proliferation of cloud computing has expanded business opportunities, but not equally. By utilizing the cloud, companies in rural areas may overcome the constraints of their physical location [315]. Cloud computing's greater availability, decreased cost, scalable effectiveness, and improved cooperation may breathe new life into the rural economy and propel it towards long-term success.

#### 5.3.2. Ethical AI

AI systems require vast amounts of data, including details on businesses and their clients [316]. The value of knowing the data owner surpasses that of having private information that cannot be linked to a specific person. When dealing with sensitive information, companies regularly face problems related to data security and regulatory compliance [317]. Autonomic computing using AI needs to take into account privacy rules and data protection. While AI has the potential to be a game-changer, it has not always been successful in achieving its aims. A hunt for answers by an AI may result in a flood of insensitive comments [318]. The vast number of AI decisions and the stakes involved make this field fraught with peril. Prior to expanding the use of this invention, it is crucial to develop accountability and ownership.

#### 5.3.3. Digital humanism

The unavoidable consequences of digital colonization driven by business need a counter-force of digital humanism motivated by care for humanity and the Earth [319]. We have never been both so interdependent, yet so isolated. Modern digital systems allow for global communication. One no longer has to be in the same room as someone else to have a conversation, collaborate on a project, or just have fun with them. The cell phone is rapidly becoming an integral part of people's daily lives all across the world. Connectivity between the developing world and the developed nations of the world is rapidly expanding, for both good and ill. These interconnections are causing conflicts that could have been prevented when individuals and ideas were separated by space. Western materialism and commerce meet Eastern spirituality and culture in the virtual world [320]. Therefore, although humans may all end up in the cloud at some point, the barriers of mutual respect and compassion that keep us from crashing into one another are more than frayed. Most modern digital accounting and tracking systems are used by private companies seeking to maximize profits at the expense of others, enriching a few elites at the expense of a much larger underclass [321]. In contrast, if the cloud could be utilized for humanity's benefit, manufacturing and distribution might be dramatically enhanced. Controlled well, such instruments will allow for fine-tuning of many crucial societal functions, particularly at the subnational and neighborhood levels.

### 5.4. Security and compliance

We are considering data protection, privacy regulations, and resilience to attacks as security and compliance metrics for computing systems.

#### 5.4.1. Security, privacy and resiliency

In recent years, there has been a dramatic change in academia and business towards the IoT, edge computing, and cloud computing in order to serve customers better. With this massive paradigm shift, comes a slew of problems and difficulties with protecting the confidentiality and safety of the information stored on these devices [322]. Edge computing's many distinguishing features – its low latency, geographical dispersion, end-device accessibility, high processing power, variability, etc. – make it imperative that security and privacy mechanisms be both flexible and powerful [323]. In addition, creating universally compatible software platforms is challenging due to the wide variety of use cases and device types.

Several elements become important in the research of these security and associated challenges in the cloud and fog computing models: End-user confidence and privacy; verification and validation of sources inside nodes; secure communications between sensor, compute, and broker nodes; detection and prevention of malicious attacks; secure, reliable and decentralized data storage, such as Blockchain [231]. Some of the problems that have already been addressed in this field include adaptive mutual authentication, identifying and retrieval of harmful or malfunctioning nodes, the detection and defense against assaults, the avoidance of harmful hazards, and the protection of user information from theft. Unmanned Aerial Vehicle (UAV)-aided computing devices can now maintain their anonymity while contributing to distributed frameworks in AI technology, such as computer vision and path learning, supporting data processing and decision-making [324]. Other efforts in fog forensics have also given digital evidence by recreating prior computer activities and identifying how these events contrast with cloud forensics in important ways.

The past few years have seen significant progress in several key areas related to Fog Radio Access Networks (F-RANs), including mobility management, interference reduction, and resource optimization [325]. Novel approaches have evolved for varied applications handling privacy challenges. Face recognition and resolution, vehicle crowd sensing, geographic location sensing and data processing, renewable node storage systems and data centers, and fog-based public cloud computing are promising new research areas. Prevention against data theft, attacks involving man-in-the-middle, confidentiality of users, location confidentiality, forward privacy, reliable user-level key management, and many other weaknesses have all been addressed through such efforts [4].

There are scaling issues with many fog/cloud privacy and security models that prevent them from fully applying to the next-generation edge computing transition [326]. Because of fog computing's decentralized nature, numerous new security concerns, which are not an issue in the cloud, emerge in the fog layer and IoT devices. The deployment of authentication systems is hampered by the prevalence of threats such as advanced persistent threats (APT attacks), malware, distributed denial of service (DDoS) attacks, two-way communication, and micro-servers without hardware protection mechanisms in edge data centers [327]. Additionally, these studies show how the mobile edge computing architecture might change in the future. For example, edge nodes working together could make real-time encryption more efficient. The computational capacity of both edge and distant resources has not been completely used in previous efforts, and security flaws have been addressed from a restricted viewpoint. New phenomena appear when cloud-like capacities are distributed to the network's periphery [231]. Edge data center collaboration, service migration on a local and global scale, end-user concurrency, QoS, real-time applications, load distribution, server overflow issues, stolen

device detection, and dependable node interaction are all examples of such scenarios. Future studies can focus on new areas, such as evolving game-theoretical strategies to the privacy algorithms encouraged by adversarial attack scenarios, communication protocols in sensor cloud systems, and clustering model-based security evaluation (AI-based forecasting approaches), which can be investigated as potential solutions to these issues [236]. Mobile devices' presence in these data centers should be taken into account by safeguarding systems.

### 5.5. Economic and management

We are considering cost-efficiency, resource allocation, application design, computing economics, and data management under economics and management for computing systems.

#### 5.5.1. Cost-efficiency

Minimizing cloud expenditures while maximizing application performance and efficacy is the goal of cloud cost optimization, which entails striking a fine balance between technological standards and corporate goals [304]. Cost-effective cloud computing refers to the practice of utilizing cloud providers in the most economical way feasible to operate software, complete tasks, and create value for a company. Optimization as a practice varies from fundamental business management to challenging scientific and technical fields including operational research, statistical and data analysis, and modeling and prediction [316]. Corporations may maximize the return on their investments in cloud computing through cost optimization, which reduces wasteful expenditures and strengthens their operational effectiveness [328]. By avoiding economic hazards, aligning spending with company goals, and establishing a secure, scalable, and cost-effective cloud infrastructure, corporations can maximize the return on their investments in cloud computing. In general, efficient cloud cost management preserves essential resources against the risk of unanticipated expenditures and financial mismanagement. Changing to a cloud-native methodology involves more than just updating technology; it also necessitates a substantial adjustment in mindset [1]. Building scalable apps that make efficient use of resources requires developers to think in terms of the cloud from the start. To optimize cloud expenditures, a cloud-native application design requires an in-depth familiarity with the services and resources offered by different cloud service providers. Managed service options are superior to autonomous technologies since they require less effort and time investment [329]. A sophisticated knowledge of the user application's demands, regulatory demands, and possible financial consequences is necessary to choose between a single and multi-cloud installation plan. An organization's administration might be simplified by adopting a single-cloud approach, but doing so could leave it vulnerable to vendor lock-in and service restrictions [2]. Contrarily, a multi-cloud strategy can increase complexity in administration but has the ability to optimize costs, provide greater flexibility, and lessen the danger of vendor lock-in. Identifying which is the most economical and profitable implementation approach requires careful consideration of the specific features, pricing methods, and competencies of different cloud services.

#### 5.5.2. Resource allocation

The sheer size of today's CDCs makes resource management in networked systems a formidable challenge. In large-scale distributed architectures, the variety of network devices, elements, and ways to connect raises the difficulty of resource management strategies [330]. Consequently, there is a necessity for innovative resource allocation methodologies that would add to the reliability and effectiveness of these systems while keeping them cost-effective and sustainable. While resource management is fundamental to distributed systems (be it the cloud, the IoT, or fog computing), additional guarantees are needed to ensure that these systems operate well in terms of latency, dependability, cost-effectiveness, and throughput [331]. The software layer is just one part of these larger systems, which also require consideration of networking, server architecture, and ventilation. By incorporating blockchain technology into operations like resource sharing and VM migration, cloud systems may be more secure [332]. There is a pressing need to investigate novel approaches to managing computer system resources by taking a systemic perspective and using AI models. Moreover, experiment-driven strategies for examining methods to optimize resource management methods may be investigated [333]. Borg was opened up by Google as Kubernetes, which is an instance of a cluster management system that incorporates data abstraction into resource management. Users are freed from worrying about the nuts and bolts of resource management and may instead focus on composing cloud-native applications.

Borg conceptually separates the whole cluster into cells, each housing a Borgmaster (controller) and a Borglet (which initiates and terminates tasks within the cell's perimeter). The master node coordinates with the Borglets and processes RPCs from clients requesting actions like creating jobs or reading data [253]. This centralized design is very suitable for scaling. The primary benefit of this architecture is that operations that have already been started will continue to execute even if the master or a Borglet fails [334].

A system known as Mesos can facilitate the equitable distribution of commodity clusters. It coordinates the use of commodity clusters by many systems. The fundamental idea is to make use of available resources [335]. In this model, Mesos determines how many resources to give to every framework depending on the limitations associated with that framework, and the frameworks then choose which offers to take. Thus, scheduling choices must be made by frameworks. In addition, Mesos facilitates the creation of domain-specific frameworks (like Spark) that may greatly enhance performance. To schedule and manage available resources, YARN is used as a framework [1]. It enables services to ask for computing power at various topological levels, including individual servers, networks, and whole racks. The primary component in charge of allocation is YARN's resource management. Similarly to Mesos, it enables several frameworks to collaborate on the same commodity clusters [334]. YARN's integrated reliability masks the complexities of failure identification and recovery.

- **Heterogeneous Resources and Workloads:** There is a lack of cohesion in the existing literature about managing resources and workloads in diverse cloud settings. As a result, there is no common setting in which cloud applications can make optimal use of heterogeneity in VMs, vendors, and hardware architectures [151]. Consequently, the initiative recommends an overarching program that takes into consideration diversity throughout. Effective solutions can be picked from a collection of workload and resource handling methods, depending on an application's needs [336]. Heterogeneous memory control is necessary for this purpose. Modern memory control techniques rely heavily on hypervisors, thereby minimizing the potential advantages of heterogeneity. Recent calls for action have advocated for alternatives that focus on heterogeneity awareness in the guest OS. Another chasm is that between heterogeneity and abstraction [337]. Accelerator-specific languages and low-level programming initiatives are necessary for today's programming paradigms to utilize hardware processors. Furthermore, such models allow for the creation of useful research software. As a result, service-oriented and user-driven applications on cloud platforms are hampered in their ability to take advantage of heterogeneity. Kick-starting an international community initiative to come up with an open-source, high-level programming language that is suitable for cutting-edge and creative Web-based applications in a heterogeneous setting is a worthwhile step to take [338]. Whenever fog computing matures and application migration occurs, such aids will be invaluable.

