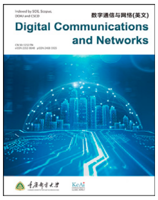
Digital Communications and Networks 9 (2023) 1033–1052

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan



Cache in fog computing design, concepts, contributions, and security issues in machine learning prospective

Muhammad Ali Naeem [a,1], Yousaf Bin Zikria [b,1], Rashid Ali [c], Usman Tariq [d], Yahui Meng [a,**], Ali Kashif Bashir [e,f,g,*]

[a] *School of Science, Guangdong University of Petrochemical Technology, Maoming, 525000, China*
[b] *Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Republic of Korea*
[c] *Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain*
[d] *College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj, 11942, Saudi Arabia*
[e] *Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, United Kingdom*
[f] *Woxsen School of Business, Woxsen University, Hyderabad, India*
[g] *Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon*

ARTICLE INFO

Keywords:
Internet of things
Cloud computing
Fog computing
Caching
Latency

ABSTRACT

The massive growth of diversified smart devices and continuous data generation poses a challenge to communication architectures. To deal with this problem, communication networks consider fog computing as one of promising technologies that can improve overall communication performance. It brings on-demand services proximate to the end devices and delivers the requested data in a short time. Fog computing faces several issues such as latency, bandwidth, and link utilization due to limited resources and the high processing demands of end devices. To this end, fog caching plays an imperative role in addressing data dissemination issues. This study provides a comprehensive discussion of fog computing, Internet of Things (IoTs) and the critical issues related to data security and dissemination in fog computing. Moreover, we determine the fog-based caching schemes and contribute to deal with the existing issues of fog computing. Besides, this paper presents a number of caching schemes with their contributions, benefits, and challenges to overcome the problems and limitations of fog computing. We also identify machine learning-based approaches for cache security and management in fog computing, as well as several prospective future research directions in caching, fog computing, and machine learning.

1. Introduction

Currently, the number of heterogeneous smart devices is exponentially increasing, using the idea of the Internet of Things (IoT) to connect everything together [1]. In IoT, diversified devices such as smart-phone, smart meters, sensors, Personal Digital Assistants (PDAs), and smart vehicles are connected through the Internet to communicate and exchange different kinds of information. Such devices' interconnection facilitates modern IoT-based applications such as energy management, product tracking, patient surveillance, and environment monitoring [2]. Indeed, IoT plays a significant role in multiple domains including e-health-care [3], smart city, intelligent transportation [4], smart grid [5], disaster management, smart homes, and industrial automation. It comprises the new interaction between humans and things to deliver new services and infrastructures that can improve the quality of modern life. In addition, the increasing of heterogeneous connected devices and IoT-based applications leads to excessive data generation with many computing resources such as communication bandwidth, storage, power, and energy [6]. According to Cisco's latest forecast, 50 billion devices will be connected through the Internet at the end of 2021 and this number will reach 500 billion by 2025 [7]. Therefore, a gigantic amount of data is continuously producing and this amount will reach up to 500 zettabytes within 2021 [6]. According to the global data centers, the IP data traffic alone will reach up to 10.4 zettabytes, with IoT-based environments generating

* Corresponding author.

** Corresponding author.

E-mail addresses: malinaeem7@gmail.com (M.A. Naeem), yousafbinzikria@ynu.ac.kr (Y.B. Zikria), rashid.ali@upf.edu (R. Ali), u.tariq@psau.edu.sa (U. Tariq), mengyahui@gdupt.edu.cn (Y. Meng), a.bashir@mmu.ac.uk (A.K. Bashir).

[1] Muhammad Ali Naeem and Yousaf Bin Zikria are the co-first authors.

https://doi.org/10.1016/j.dcan.2022.08.004
Received 1 June 2021; Received in revised form 30 June 2022; Accepted 6 August 2022
Available online 12 August 2022

45% of data, that will be processed, stored, and analyzed at the network edge [8].

Concerning the heterogeneity, some IoT-based applications demand fast data dissemination, are associated with generating a large amount of data volume that could be abundant for the network, and some include private information that should be cached and processed nearby the clients [9]. Besides, IoT-based smart devices are used to collect and transmit multimedia-related data such as voice, photos, and videos, which usually consume a large number of network resources. Consequently, such type of data results in high network congestion and maximizes the processing load on the control system and devices [10]. To address such challenges of IoT-based environments, fog computing has recently been combined with IoT applications to bring the data processing and caching facilities nearby the end-users [11]. Indeed, fog computing distributes the available computing resources (caching, data processing) at the network edges [12,13].

Fog computing is a distributed computing infrastructure that uses IoT devices at the network edges to perform extensive communication through caching facilities collaboratively. It is a most flexible approach to connect the fog nodes with IoT devices that can minimize the processing burden on a device and data retrieval latency for the delay-sensitive applications [14]. Moreover, it can save bandwidth and resources (power and energy). Fog computing is the extension of cloud computing that was developed to distribute computing resources and services close to the consumers. It offers a local cache with data processing capabilities and fulfills the subsequent end-consumers requirements quickly [15]. However, cache management in a fog networks have several challenges. For example, fog computing offers or sales by service providers who want to increase their revenue. Therefore, a critical issue is to incentivize the content providers to sell their fog-based resources concerning the user Quality of Service (QoS).