*5.5.3. Application design*

By 2025, analysts predict 61 billion connected devices will generate 40 percent of global data at the cost of $2.5 trillion [339]. Medical services, near-real-time traffic management systems, precise farming, intelligent towns and cities, etc., are just a few examples of IoT applications that are driving the need for improved processing capacity, data storage, confidentiality, security, and trustworthy communication. Additionally, as the data produced by these devices is used to resolve real-time challenges, credibility, uniformity, and accessibility of the data must be maintained. It is challenging to design such complex applications for IoT systems [340]. As a result, it is essential to develop application designs and architectures that are not only dependable and quick enough to deliver effective efficiency but additionally, scalable to manage massive amounts of data through these devices. These are the most important factors to consider when developing such apps for cloud environments. Firstly, a data packet's latency is the time it takes to travel between an IoT device and the cloud before returning. For time-sensitive information, even a millisecond delay might have drastic consequences. For instance, having a crisis-sensing instrument that only sounds an alert after a disaster has already taken place is not a viable solution. Data needing immediate reaction should be analyzed as close as possible to the origin [341]. Secondly, if all this data is transferred to the cloud for storage and analysis, the resulting traffic will be massive, using up all available bandwidth. The distance between the device and the cloud also increases transmission latency, which slows down responses and reduces user experience. Therefore, some tasks must be transferred from the cloud to an edge server located between the Internet servers and the mobile device: such solutions better satisfy end-users' requirements.

By storing and processing certain IoT data directly on IoT devices, the fog computing model reduces the load on the cloud and keeps costs down. Large-scale, geographically dispersed applications that rely heavily on real-time data benefit from the fog's consistency [342]. Fog computing may be the most appropriate choice to enable effective IoT and provide reliable and safe services and resources to many IoT users. Big data analytics, IoT devices, fog, and edge computing have become the foundations for smart city programs worldwide [343]. In transport, fog computing is useful for several tasks, including vehicle-to-vehicle interaction, smart-sensor-based congestion control system management, driverless car management, and self-parking, among others. Furthermore, governments may employ these applications to make the lives of their residents safer and more environmentally friendly, making them a sustainable approach. Emergency services, such as those dealing with fires or natural disasters, can also benefit from this technology by receiving timely alerts about developing crises to help them make informed choices.

Farming software that tracks weather and climatic data like rainfall, wind speed, and temperatures, makes it easier for farmers to reap a harvest. An IoT agriculture platform is suggested for cloud and fog computing, with applications including automated agricultural monitoring, visual inspection for pest control, and more efficient use of farm resources [340]. Meanwhile, in the medical field, more and more people are using fitness trackers, blood pressure monitors, and heart rate monitors to track vital signs and gather data for medical analysis. Thanks to these innovations, physicians can check their patients' health from afar, and patients have more say in their care and decisions.

*5.5.4. Computing economics*

There are several promising new avenues for study in the financial aspects of cloud computing. It is becoming clearer that the lower costs of container deployment can be used to handle real-time workloads [344]. This is speeding up the switch from VMs to containers for cloud computing.

- **Cost-Effective Computing Models:** In serverless computing, no billing for computing resources is made until a function is invoked. Processes executed in these lambda functions tend to be narrower in focus and designed for processing data streams. Whether or not serverless computing is beneficial for a given application depends on its projected runtime behavior and workload [1]. Averaged versus peak transaction rates; scaling the number of simultaneous operations on the infrastructure (i.e., operating multiplies simultaneous functions with a growing number of consumers); and benchmark implementation of serverless functions across various backend hardware platforms [345]. Conversely, increased employment of fog and edge computing characteristics with cloud-based data centers gives tremendous study potential in cloud economics.
- **Economic Impact of Computing Technologies:** It is possible to lower the expenses of running cloud services and infrastructure by combining reliable resources of the cloud with more ephemeral resources at the consumer's edge. To make such technology accessible at the edge, nevertheless, it is anticipated that consumers will require some sort of inducement [157]. Expanding the cloud market to include new types of service providers is possible because of the accessibility of cloud and edge resources. Researchers call these intermediate facilities located between the conventional data center and the user-owned or provisioned resources, microdata centers [346]. The federation concept in computing allows for many microdata center operators to operate together to distribute workloads in a given region at desired pricing.

*5.5.5. Data management*

Metadata handling for datasets is not given much attention in cloud IaaS and PaaS services for storing and information administration, which instead prioritize file, partially structured, and structured data separately. In contrast to traditional, organized data warehouses, proponents of "Data Lakes" advocate for businesses to store all their data in unstructured formats on the cloud, using services like Hadoop [1]. Nevertheless, using them might be difficult due to the absence of information for tracking and defining the origin and authenticity of the data.

Throughout the past ten years, research archives have become exceptional in handling vast, varied datasets and the accompanying information that provides context for their usage. Collocating data and computing resources in a small number of strategically located data centers worldwide allow for economies of scale, a major advantage of CDCs [348]. Nevertheless, bandwidth restrictions across worldwide networks and delays in gaining access to data present obstacles [350]. This becomes an increasingly pressing issue as IoT and 5G mobile networks expand. However, the cloud providers' access to private data and critical confidential information still poses a risk for businesses that need to guarantee strict privacy for their end-users. Likewise, there are no foolproof auditing techniques to prove that the cloud service provider has not obtained the data, even though regulatory measures are in place. In a hybrid setup, customers may handle confidential information under their watchful eye while still taking advantage of the advantages of public clouds, thanks to the proximity of private data centers to public CDCs connected by an independent high-bandwidth network. Furthermore, effective approaches to managing resource flexibility in such contexts should be explored [351]. In addition, it is preferable to have high-level programming abstractions and bindings to platforms that can allocate and oversee resources in these massively dispersed settings.

Finally, with the IoT, deep learning, and blockchain all set to be housed on clouds, it is important to look at specialist data management services to ensure their success [352]. As indicated above, IoT will include a strengthened requirement to deal with streaming data, their effective storage, and a requirement to integrate data management on the edge effortlessly with administration in the cloud [38]. When unregulated edge devices are involved, integrity and authenticity become even more crucial. As the use of deep learning grows, it will become more important to be able to manage trained models well and

**Table 5**
Summary of open challenges and future directions in the above-discussed impact of modern computing and performance criteria with future reading.

| Impact and performance criteria | Open challenges and future directions | Further reading |
|---|---|---|
| QoS and SLA | How can SLAs and QoS be preserved in real-time when cloud computing and edge resources and tasks are executed? | ACM CSUR [287] and Wiley IJCS [289] |
| Autoscaling | How can it be ensured that computing resources need to meet SLAs and QoS are effectively autoscaled in real-time? | ACM CSUR [295] |
| Fault Tolerance | How can reliable support be continuously provided with environmentally-friendly services? | Elsevier SETA [300] |
| Energy Consumption | How can modern computing benefit from AI/ML to provide environmentally-friendly services and low energy consumption? | Springer Cluster Computing [304] |
| Carbon Footprint | What technological advancements may decrease the impact of climate change and how could environmentally-friendly computing have a lower-carbon footprint? | IEEE COMST [190] |
| Serviceability | What methodologies should be employed to develop and measure key performance indicators, also known as KPIs, in order to assess the success of initiatives that aim to make cloud computing more usable and secure? | Wiley ETT [309] |
| Digital Divide | How does the use of the cloud help overcome the digital divide? Can ICTs help bridge the digital divide in infrastructural growth? | Elsevier Telematics and Informatics [313] |
| Ethical AI | When designing and implementing AI in computing devices, what ethical concerns must be taken into account? | Nature Machine Intelligence [347] |
| Digital Humanism | How may digital tools stimulate original thought and the independent thinking of individuals, and whether or not the synergy of these traits can promote a digital shift in the workplace? | Elsevier Journal of Business Research [319] |
| Security, Privacy & Resiliency | What measures can be taken to ensure that personal information is protected and data is securely processed in the cloud when IoT apps collect and analyze massive amounts of data? | IEEE COMST [323] |
| Cost-Efficiency | How can impending difficulties like the prohibitive cost of setting up and running big systems testing environments and the influence of global warming on the architecture of upcoming systems be overcome? | Springer Cluster Computing [304] |
| Resource Allocation | What are the best practices for successfully provisioning cloud and edge resources for many IoT apps before scheduling such resources? | ACM CSUR [333] |
| Heterogeneous Workloads/ Resources | How can the heterogeneity of resources and workloads impact the efficiency of a computing system at runtime? | ACM CSUR [151] |
| Application Design | How can more efficient IoT apps be developed to make greater use of available computer power? | ACM CSUR [201] |
| Computing Economics | How can businesses strengthen their CapEx (Capital Expenditure) and OpEx (Operational Expenditure) strategies by learning about the primary economic advantages of cloud computing in terms of return on investment (ROI), total cost of ownership (TCO), and relocation? | Elsevier Telecommunications Policy [344] |
| Data Management | How can organizations make optimal use of AI/ML approaches for enormous amounts of data to ensure efficient data administration and analysis? | Springer JBD [348] & ACM CSUR [349] |

make sure they can be quickly loaded and switched between to make online and distributed analytics applications possible [349]. Finally, blockchain and decentralized ledgers can improve data management and tracking by providing greater transparency and auditability. While initially used by the financial sector (of which cryptocurrencies are only one prominent example), these systems may be expanded to store other company data safely with an inherent auditing record.

**Summary:** Table 5 lists the summary of open challenges and future directions in the above-discussed impact of modern computing and performance criteria, along with recommendations for future reading.

## 6. Emerging trends in modern computing

The advent of modern computing technology has made it possible to resolve several real-world issues, including delayed responses and low latency. It has facilitated the development of start-ups led by promising young minds from all over the world, providing access to massive computing capacity for tackling difficult issues and accelerating scientific advancement. Thanks to its ground-breaking improvements in efficiency in domains like neural networks, Natural Language Processing (NLP), and related applications, AI has been gaining popularity lately. Computing is a vital infrastructure for running AI services due to its enormous processing power, and AI has the potential to improve existing computing by making resource management effective. Several AI models rely on outside data sets and large-scale computer capacity, both of which might be easier to access with today's computing systems. Currently, training advanced models of AI in large numbers is becoming even more crucial. Additionally, extensive application of AI in contemporary computer systems may be possible due to ground-breaking XAI research. In the decades to come, AI will place substantial stress on computing resources. To meet these demands, it is necessary to develop new approaches to research and methodology that make use of AI models to solve problems with adaptability, delay, and handling of resources and cybersecurity. Scalability and adaptability are two open issues that have not yet made full use of AI models as an economical way to boost the performance of computer applications.

Our analysis has led us to categorize certain areas of computing into three separate maturity levels: a period of five to ten years, over a decade, and under five years. Several novel innovations are on the horizon that might significantly improve the utilization of modern computing, and the article has highlighted them all over the coming decade. Fig. 2 depicts the hype cycle for modern computing systems along with their new trends. Researchers extensively study computing paradigms and technologies, with edge AI and federated learning now dominating. New areas of study within computing, such as distributed computing continuum and AI-driven computing are just scratching the surface. Applications for computing in these domains may not mature for another five to 10 years. Quantum ML, sustainability, Net Zero

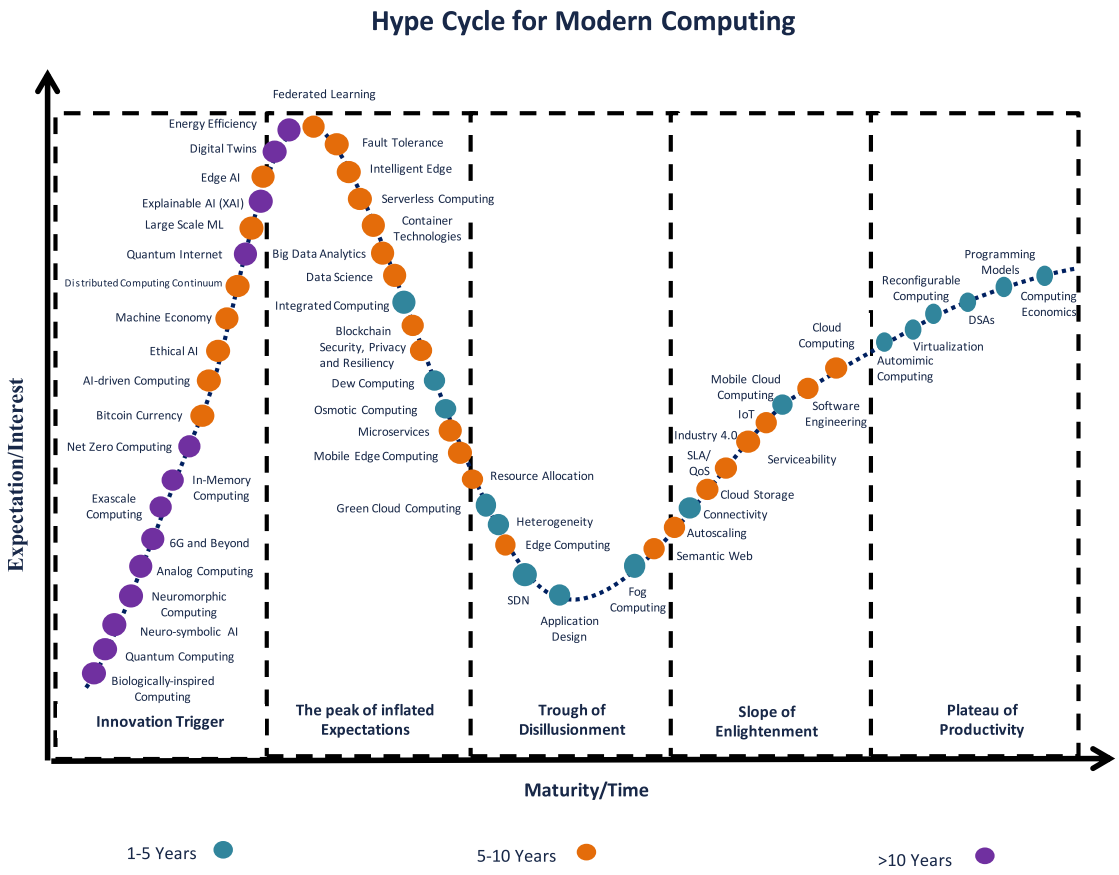


**Fig. 2.** Hype cycle for modern computing.

Computing, XAI, and the quantum Internet are all expected to be in the spotlight for at least another decade. Digital twins, cybersecurity, edge intelligence, edge computing, and blockchain technology have generated an unprecedented level of excitement. They are expected to be completely built-in under five years with the help of modern technology. Machine Economics, In-Memory Computing, Bitcoin Currency and AIOps/MLOps have all reached their peak of inflated expectations for the following five to 10 years of noteworthy evolution. Significant progress needs to be made before biologically inspired computing, neuro-symbolic AI, analog computing, neuromorphic computing, 6G, and quantum computing can be considered hype-worthy. Cloud and fog computing has been trending heavily over the past few years, and that trend could persist for the next five to ten years.

## 7. Summary and conclusions

This research offers a comprehensive exploration of the evolution of modern computing systems over the past sixty years, tracking the transition from classical computers to quantum computing and examining their key components, such as physical architecture, conceptual units, and communication methods. We analyze the influence of conceptualization and physical models on the shift from centralized to decentralized structures, a significant change since the Internet's inception. Developments in microcontroller architecture, operating system design, and networking infrastructure have given rise to ubiquitous computing models like the Internet of Things (IoT), pushing the boundaries of both physical and conceptual realms. The move towards specialized hardware and software, particularly in data-driven fields like AI, represents a shift from earlier focuses on system flexibility and adaptability. This article also addresses issues of accessibility and potential inequalities, emphasizing the need to ensure these technologies positively impact society and everyday life. Integrating recent advancements with ongoing challenges in the application of established technological trends, this work provides an in-depth analysis of the next wave of scientific research in computing. It summarizes current findings, acknowledges limitations, and outlines new trends and key challenges, considering the impact of emerging trends and envisioning future research paths in modern computing. This review aims to be a valuable resource for experts, technologists, and academics interested in the latest developments and future directions in the field of modern computing.

## CRediT authorship contribution statement

**Sukhpal Singh Gill:** Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization, Visualization, Writing – review & editing. **Huaming Wu:** Writing – review & editing, Writing – original draft, Conceptualization, Data curation, Formal analysis, Investigation, Methodology. **Panos Patros:** Writing – review & editing, Writing – original draft, Data curation, Conceptualization, Investigation, Methodology. **Carlo Ottaviani:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization, Investigation. **Priyansh Arora:** Writing – original draft, Formal analysis, Conceptualization, Investigation, Methodology, Writing – review & editing. **Victor Casamayor Pujol:** Writing – original draft, Data curation, Conceptualization, Investigation, Methodology, Writing – review & editing. **David Haunschild:** Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Ajith Kumar Parlikad:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Oktay Cetinkaya:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Hanan Lutfiyya:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Vlado Stankovski:** Writing –

**Table 6**
List of acronyms.

| Abbreviation | Description |
|---|---|
| PCs | Personal Computers |
| DNS | Domain Name System |
| MPP | Massive Parallel Processing |
| AI | Artificial Intelligence |
| SMP | Symmetric Multi Processing |
| OS | Operating System |
| GUI | Graphical User Interfaces |
| IoT | Internet of Things |
| HTTP | Hyper Text Transport Protocol |
| HTML | Hyper Text Markup Language |
| WWW | World Wide Web |
| RPC | Remote Procedure Calls |
| JSON | JavaScript Object Notation |
| XML | Extensible Markup Language |
| SOA | Service-Oriented Architecture |
| CDC | Cloud Data Centers |
| HPC | High Performance Computing |
| IT | Information Technology |
| SaaS | Software as a Service |
| PaaS | Platform as a Service |
| IaaS | Infrastructure as a Service |
| SBC | Single-board Computers |
| SDN | Software-Defined Networking |
| NVF | Network Function Virtualization |
| IIoT | Industrial Internet of Things |
| QoS | Quality of Service |
| IoE | Internet of Energy |
| B5G | Beyond 5G |
| SLA | Service-Level Agreement |
| FPGA | Field-Programmable Gate Arrays |
| ASICs | Application-Specific Integrated Circuits |
| GPU | Graphics Processing Units |
| CUDA | Compute Unified Device Architecture |
| TPU | Tensor Processing Units |
| ICT | Information and Communication Technology |
| CaaS | Container as a Service |
| QoE | Quality of Experience |
| V2X | Vehicle-to-Everything |
| MEC | Multi-access Edge Computing |
| VM | Virtual Machines |
| M2M | Machine-to-Machine |
| PoW | Proof of Work |
| XAI | Explainable Artificial Intelligence |
| UAV | Unmanned Aerial Vehicle |
| DDoS | Distributed Denial of Service |
| STCO | Systems-Technology Co-Optimization |
| SoC | System-on-a-Chip |
| ML | Machine Learning |
| SLO | Service Level Objective |

review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Ruidong Li:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Yuemin Ding:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Junaid Qadir:** Writing – review & editing, Writing – original draft, Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Visualization. **Ajith Abraham:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Soumya K. Ghosh:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Houbing Herbert Song:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization, Investigation. **Rizos Sakellariou:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization, Investigation, Methodology, Supervision. **Omer Rana:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology, Supervision. **Joel J.P.C. Rodrigues:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Salil S. Kanhere:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology. **Schahram Dustdar:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology, Supervision. **Steve Uhlig:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology, Supervision. **Kotagiri Ramamohanarao:** Writing – review & editing, Writing – original draft, Conceptualization, Investigation, Methodology, Supervision. **Rajkumar Buyya:** Writing – review & editing, Writing – original draft, Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Visualization.

## Declaration of competing interest

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

## Data availability

No data was used for the research described in the article.

## Acknowledgments

We thank the Editor-in-Chief (Prof. Ke Xue) and anonymous reviewers for their insightful comments and recommendations to improve the overall quality and organization of the article. We would also like to express our gratitude to Neil Butler (CEO, CloudScaler, UK), Marco AS Netto (Microsoft Azure HPC, USA) and Manmeet Singh (University of Texas at Austin, USA) for their thoughtful remarks and valuable suggestions.

## Appendix. List of acronyms

Table 6 shows the list of acronyms.

## References

[1] R. Buyya, et al., A manifesto for future generation cloud computing: Research directions for the next decade, ACM Comput. Surv. 51 (5) (2018) 1–38.
[2] D. Lindsay, et al., The evolution of distributed computing systems: from fundamental to new frontiers, Computing 103 (8) (2021) 1859–1878.
[3] R. Yamashita, History of personal computers in Japan, Int. J. Parallel Emergent Distrib. Syst. 35 (2) (2020) 143–169.
[4] S.S. Gill, et al., AI for next generation computing: Emerging trends and future directions, Int. Things 19 (2022) 100514.
[5] J. Gubbi, et al., Internet of Things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.
[6] R. Muralidhar, et al., Energy efficient computing systems: Architectures, abstractions and modeling to techniques and standards, ACM Comput. Surv. 54 (11s) (2022) 1–37.
[7] A. Chakraborty, et al., Journey from cloud of things to fog of things: Survey, new trends, and research directions, Softw. - Pract. Exp. 53 (2) (2023) 496–551.
[8] A. Beloglazov, et al., Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Gener. Comput. Syst. 28 (5) (2012) 755–768.
[9] V. Casamayor Pujol, et al., Fundamental research challenges for distributed computing continuum systems, Information 14 (3) (2023) 198.
[10] J. Shalf, The future of computing beyond Moore's law, Phil. Trans. R. Soc. A 378 (2166) (2020) 20190061.
[11] N.A. Angel, et al., Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies, Sensors 22 (1) (2021) 196.
[12] B.P. Rimal, et al., A taxonomy and survey of cloud computing systems, in: 2009 Fifth International Joint Conference on INC, IMS and IDC, IEEE, 2009, pp. 44–51.
[13] S.S. Gill, et al., Transformative effects of IoT, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges, Int. Things 8 (2019) 100118.
[14] M.J. Flynn, Very high-speed computing systems, Proc. IEEE 54 (12) (1966) 1901–1909.
[15] C.E. Kozyrakis, et al., A new direction for computer architecture research, Computer 31 (11) (1998) 24–32.
[16] T.L. Casavant, et al., A taxonomy of scheduling in general-purpose distributed computing systems, IEEE Trans. Softw. Eng. 14 (2) (1988) 141–154.

[17] J. Yu, et al., A taxonomy of workflow management systems for grid computing, J. Grid Comput. 3 (2005) 171–200.
[18] J.D. Owens, et al., GPU computing, Proc. IEEE 96 (5) (2008) 879–899.
[19] K. Compton, et al., Reconfigurable computing: a survey of systems and software, ACM Comput. Surv. (csuR) 34 (2) (2002) 171–210.
[20] S. Wright, Cybersquatting at the intersection of internet domain names and trademark law, IEEE Commun. Surv. Tutor. 14 (1) (2010) 193–205.
[21] B.J. Jansen, The graphical user interface, ACM SIGCHI Bull. 30 (2) (1998) 22–26.
[22] B.H. Tay, et al., A survey of remote procedure calls, Oper. Syst. Rev. 24 (3) (1990) 68–79.
[23] R.R. Suryono, et al., Peer to peer (P2P) lending problems and potential solutions: A systematic literature review, Procedia Comput. Sci. 161 (2019) 204–214.
[24] R. Schollmeier, et al., Protocol for peer-to-peer networking in mobile environments, in: Proceedings. 12th International Conference on Computer Communications and Networks (IEEE Cat. No. 03EX712), IEEE, 2003, pp. 121–127.
[25] G. Alonso, et al., Web Services, Springer, 2004.
[26] R. Perrey, et al., Service-oriented architecture, in: 2003 Symposium on Applications and the Internet Workshops, 2003. Proceedings, IEEE, 2003, pp. 116–119.
[27] V. Maffione, et al., A software development kit to exploit RINA programmability, in: 2016 IEEE International Conference on Communications (ICC), IEEE, 2016, pp. 1–7.
[28] L. Resende, Handling heterogeneous data sources in a SOA environment with service data objects (SDO), in: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, 2007, pp. 895–897.
[29] M.F. Mergen, et al., Virtualization for high-performance computing, Oper. Syst. Rev. 40 (2) (2006) 8–11.
[30] J.O. Kephart, et al., The vision of autonomic computing, Computer 36 (1) (2003) 41–50.
[31] S. Singh, et al., STAR: SLA-aware autonomic management of cloud resources, IEEE Trans. Cloud Comput. 8 (4) (2017) 1040–1053.
[32] M. Othman, et al., A survey of mobile cloud computing application models, IEEE Commun. Surv. Tutorials 16 (1) (2013) 393–413.
[33] A.S. AlAhmad, et al., Mobile cloud computing models security issues: A systematic review, J. Netw. Comput. Appl. 190 (2021) 103152.
[34] M.H. Anwar, et al., Recommender system for optimal distributed deep learning in cloud datacenters, Wirel. Pers. Commun. (2022) 1–25.
[35] F. Durao, et al., A systematic review on cloud computing, J. Supercomput. 68 (2014) 1321–1346.
[36] S.S. Gill, et al., ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices, J. Syst. Softw. 154 (2019) 125–138.
[37] S. Iftikhar, et al., AI-based fog and edge computing: A systematic review, taxonomy and future directions, Int. Things (2022) 100674.
[38] S.S. Gill, et al., Fog-based smart healthcare as a big data and cloud service for heart patients using IoT, in: International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018, Springer, 2019, pp. 1376–1383.
[39] J. Singh, et al., Fog computing: A taxonomy, systematic review, current trends and research challenges, J. Parallel Distrib. Comput. 157 (2021) 56–85.
[40] W. Shi, et al., Edge computing: Vision and challenges, IEEE Int. Things J. 3 (5) (2016) 637–646.
[41] G.K. Walia, et al., AI-empowered fog/edge resource management for IoT applications: A comprehensive review, research challenges and future perspectives, IEEE Commun. Surv. Tutor. 26 (1) (2023) 1–56.
[42] W.Z. Khan, et al., Edge computing: A survey, Future Gener. Comput. Syst. 97 (2019) 219–235.
[43] E. Jonas, et al., Cloud programming simplified: A berkeley view on serverless computing, 2019, arXiv preprint arXiv:1902.03383.
[44] H.B. Hassan, et al., Survey on serverless computing, J. Cloud Comput. 10 (1) (2021) 1–29.
[45] A. Buzachis, et al., Modeling and emulation of an osmotic computing ecosystem using osmotictoolkit, in: Proceedings of the 2021 Australasian Computer Science Week Multiconference, 2021, pp. 1–9.
[46] B. Neha, et al., A systematic review on osmotic computing, ACM Trans. Int. Things 3 (2) (2022) 1–30.
[47] P.P. Ray, An introduction to dew computing: definition, concept and implications, IEEE Access 6 (2017) 723–737.
[48] M. Gushev, Dew computing architecture for cyber-physical systems and IoT, Int. Things 11 (2020) 100186.
[49] Y. Qu, et al., A blockchained federated learning framework for cognitive computing in industry 4.0 networks, IEEE Trans. Ind. Inform. 17 (4) (2020) 2964–2973.
[50] T. Kovachy, et al., Quantum superposition at the half-metre scale, Nature 528 (7583) (2015) 530–533.
[51] S.S. Gill, et al., Quantum computing: A taxonomy, systematic review and future directions, Softw. - Pract. Exp. 52 (1) (2022) 66–114.
[52] S.R. Gulliver, et al., Pervasive and standalone computing: the perceptual effects of variable multimedia quality, Int. J. Hum.-Comput. Stud. 60 (5–6) (2004) 640–665.
[53] S. Ravi, et al., Security in embedded systems: Design challenges, ACM Trans. Embed. Comput. Syst. (TECS) 3 (3) (2004) 461–491.
[54] L. De Micco, et al., A literature review on embedded systems, IEEE Latin Am. Trans. 18 (02) (2019) 188–205.
[55] P.J. Basford, et al., Performance analysis of single board computer clusters, Future Gener. Comput. Syst. 102 (2020) 278–291.
[56] A. Pajankar, Raspberry pi supercomputing and scientific programming, Ashwin Pajankar (2017).
[57] T. Hwu, et al., A self-driving robot using deep convolutional neural networks on neuromorphic hardware, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 635–641.
[58] A.A. Süzen, et al., Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn, in: 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE, 2020, pp. 1–5.
[59] A. Kumar, et al., Securing the future internet of things with post-quantum cryptography, Secur. Priv. 5 (2) (2022) e200.
[60] J. Ren, et al., A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet, ACM Comput. Surv. 52 (6) (2019) 1–36.
[61] C. Wang, et al., Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges, IEEE Commun. Surv. Tutor. 20 (1) (2017) 7–38.
[62] J.Z. Ahmadabadi, et al., Star-quake: A new operator in multi-objective gravitational search algorithm for task scheduling in IoT based cloud-fog computing system, IEEE Trans. Consum. Electron. (2023).
[63] A. Asghari, et al., Server placement in mobile cloud computing: a comprehensive survey for edge computing, fog computing and cloudlet, Computer Science Review 51 (2024) 100616.
[64] M.F. Bari, et al., On orchestrating virtual network functions, in: 2015 11th International Conference on Network and Service Management (CNSM), IEEE, 2015, pp. 50–56.
[65] Y. Cai, et al., Compute-and data-intensive networks: The key to the metaverse, in: 2022 1st International Conference on 6G Networking (6GNet), IEEE, 2022, pp. 1–8.
[66] E. Al-Masri, et al., Energy-efficient cooperative resource allocation and task scheduling for Internet of Things environments, Int. Things 23 (2023) 100832.
[67] M. Sriraghavendra, et al., DoSP: A deadline-aware dynamic service placement algorithm for workflow-oriented IoT applications in fog-cloud computing environments, in: Energy Conservation Solutions for Fog-Edge Computing Paradigms, Springer, 2022, pp. 21–47.
[68] P. Verma, et al., FCMCPS-COVID: AI propelled fog–cloud inspired scalable medical cyber-physical system, specific to coronavirus disease, Int. Things 23 (2023) 100828.
[69] F. Desai, et al., HealthCloud: A system for monitoring health status of heart patients using machine learning and cloud computing, Int. Things 17 (2022) 100485.
[70] S. Iftikhar, et al., FogDLearner: A deep learning-based cardiac health diagnosis framework using fog computing, in: Proceedings of the 2022 Australasian Computer Science Week, ACM, 2022, pp. 136–144.
[71] S.S. Gill, et al., IoT based agriculture as a cloud and big data service: the beginning of digital India, J. Organ. End User Comput. (JOEUC) 29 (4) (2017) 1–23.
[72] A. Sengupta, et al., Mobile edge computing based internet of agricultural things: a systematic review and future directions, Mob. Edge Comput. (2021) 415–441.
[73] S. Iftikhar, et al., Fog computing based router-distributor application for sustainable smart home, in: 2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring), IEEE, 2022, pp. 1–5.
[74] K. Bansal, et al., DeepBus: Machine learning based real time pothole detection system for smart transportation using IoT, Int. Technol. Lett. 3 (3) (2020) e156.
[75] S. Tuli, et al., IThermoFog: IoT-fog based automatic thermal profile creation for cloud data centers using artificial intelligence techniques, Int. Technol. Lett. 3 (5) (2020) e198.
[76] M. Singh, et al., Quantum artificial intelligence for the science of climate change, in: Artificial Intelligence, Machine Learning and Blockchain in Quantum Satellite, Drone and Network, CRC Press, 2022, pp. 199–207.
[77] M. Singh, et al., Quantifying COVID-19 enforced global changes in atmospheric pollutants using cloud computing based remote sensing, Remote Sens. Appl.: Soc. Environ. 22 (2021) 100489.
[78] M. Stoyanova, et al., A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1191–1221.
[79] N. Mansouri, et al., Cloud computing simulators: A comprehensive review, Simul. Model. Pract. Theory 104 (2020) 102144.
[80] S. Tuli, et al., HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments, Future Gener. Comput. Syst. 104 (2020) 187–200.

[81] S.S. Gill, et al., ChatGPT: Vision and challenges, Int. Things Cyb.-Phys. Syst. 3 (2023) 262–271.
[82] M. Vila, et al., Edge-to-cloud sensing and actuation semantics in the industrial Internet of Things, Pervasive Mob. Comput. 87 (2022) 101699.
[83] D. Kreutz, et al., Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2014) 14–76.
[84] T. Mekki, et al., Software-defined networking in vehicular networks: A survey, Trans. Emerg. Telecommun. Technol. 33 (10) (2022) e4265.
[85] J. Son, et al., A taxonomy of software-defined networking (SDN)-enabled cloud computing, ACM Comput. Surv. (csuR) 51 (3) (2018) 1–36.
[86] L. Poutievski, et al., Jupiter evolving: transforming google's datacenter network via optical circuit switches and software-defined networking, in: Proceedings of the ACM SIGCOMM 2022 Conference, 2022, pp. 66–85.
[87] A. Kumar, et al., A secure drone-to-drone communication and software defined drone network-enabled traffic monitoring system, Simul. Model. Pract. Theory 120 (2022) 102621.
[88] X. Wang, et al., Convergence of edge computing and deep learning: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (2) (2020) 869–904.
[89] J. Zhang, et al., Mobile edge intelligence and computing for the internet of vehicles, Proc. IEEE 108 (2) (2019) 246–261.
[90] S. Chen, et al., Internet of things based smart grids supported by intelligent edge computing, IEEE Access 7 (2019) 74089–74102.
[91] V.C. Pujol, et al., Edge intelligence—Research opportunities for distributed computing continuum systems, IEEE Internet Comput. 27 (4) (2023) 53–74.
[92] R. Singh, et al., Edge AI: a survey, Int. Things Cyb.-Phys. Syst. 3 (2023) 71–92.
[93] Y. Jia, et al., Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks, IEEE Internet Things J. 7 (10) (2020) 9552–9562.
[94] B. Yang, et al., Edge intelligence for autonomous driving in 6G wireless system: Design challenges and solutions, IEEE Wirel. Commun. 28 (2) (2021) 40–47.
[95] F. Liu, et al., Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond, IEEE J. Selected Areas Commun. 40 (6) (2022) 1728–1767.
[96] M. Ishtiaq, et al., Edge computing in IoT: A 6G perspective, 2021, arXiv preprint arXiv:2111.08943.
[97] A. Kumar, et al., A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic, Future Gener. Comput. Syst. 115 (2021) 1–19.
[98] Y. Shi, et al., Machine learning for large-scale optimization in 6g wireless networks, IEEE Commun. Surv. Tutor. (2023).
[99] A. Alkhateeb, et al., Real-time digital twins: Vision and research directions for 6G and beyond, IEEE Commun. Mag. (2023).
[100] S.A. Ansar, et al., Intelligent Fog-IoT Networks with 6G endorsement: Foundations, applications, trends and challenges, in: 6G Enabled Fog Computing in IoT: Applications and Opportunities, Springer, 2023, pp. 287–307.
[101] I.F. Akyildiz, et al., 6G and beyond: The future of wireless communications systems, IEEE Access 8 (2020) 133995–134030.
[102] S. Ghafouri, et al., Mobile-kube: Mobility-aware and energy-efficient service orchestration on kubernetes edge servers, in: 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), IEEE, 2022, pp. 82–91.
[103] H. Wu, et al., Energy-efficient decision making for mobile cloud offloading, IEEE Trans. Cloud Comput. 8 (2) (2020) 570–584.
[104] H. Wu, et al., Lyapunov-guided delay-aware energy efficient offloading in iIoT-mec systems, IEEE Trans. Ind. Inform. 19 (2) (2023) 2117–2128.
[105] J.D. Owens, et al., A survey of general-purpose computation on graphics hardware, Comput. Graph. Forum 26 (1) (2007) 80–113.
[106] J. Von Neumann, John Von Neumann: Selected Letters, Vol. 27, American Mathematical Soc., 2005.
[107] D. Kimovski, et al., Beyond von neumann in the computing continuum: Architectures, applications, and future directions, IEEE Internet Comput. (2023).
[108] R. Yang, et al., Integrated blockchain and edge computing systems: A survey, some research issues and challenges, IEEE Commun. Surv. Tutor. 21 (2) (2019) 1508–1532.
[109] S.H. Alsamhi, et al., Computing in the sky: A survey on intelligent ubiquitous computing for uav-assisted 6g networks and industry 4.0/5.0, Drones 6 (7) (2022) 177.
[110] J. Chen, et al., Deep learning with edge computing: A review, Proc. IEEE 107 (8) (2019) 1655–1674.
[111] H. Singh, et al., Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions, Simul. Model. Pract. Theory 111 (2021) 102353.
[112] A. Botta, et al., Integration of cloud computing and internet of things: a survey, Future Gener. Comput. Syst. 56 (2016) 684–700.
[113] F. Cappello, et al., Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with grid, Future Gener. Comput. Syst. 21 (3) (2005) 417–437.
[114] D. Andrews, et al., Achieving programming model abstractions for reconfigurable computing, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 16 (1) (2007) 34–44.
[115] J.C. Jackson, et al., Survey on programming models and environments for cluster, cloud, and grid computing that defends big data, Procedia Comput. Sci. 50 (2015) 517–523.
[116] C. Cao, et al., A novel multi-objective programming model of relief distribution for sustainable disaster supply chain in large-scale natural disasters, J. Clean. Prod. 174 (2018) 1422–1435.
[117] M. Butts, et al., A structural object programming model, architecture, chip and tools for reconfigurable computing, in: 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2007), IEEE, 2007, pp. 55–64.
[118] X. Shen, et al., Holistic network virtualization and pervasive network intelligence for 6G, IEEE Commun. Surv. Tutor. 24 (1) (2021) 1–30.
[119] S. Jin, et al., H-svm: Hardware-assisted secure virtual machines under a vulnerable hypervisor, IEEE Trans. Comput. 64 (10) (2015) 2833–2846.
[120] Y. Mansouri, et al., A review of edge computing: Features and resource virtualization, J. Parallel Distrib. Comput. 150 (2021) 155–183.
[121] J. Zhang, et al., Performance analysis of 3D XPoint SSDs in virtualized and non-virtualized environments, in: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2018, pp. 1–10.
[122] I. Alam, et al., A survey of network virtualization techniques for Internet of Things using SDN and NFV, ACM Comput. Surv. 53 (2) (2020) 1–40.
[123] Y. Xing, et al., Virtualization and cloud computing, in: Future Wireless Networks and Information Systems: Volume 1, Springer, 2012, pp. 305–312.
[124] A. Agache, et al., Firecracker: Lightweight virtualization for serverless applications, in: 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20), 2020, pp. 419–434.
[125] G. Blake, et al., A survey of multicore processors, IEEE Signal Process. Mag. 26 (6) (2009) 26–37.
[126] D. Gizopoulos, et al., Architectures for online error detection and recovery in multicore processors, in: 2011 Design, Automation & Test in Europe, IEEE, 2011, pp. 1–6.
[127] R. Delgado, et al., New insights into the real-time performance of a multicore processor, IEEE Access 8 (2020) 186199–186211.
[128] M. Piattini, et al., Toward a quantum software engineering, IT Prof. 23 (1) (2021) 62–66.
[129] E.-M. Arvanitou, et al., Software engineering practices for scientific software development: A systematic mapping study, J. Syst. Softw. 172 (2021) 110848.
[130] R.R. Althar, et al., The realist approach for evaluation of computational intelligence in software engineering, Innov. Syst. Softw. Eng. 17 (1) (2021) 17–27.
[131] M. De Stefano, et al., Software engineering for quantum programming: How far are we? J. Syst. Softw. 190 (2022) 111326.
[132] G. Sharma, et al., Applications of blockchain in automated heavy vehicles: Yesterday, today, and tomorrow, in: Autonomous and Connected Heavy Vehicle Technology, Elsevier, 2022, pp. 81–93.
[133] J. Al-Jaroodi, et al., Blockchain in industries: A survey, IEEE Access 7 (2019) 36500–36515.
[134] J. Doyle, et al., Blockchainbus: A lightweight framework for secure virtual machine migration in cloud federations using blockchain, Secur. Priv. 5 (2) (2022) e197.
[135] L. Jurado Perez, et al., Simulation of scalability in cloud-based iot reactive systems leveraged on a wsan simulator and cloud computing technologies, Appl. Sci. 11 (4) (2021) 1804.
[136] R. Buyya, et al., A strategy for advancing research and impact in new computing paradigms, in: Green Mobile Cloud Computing, Springer, 2022, pp. 297–308.
[137] C. Brady, et al., All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science, IEEE Trans. Educ. 60 (1) (2016) 59–66.
[138] O. Ferraz, et al., A survey on high-throughput non-binary LDPC decoders: ASIC, FPGA, and GPU architectures, IEEE Commun. Surv. Tutor. 24 (1) (2021) 524–556.
[139] N.P. Jouppi, et al., A domain-specific architecture for deep neural networks, Commun. ACM 61 (9) (2018) 50–59.
[140] J. Cong, et al., Customizable computing—from single chip to datacenters, Proc. IEEE 107 (1) (2018) 185–203.
[141] H. Ji, et al., Magnetic reconnection in the era of exascale computing and multiscale experiments, Nat. Rev. Phys. 4 (4) (2022) 263–282.
[142] S. Heldens, et al., The landscape of exascale research: A data-driven literature analysis, ACM Comput. Surv. 53 (2) (2020) 1–43.
[143] Y. Kim, et al., Evidence for the utility of quantum computing before fault tolerance, Nature 618 (7965) (2023) 500–505.
[144] H. Anzt, et al., Preparing sparse solvers for exascale computing, Phil. Trans. R. Soc. A 378 (2166) (2020) 20190053.
[145] F. Zangeneh-Nejad, et al., Analogue computing with metamaterials, Nat. Rev. Mater. 6 (3) (2021) 207–225.
[146] W. Zhang, et al., Neuro-inspired computing chips, Nat. Electron. 3 (7) (2020) 371–382.
[147] M. Zhao, et al., Reliability of analog resistive switching memory for neuromorphic computing, Appl. Phys. Rev. 7 (1) (2020).
[148] A. Zador, et al., Catalyzing next-generation artificial intelligence through neuroai, Nat. Commun. 14 (1) (2023) 1597.
[149] C.D. Schuman, et al., Opportunities for neuromorphic computing algorithms and applications, Nat. Comput. Sci. 2 (1) (2022) 10–19.

[150] F.C. Morabito, et al., Advances in AI, neural networks, and brain computing: An introduction, in: Artificial Intelligence in the Age of Neural Networks and Brain Computing, Elsevier, 2024, pp. 1–8.
[151] V. Rosenfeld, et al., Query processing on heterogeneous CPU/GPU systems, ACM Comput. Surv. 55 (1) (2022) 1–38.
[152] J. Sanders, et al., CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010.
[153] S. Tuli, et al., Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing, Int. Things 11 (2020) 100222.
[154] L.E. Lwakatare, et al., Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions, Inf. Softw. Technol. 127 (2020) 106368.
[155] M. Wang, et al., A survey on large-scale machine learning, IEEE Trans. Knowl. Data Eng. 34 (6) (2020) 2574–2594.
[156] M.N. Angenent, et al., Large-scale machine learning for business sector prediction, in: Proceedings of the 35th Annual ACM Symposium on Applied Computing, 2020, pp. 1143–1146.
[157] R. Buyya, et al., Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Gener. Comput. Syst. 25 (6) (2009) 599–616.
[158] S.U. Malik, et al., EFFORT: Energy efficient framework for offload communication in mobile cloud computing, Softw. - Pract. Exp. 51 (9) (2021) 1896–1909.
[159] X. Jin, et al., A survey of research on computation offloading in mobile cloud computing, Wirel. Netw. 28 (4) (2022) 1563–1585.
[160] P. Patros, et al., Toward sustainable serverless computing, IEEE Internet Comput. 25 (6) (2021) 42–50.
[161] M. Masdari, et al., Green cloud computing using proactive virtual machine placement: challenges and issues, J. Grid Comput. 18 (4) (2020) 727–759.
[162] S.S. Gill, et al., A taxonomy and future directions for sustainable cloud computing: 360 degree view, ACM Comput. Surv. 51 (5) (2018) 1–33.
[163] W. Shu, et al., Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing, Future Gener. Comput. Syst. 124 (2021) 12–20.
[164] Q. Zhou, et al., Energy efficient algorithms based on VM consolidation for cloud computing: comparisons and evaluations, in: 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), IEEE, 2020, pp. 489–498.
[165] R.F. Mansour, et al., Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment, Cluster Comput. 26 (1) (2023) 575–586.
[166] M. Singh, et al., Dynamic shift from cloud computing to industry 4.0: Eco-friendly choice or climate change threat, in: IoT-Based Intelligent Modelling for Environmental and Ecological Engineering: IoT Next Generation EcoAgro Systems, Springer, 2021, pp. 275–293.
[167] W. Zeng, et al., Research on cloud storage architecture and key technologies, in: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, 2009, pp. 1044–1048.
[168] M. Hota, et al., Leveraging cloud-native microservices architecture for high performance real-time intra-day trading: A tutorial, in: 6G Enabled Fog Computing in IoT: Applications and Opportunities, Springer, 2023, pp. 111–129.
[169] M. Kumar, et al., Qos-aware resource scheduling using whale optimization algorithm for microservice applications, Softw. - Pract. Exp. (2023).
[170] J. Ghofrani, et al., Challenges of microservices architecture: A survey on the state of the practice, ZEUS 2018 (2018) 1–8.
[171] C. Song, et al., ChainsFormer: A chain latency-aware resource provisioning approach for microservices cluster, in: International Conference on Service-Oriented Computing, Springer, 2023, pp. 197–211.
[172] F. Al-Doghman, Mothers, AI-enabled secure microservices in edge computing: Opportunities and challenges, IEEE Trans. Serv. Comput. (2022).
[173] M. Xu, et al., CoScal: Multifaceted scaling of microservices with reinforcement learning, IEEE Trans. Netw. Serv. Manag. 19 (4) (2022) 3995–4009.
[174] O. Bentaleb, et al., Containerization technologies: Taxonomies, applications and challenges, J. Supercomput. 78 (1) (2022) 1144–1181.
[175] A. Barbalace, et al., Edge computing: The case for heterogeneous-ISA container migration, in: Proceedings of the 16th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, 2020, pp. 73–87.
[176] M. Golec, et al., BioSec: A biometric authentication framework for secure and private communication among edge devices in IoT and industry 4.0, IEEE Consum. Electron. Mag. 11 (2) (2020) 51–56.
[177] V. Struhár, et al., Real-time containers: A survey, in: 2nd Workshop on Fog Computing and the IoT (Fog-IoT 2020), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
[178] E. Casalicchio, et al., The state-of-the-art in container technologies: Application, orchestration and security, Concurr. Comput.: Pract. Exper. 32 (17) (2020) e5668.
[179] Z. Zhong, et al., A cost-efficient container orchestration strategy in kubernetes-based cloud computing infrastructures with heterogeneous resources, ACM Trans. Int. Technol. (TOIT) 20 (2) (2020) 1–24.
[180] V. Mallikarjunaradhya, et al., An overview of the strategic advantages of AI-powered threat intelligence in the cloud, J. Sci. Technol. 4 (4) (2023) 1–12.
[181] P. Patros, et al., Investigating resource interference and scaling on multitenant paas clouds, in: Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering, 2016, pp. 166–177.
[182] S. Kounev, et al., Toward a definition for serverless computing, Leibniz-Zentrum fur Informatik (2021).
[183] H. Shafiei, et al., Serverless computing: a survey of opportunities, challenges, and applications, ACM Comput. Surv. 54 (11s) (2022) 1–32.
[184] M. Golec, et al., Qos analysis for serverless computing using machine learning, in: Serverless Computing: Principles and Paradigms, Springer, 2023, pp. 175–192.
[185] M.S. Aslanpour, et al., Serverless edge computing: vision and challenges, in: Proceedings of the 2021 Australasian Computer Science Week Multiconference, 2021, pp. 1–10.
[186] Y. Li, et al., Serverless computing: state-of-the-art, challenges and opportunities, IEEE Trans. Serv. Comput. 16 (2) (2022) 1522–1539.
[187] M. Kumar, et al., AI-based sustainable and intelligent offloading framework for iIoT in collaborative cloud-fog environments, IEEE Trans. Consum. Electron. (2023).
[188] S. Iftikhar, et al., TESCO: Multiple simulations based AI-augmented Fog computing for QoS optimization, in: 2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta), IEEE, 2022, pp. 2092–2099.
[189] F. Firouzi, et al., The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT), Inf. Syst. 107 (2022) 101840.
[190] Z. Cao, et al., Toward a systematic survey for carbon neutral data centers, IEEE Commun. Surv. Tutor. 24 (2) (2022) 895–936.
[191] M.A.B. Siddik, et al., The environmental footprint of data centers in the United States, Environ. Res. Lett. 16 (6) (2021) 064017.
[192] A. Senthilkumar, et al., Enhancement of R600a vapour compression refrigeration system with MWCNT/TiO2 hybrid nano lubricants for net zero emissions building, Sustain. Energy Technol. Assess. 56 (2023) 103055.
[193] T.A. Kurniawan, et al., Decarbonization in waste recycling industry using digitalization to promote net-zero emissions and its implications on sustainability, J. Environ. Manag. 338 (2023) 117765.
[194] R. Wilkinson, et al., Environmental impacts of earth observation data in the constellation and cloud computing era, Sci. Total Environ. 909 (2024) 168584.
[195] A.K. Bhardwaj, et al., HEART: Unrelated parallel machines problem with precedence constraints for task scheduling in cloud computing using heuristic and meta-heuristic algorithms, Softw. - Pract. Exp. 50 (12) (2020) 2231–2251.
[196] G.C. Fox, et al., Parallel Computing Works!, Elsevier, 2014.
[197] H. Wu, et al., A multi-dimensional parametric study of variability in multi-phase flow dynamics during geologic CO2 sequestration accelerated with machine learning, Appl. Energy 287 (2021) 116580.
[198] S.S. Gill, Quantum and blockchain based serverless edge computing: A vision, model, new trends and future directions, Int. Technol. Lett. (2021) e275.
[199] Z.M. Nayeri, et al., Application placement in fog computing with AI approach: Taxonomy and a state of the art survey, J. Netw. Comput. Appl. 185 (2021) 103078.
[200] P. Patros, et al., Rural AI: Serverless-powered federated learning for remote applications, IEEE Internet Comput. 27 (2) (2023) 28–34.
[201] R. Mahmud, et al., Application management in fog computing environments: A taxonomy, review and future directions, ACM Comput. Surv. 53 (4) (2020) 1–43.
[202] A. Ruggeri, et al., An innovative blockchain-based orchestrator for osmotic computing, J. Grid Comput. 20 (2022) 1–17.
[203] S.S. Gill, et al., SECURE: Self-protection approach in cloud resource management, IEEE Cloud Comput. 5 (1) (2018) 60–72.
[204] I. Ahammad, et al., A review on cloud, fog, roof, and dew computing: Iot perspective, Int. J. Cloud Appl. Comput. (IJCAC) 11 (4) (2021) 14–41.
[205] Y. Mao, et al., A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. Tutorials 19 (4) (2017) 2322–2358.
[206] Q. Luo, et al., Resource scheduling in edge computing: A survey, IEEE Commun. Surv. Tutor. 23 (4) (2021) 2131–2165.
[207] K. Cao, et al., An overview on edge computing research, IEEE Access 8 (2020) 85714–85728.
[208] N. Kotsehub, et al., FLoX: Federated learning with FaaS at the edge, in: 2022 IEEE 18th International Conference on E-Science (E-Science), 2022, pp. 11–20.
[209] O. Almurshed, et al., Adaptive edge-cloud environments for rural AI, in: 2022 IEEE International Conference on Services Computing (SCC), 2022, pp. 74–83.
[210] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: A survey, IEEE Internet Things J. 5 (1) (2017) 450–465.
[211] J. Du, et al., Computation energy efficiency maximization for NOMA-based and wireless-powered mobile edge computing with backscatter communication, IEEE Trans. Mob. Comput. (2023) 1–16.
[212] P. Mach, et al., Mobile edge computing: A survey on architecture and computation offloading, IEEE Commun. Surv. Tutorials 19 (3) (2017) 1628–1656.

[213] Y. Siriwardhana, et al., A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects, IEEE Commun. Surv. Tutor. 23 (2) (2021) 1160–1192.
[214] M. Golec, et al., BlockFaaS: Blockchain-enabled serverless computing framework for AI-driven IoT healthcare applications, J. Grid Comput. 21 (4) (2023) 63.
[215] Z. Zheng, et al., Blockchain challenges and opportunities: A survey, Int. J. Web Grid Serv. 14 (4) (2018) 352–375.
[216] K. Gai, et al., Blockchain meets cloud computing: A survey, IEEE Commun. Surv. Tutor. 22 (3) (2020) 2009–2030.
[217] S.A. Moqurrab, et al., A deep learning-based privacy-preserving model for smart healthcare in internet of medical things using fog computing, Wirel. Pers. Commun. 126 (3) (2022) 2379–2401.
[218] M. Golec, et al., Aiblock: Blockchain based lightweight framework for serverless computing using ai, in: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), IEEE, 2022, pp. 886–892.
[219] M. Kumar, et al., Blockchain inspired secure and reliable data exchange architecture for cyber-physical healthcare system 4.0, Int. Things Cyber-Phys. Syst. (2023).
[220] L. Li, et al., A review of applications in federated learning, Comput. Ind. Eng. 149 (2020) 106854.
[221] J. Yang, et al., A federated learning attack method based on edge collaboration via cloud, Softw. - Pract. Exp. (2022).
[222] C. Zhang, et al., A survey on federated learning, Knowl.-Based Syst. 216 (2021) 106775.
[223] W. Jiang, et al., Federated split learning for sequential data in satellite–terrestrial integrated networks, Inf. Fusion 103 (2024) 102141.
[224] P. Kairouz, et al., Advances and open problems in federated learning, Found. Trends Mach. Learn. 14 (1–2) (2021) 1–210.
[225] G. Wu, et al., Privacy-preserving offloading scheme in multi-access mobile edge computing based on MADRL, J. Parallel Distrib. Comput. 183 (2024) 104775.
[226] M.S. Ferdous, et al., A survey of consensus algorithms in public blockchain systems for crypto-currencies, J. Netw. Comput. Appl. 182 (2021) 103035.
[227] A. Manimuthu, et al., A literature review on Bitcoin: Transformation of crypto currency into a global phenomenon, IEEE Eng. Manag. Rev. 47 (1) (2019) 28–35.
[228] J. Xu, et al., A survey of blockchain consensus protocols, ACM Comput. Surv. (2023).
[229] X. Wang, et al., Blockchain intelligence for internet of vehicles: Challenges and solutions, IEEE Commun. Surv. Tutor. (2023).
[230] U. Rahardja, et al., GOOD, bad and dark bitcoin: a systematic literature review, Aptisi Trans. Technopreneurship (ATT) 3 (2) (2021) 115–119.
[231] M. Golec, et al., IFaaSBus: A security-and privacy-based lightweight framework for serverless computing using IoT and machine learning, IEEE Trans. Ind. Inform. 18 (5) (2021) 3522–3529.
[232] G. Qu, et al., ChainFL: A simulation platform for joint federated learning and blockchain in edge/cloud computing environments, IEEE Trans. Ind. Inform. 18 (5) (2022) 3572–3581.
[233] M. Golec, et al., HealthFaaS: AI based smart healthcare system for heart patients using serverless computing, IEEE Internet Things J. (2023).
[234] S. Svorobej, et al., Orchestration from the cloud to the edge, in: The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing, Springer International Publishing, 2020, pp. 61–77.
[235] W.K. Härdle, et al., Understanding cryptocurrencies, J. Financ. Econom. 18 (2) (2020) 181–208.
[236] P. Weichbroth, et al., Security of cryptocurrencies: A view on the state-of-the-art research and current developments, Sensors 23 (6) (2023) 3155.
[237] A. Schweizer, et al., To what extent will blockchain drive the machine economy? Perspectives from a prospective study, IEEE Trans. Eng. Manage. 67 (4) (2020) 1169–1183.
[238] M. Khan, et al., A review of distributed ledger technologies in the machine economy: challenges and opportunities in industry and research, Proc. CIRP 107 (2022) 1168–1173.
[239] S. Dustdar, et al., On distributed computing continuum systems, IEEE Trans. Knowl. Data Eng. 35 (4) (2022) 4092–4105.
[240] P.K. Donta, et al., Exploring the potential of distributed computing continuum systems, Computers 12 (10) (2023) 198.
[241] A. Morichetta, et al., A roadmap on learning and reasoning for distributed computing continuum ecosystems, in: IEEE International Conference on Edge Computing (EDGE), Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 25–31.
[242] C.J. Beasley, et al., A new look at simultaneous sources, in: Seg Technical Program Expanded Abstracts 1998, Society of Exploration Geophysicists, 1998, pp. 133–135.
[243] S. Aminizadeh, et al., The applications of machine learning techniques in medical data processing based on distributed computing and the Internet of Things, Comput. Methods Programs Biomed. (2023) 107745.
[244] L. Petrou, et al., The first family of application-specific integrated circuits for programmable and reconfigurable metasurfaces, Sci. Rep. 12 (1) (2022) 5826.
[245] K.E. Murray, et al., Vtr 8: High-performance cad and customizable fpga architecture modelling, ACM Trans. Reconfigurable Technol. Syst. (TRETS) 13 (2) (2020) 1–55.
[246] P. Hitzler, et al., Neuro-Symbolic Artificial Intelligence: The State of the Art, IOS Press, 2022.
[247] M. Gaur, et al., Knowledge-infused learning: A sweet spot in neuro-symbolic ai, IEEE Internet Comput. 26 (4) (2022) 5–11.
[248] J. Du, et al., Computation energy efficiency maximization for intelligent reflective surface-aided wireless powered mobile edge computing, IEEE Trans. Sustain. Comput. (2023).
[249] J. Cuadrado, et al., Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments, Multibody Syst. Dyn. 4 (2000) 55–73.
[250] Y. Zhang, et al., Transparent computing: Spatio-temporal extension on von Neumann architecture for cloud services, Tsinghua Sci. Technol. 18 (1) (2013) 10–21.
[251] Q. Jiang, et al., Adaptive scheduling of stochastic task sequence for energy-efficient mobile cloud computing, IEEE Syst. J. 13 (3) (2019) 3022–3025.
[252] D. Bufistov, et al., A general model for performance optimization of sequential systems, in: 2007 IEEE/ACM International Conference on Computer-Aided Design, IEEE, 2007, pp. 362–369.
[253] M.S. Aslanpour, et al., Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research, Int. Things 12 (2020) 100273.
[254] A. Singh, et al., Quantum internet—applications, functionalities, enabling technologies, challenges, and research directions, IEEE Commun. Surv. Tutor. 23 (4) (2021) 2218–2247.
[255] N.P. De Leon, et al., Materials challenges and opportunities for quantum computing hardware, Science 372 (6539) (2021) eabb2823.
[256] K.N. Smith, et al., Scaling superconducting quantum computers with chiplet architectures, in: 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO), IEEE, 2022, pp. 1092–1109.
[257] R.F. Spivey, et al., High-stability cryogenic system for quantum computing with compact packaged ion traps, IEEE Trans. Quant. Eng. 3 (2021) 1–11.
[258] A.R. Nandhakumar, et al., EdgeAISim: A Toolkit for Simulation and Modelling of AI Models in Edge Computing Environments, Meas.: Sensors (2023).
[259] M. Xue, et al., DDPQN: An efficient DNN offloading strategy in local-edge-cloud collaborative environments, IEEE Trans. Serv. Comput. 15 (2) (2022) 640–655.
[260] Y.-L. Lee, et al., Techology trend of edge AI, in: 2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), IEEE, 2018, pp. 1–2.
[261] A.Y. Ding, et al., Roadmap for edge ai: A dagstuhl perspective, ACM SIGCOMM Comput. Commun. Rev. 52 (1) (2022) 28–33.
[262] D. Murugesan, et al., Comparison of biologically inspired algorithm with socio-inspired technique on load frequency control of multi-source single-area power system, in: Applied Genetic Algorithm and Its Variants: Case Studies and New Developments, Springer, 2023, pp. 185–208.
[263] A.K. Kar, Bio inspired computing–a review of algorithms and scope of applications, Expert Syst. Appl. 59 (2016) 20–32.
[264] M. Xu, et al., esDNN: deep neural network based multivariate workload prediction in cloud computing environments, ACM Trans. Int. Technol. (TOIT) 22 (3) (2022) 1–24.
[265] B. Denkena, et al., Reprint of: Gentelligent processes in biologically inspired manufacturing, CIRP J. Manuf. Sci. Technol. 34 (2021) 105–118.
[266] R. Dwivedi, et al., Explainable AI (XAI): Core ideas, techniques, and solutions, ACM Comput. Surv. 55 (9) (2023) 1–33.
[267] A.B. Tosun, et al., Histomapr™: An explainable AI (XAI) platform for computational pathology solutions, in: Artificial Intelligence and Machine Learning for Digital Pathology: State-of-the-Art and Future Challenges, Springer, 2020, pp. 204–227.
[268] A.B. Arrieta, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.
[269] P. Kochovski, et al., Trust management in a blockchain based fog computing platform with trustless smart oracles, Future Gener. Comput. Syst. 101 (2019) 747–759.
[270] K. Shkembi, et al., Semantic web and blockchain technologies: Convergence, challenges and research trends, J. Web Semant. 79 (2023) 100809.
[271] A.D. Córcoles, et al., Challenges and opportunities of near-term quantum computing systems, Proc. IEEE 108 (8) (2019) 1338–1352.
[272] S. Pirandola, et al., Physics: unite to build a quantum internet, Nature 532 (7598) (2016) 169–171.
[273] S. Wehner, et al., Quantum internet: a vision for the road ahead, Science 362 (6412) (2018) eaam9288.
[274] K.C. Seto, et al., From low-to net-zero carbon cities: The next global agenda, Ann. Rev. Environ. Resour. 46 (2021) 377–415.
[275] G. Aceto, et al., A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3467–3501.
[276] G. Aceto, et al., Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0, J. Ind. Inf. Integr. 18 (2020) 100129.
[277] Y.K. Teoh, et al., IoT and fog computing based predictive maintenance model for effective asset management in industry 4.0 using machine learning, IEEE Internet Things J. (2021).

[278] T. Zheng, et al., The applications of Industry 4.0 technologies in manufacturing context: a systematic literature review, Int. J. Prod. Res. 59 (6) (2021) 1922–1954.

[279] W. Yu, et al., Energy digital twin technology for industrial energy management: Classification, challenges and future, Renew. Sustain. Energy Rev. 161 (2022) 112407.

[280] S. Mihai, et al., Digital twins: A survey on enabling technologies, challenges, trends and future prospects, IEEE Commun. Surv. Tutor. (2022).

[281] Y. Wang, et al., A survey on digital twins: architecture, enabling technologies, security and privacy, and future prospects, IEEE Internet Things J. (2023).

[282] M. Kor, et al., An investigation for integration of deep learning and digital twins towards construction 4.0, Smart Sustain. Built Environ. 12 (3) (2023) 461–487.

[283] S. Singh, et al., Qos-aware autonomic resource management in cloud computing: a systematic review, ACM Comput. Surv. 48 (3) (2015) 1–46.

[284] A. Morichetta, et al., Demystifying deep learning in predictive monitoring for cloud-native SLOs, in: 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), 2023, pp. 1–11.

[285] S.A. Wright, Performance modeling, benchmarking and simulation of high performance computing systems, Future Gener. Comput. Syst. 92 (2019) 900–902.

[286] H. Materwala, et al., QoS-SLA-aware adaptive genetic algorithm for multi-request offloading in integrated edge-cloud computing in internet of vehicles, Veh. Commun. 43 (2023) 100654.

[287] Y. Sharma, et al., SLA management in intent-driven service management systems: A taxonomy and future directions, ACM Comput. Surv. (2023).

[288] S. Khan, et al., Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: A survey and open issues, Future Gener. Comput. Syst. 119 (2021) 176–187.

[289] S. Dilek, et al., QoS-aware IoT networks and protocols: A comprehensive survey, Int. J. Commun. Syst. 35 (10) (2022) e5156.

[290] V.C. Pujol, et al., Towards a prime directive of SLOs, in: 2023 IEEE International Conference on Software Services Engineering (SSE), 2023, pp. 61–70.

[291] P. Patros, et al., SLO request modeling, reordering and scaling, in: Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering, 2017, pp. 180–191.

[292] S. Singh, et al., The journey of qos-aware autonomic cloud computing, IT Prof. 19 (2) (2017) 42–49.

[293] P.o. Patros, Investigating the effect of garbage collection on service level objectives of clouds, in: 2017 IEEE International Conference on Cluster Computing (CLUSTER), IEEE, 2017, pp. 633–634.

[294] X. Zeng, et al., SLA management for big data analytical applications in clouds: A taxonomy study, ACM Comput. Surv. 53 (3) (2020) 1–40.

[295] C. Qu, et al., Auto-scaling web applications in clouds: A taxonomy and survey, ACM Comput. Surv. 51 (4) (2018) 1–33.

[296] T. Lorido-Botran, et al., A review of auto-scaling techniques for elastic applications in cloud environments, J. Grid Comput. 12 (2014) 559–592.

[297] P. Singh, et al., RHAS: robust hybrid auto-scaling for web applications in cloud computing, Cluster Comput. 24 (2) (2021) 717–737.

[298] T. Heinze, et al., Auto-scaling techniques for elastic data stream processing, in: Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, 2014, pp. 318–321.

[299] S.S. Gill, et al., Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge, J. Syst. Softw. 155 (2019) 104–129.

[300] S. Bharany, et al., Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy, Sustain. Energy Technol. Assess. 53 (2022) 102613.

[301] S.S. Gill, et al., Failure management for reliable cloud computing: a taxonomy, model, and future directions, Comput. Sci. Eng. 22 (3) (2018) 52–63.

[302] S.S. Gill, et al., Tails in the cloud: a survey and taxonomy of straggler management within large-scale cloud data centres, J. Supercomput. 76 (2020) 10050–10089.

[303] S.S. Gill, A manifesto for modern fog and edge computing: Vision, new paradigms, opportunities, and future directions, in: Operationalizing Multi-Cloud Environments: Technologies, Tools and Use Cases, Springer, 2021, pp. 237–253.

[304] A. Katal, et al., Energy efficiency in cloud computing data centers: a survey on software technologies, Cluster Comput. 26 (3) (2023) 1845–1875.

[305] E. Masanet, et al., Recalibrating global data center energy-use estimates, Science 367 (6481) (2020) 984–986.

[306] S. Iftikhar, et al., HunterPlus: AI based energy-efficient task scheduling for cloud–fog computing environments, Int. Things 21 (2023) 100667.

[307] S. Tuli, et al., HUNTER: AI based holistic resource management for sustainable cloud computing, J. Syst. Softw. 184 (2022) 111124.

[308] T. Schneider, et al., Harnessing AI and computing to advance climate modelling and prediction, Nature Clim. Change 13 (9) (2023) 887–889.

[309] M. Hartmann, et al., Edge computing in smart health care systems: Review, challenges, and research directions, Trans. Emerg. Telecommun. Technol. 33 (3) (2022) e3710.

[310] H.J. Baek, et al., Enhancing the usability of brain-computer interface systems, Comput. Intell. Neurosci. 2019 (2019).

[311] M.H. Miraz, et al., Adaptive user interfaces and universal usability through plasticity of user interface design, Comp. Sci. Rev. 40 (2021) 100363.

[312] J. Diaz-de Arcaya, et al., A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey, ACM Comput. Surv. 56 (4) (2023) 1–30.

[313] I. Celik, Exploring the determinants of artificial intelligence (Ai) literacy: Digital divide, computational thinking, cognitive absorption, Telemat. Inform. 83 (2023) 102026.

[314] S.S. Gill, et al., Transformative effects of ChatGPT on modern education: Emerging Era of AI chatbots, Int. Things Cyber-Phys. Syst. 4 (2024) 19–23.

[315] C. Le Roux, et al., Can cloud computing bridge the digital divide in South African secondary education? Inf. Dev. 27 (2) (2011) 109–116.

[316] C.G.M. Arce, et al., Optimizing business performance: Marketing strategies for small and medium businesses using artificial intelligence tools, Migr. Lett. 21 (S1) (2024) 193–201.

[317] J. Qadir, et al., Toward accountable human-centered AI: rationale and promising directions, J. Inf., Commun. Ethics Soc. 20 (2) (2022) 329–342.

[318] L. Munn, The uselessness of AI ethics, AI Ethics 3 (3) (2023) 869–877.

[319] V. Scuotto, et al., The digital humanism era triggered by individual creativity, J. Bus. Res. 158 (2023) 113709.

[320] J. Schaap, et al., 'Gods in world of warcraft exist': Religious reflexivity and the quest for meaning in online computer games, New Media Soc. 19 (11) (2017) 1744–1760.

[321] D. Magni, et al., Digital humanism and artificial intelligence: the role of emotions beyond the human–machine interaction in society 5.0, J. Manag. History (2023).

[322] Q. Yu, et al., Lagrange coded computing: Optimal design for resiliency, security, and privacy, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1215–1225.

[323] F.O. Olowononi, et al., Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS, IEEE Commun. Surv. Tutor. 23 (1) (2020) 524–552.

[324] Z. Liu, et al., Efficient dropout-resilient aggregation for privacy-preserving machine learning, IEEE Trans. Inf. Forensics Secur. 18 (2022) 1839–1854.

[325] J.K. Samriya, et al., Secured data offloading using reinforcement learning and Markov decision process in mobile edge computing, Int. J. Netw. Manag. 33 (5) (2023) e2243.

[326] I. Ullah, et al., Privacy preserving large language models: Chatgpt case study based vision and framework, 2023, arXiv preprint arXiv:2310.12523.

[327] H. Kim, et al., Resilient authentication and authorization for the internet of things (IoT) using edge computing, ACM Trans. Int. Things 1 (1) (2020) 1–27.

[328] C. Delacour, et al., Energy-performance assessment of oscillatory neural networks based on VO _2 devices for future edge AI computing, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[329] Z. Quan, et al., A historical review on learning with technology: From computers to smartphones, in: Encyclopedia of Information Science and Technology, Sixth Edition, IGI Global, 2025, pp. 1–21.

[330] A. Mijuskovic, et al., Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification, Sensors 21 (5) (2021) 1832.

[331] S. Singh, et al., A survey on resource scheduling in cloud computing: Issues and challenges, J. Grid Comput. 14 (2016) 217–264.

[332] C.-H. Hong, et al., Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms, ACM Comput. Surv. 52 (5) (2019) 1–37.

[333] B. Jamil, et al., Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions, ACM Comput. Surv. 54 (11s) (2022) 1–38.

[334] A. Raju, et al., A comparative study of spark schedulers' performance, in: 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), IEEE, 2019, pp. 1–5.

[335] S. Henning, et al., Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud, J. Syst. Softw. 208 (2024) 111879.

[336] J. Feng, et al., Heterogeneous computation and resource allocation for wireless powered federated edge learning systems, IEEE Trans. Commun. 70 (5) (2022) 3220–3233.

[337] A. Garofalo, et al., A heterogeneous in-memory computing cluster for flexible end-to-end inference of real-world deep neural networks, IEEE J. Emerg. Sel. Top. Circuits Syst. 12 (2) (2022) 422–435.

[338] H. Wu, et al., Collaborate edge and cloud computing with distributed deep learning for smart city internet of things, IEEE Internet Things J. 7 (9) (2020) 8099–8110.

[339] V. Kumar, Digital enablers, in: The Economic Value of Digital Disruption: A Holistic Assessment for CXOs, Springer, 2023, pp. 1–110.

[340] K. Sha, et al., A survey of edge computing-based designs for IoT security, Digit. Commun. Netw. 6 (2) (2020) 195–202.

[341] J.B. Sequeiros, et al., Attack and system modeling applied to IoT, cloud, and mobile ecosystems: Embedding security by design, ACM Comput. Surv. 53 (2) (2020) 1–32.

[342] A. Kaur, et al., The future of cloud computing: opportunities, challenges and research trends, in: 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC, IEEE, 2018, pp. 213–219.
[343] A. Sebastian, et al., Memory devices and applications for in-memory computing, Nature Nanotechnol. 15 (7) (2020) 529–544.
[344] K. Vu, et al., ICT as a driver of economic growth: A survey of the literature and directions for future research, Telecommun. Policy 44 (2) (2020) 101922.
[345] L. Tesfatsion, Agent-based computational economics: Overview and brief history, Artif. Intell., Learn. Comput. Econ. Finance (2023) 41–58.
[346] C. Vairetti, et al., Analytics-driven complaint prioritisation via deep learning and multicriteria decision-making, European J. Oper. Res. 312 (3) (2024) 1108–1118.
[347] A. Jobin, et al., The global landscape of AI ethics guidelines, Nat. Mach. Intell. 1 (9) (2019) 389–399.
[348] R.H. Hariri, et al., Uncertainty in big data analytics: survey, opportunities, and challenges, J. Big Data 6 (1) (2019) 1–16.
[349] L. Cao, Data science: a comprehensive overview, ACM Comput. Surv. 50 (3) (2017) 1–42.
[350] B.K. Daniel, Big data and data science: A critical review of issues for educational research, Br. J. Educ. Technol. 50 (1) (2019) 101–113.
[351] P.K. Donta, et al., Governance and sustainability of distributed continuum systems: a big data approach, J. Big Data 10 (1) (2023) 1–31.
[352] M.H. ur Rehman, et al., The role of big data analytics in industrial Internet of Things, Future Gener. Comput. Syst. 99 (2019) 247–259.