Besides, fog computing has limited capacity and resources (cache) to efficiently allocate resources to the end-users. Efficient content caching is the core part of fog computing. Therefore, low-quality-based caching schemes can increase the burden on the network and consumes more resources. As an emerging technique, fog network has been increasing the demands for efficient caching schemes. Moreover, the caching schemes should be compatible with IoT-based applications [16]. In addition, the cache-based side channel attacks and data security for the authorized user become a challenge in cache of fog computing [17]. Therefore, Machine Learning (ML) is considered as a promising solution to cope with these challenges, and recently, it has gained significant attention from the research community. ML uses stochastic gradient descent to identify the optimal solution for complex problems [18]. In ML, several flexible techniques have been proposed to provide security and a centralized location for raw data processing. Besides, ML-based schemes facilitate and maximize the performance of each fog node to make the right decisions by caching the right data items. In addition, ML is the most suitable approach to predict the user demands and map the users' inputs with the outputs actions. Moreover, it is used to improve the overall caching performance of a network by identifying the end-users requirements to discover early information from a large number of content streams [19]. The main contributions of the present study are summarized as follows:

- We present a background and overview of fog computing with IoTs in which the adaptation of cloud to fog and their challenges are determined.
- A critical review of existing and related surveys is defined with their contributions and limitations.
- The collaboration and relationship between fog computing and IoT are analyzed for fog-based IoTs environments.
- A review of caching techniques that were developed to improve the fog networks is discussed.
- This survey provides detailed knowledge about the challenges of fog computing and identifies their solutions using caching techniques.
- We also present the role of caching techniques to improve the quality of service in IoT based fog computing.
- We introduce the contributions of caching to achieve high performance of IoT-based fog networks in terms of latency, offloading, energy/power consumption, cost reduction, efficient scheduling, bandwidth minimization, and backhaul link utilization.
- A detailed survey is presented on Machine learning-based techniques to improve the cache management and security risks in fog computing.
- Finally, we collect the issues essential to be addressed and determine important future research direction. Fig. 1 illustrates the taxonomy of this survey.

The paper is categorized as follows. Section 2 presents the background and overview of the computing paradigm. Section 3 describes the critical analysis of related surveys. Section 4 provides knowledge about the collaboration of fog computing with IoT. Section 5 presents the caching concept and fog-based caching schemes. Section 6 describes the caching contributions and schemes to overcome the fog computing paradigm's challenges. Section 7 provides a review of ML-based techniques. Section 8 refers to the summary of security risks and insights of fog computing, caching solutions, caching challenges, and the benefits of ML-based caching techniques. Section 9 presents an overview of issues and future research directions. Finally, in Section 10, we conclude the paper. Table 1 shows the notations and corresponding acronyms.

2. Background and overview

Cloud computing is considered as a resourceful approach to process data due to its flexible storage and power computation characteristics. Most of the computations are executed in the cloud because cloud computing supports a centralized model. Consequently, all the disseminated data is transmitted through the centralized cloud [20]. Therefore, the network links become congested as a large amount of data are transferred, leading to bottlenecks in cloud computing. This may also result in increased data retrieval latency. Besides, some IoT-based applications need mobility support and quick response, such as latency-sensitive applications, smart transportation, smart health-care, smart grid [32,33]. All these applications need emergency responses without delay in transferring data. In addition, some decisions could be taken locally instead of having them in the cloud. However, if there is a need to decide between clouds, there is no need to transfer entire data to the cloud for processing [34]. The reason is that the entire data is not needed for making decisions and analysis. However, the extensive growth of IoT has been imposing serious challenges for the cloud services in which latency, bandwidth, privacy, and reliability are the most difficult to be resolved by using the current cloud computing architecture [21].

To cope with these challenges, the cloudlet was proposed, in which computing resources are distributed as proximity to the consumers for having local processing and storage. It reduces the dissemination cost by minimizing transferring data and decreasing data retrieval latency by processing data near the consumers. Moreover, in cloudlet, the optimal offloading approach reduces the computation and communication costs as well [35]. However, the cloudlet can only be accessed through WiFi access points, which typically have a small coverage area and therefore do not provide ubiquitous computing. Compared to cloud computing, the cloudlet is resource-constrained and cannot meet the requirements of resource provisioning and stable services. As a result, Mobile Cloud Computing (MCC) was developed to provide cloud-based resources and services to mobile consumers and make cloud computing advantageous for the end-users. Some tasks are partially performed at the device connected to the network edges in MCC [36]. In contrast, the cloud is synchronized for data archival [22]. However, MCC is considered to be resource-constrained in terms of physical devices. After all, they have limited cache storage and are resource-constrained in terms of physical

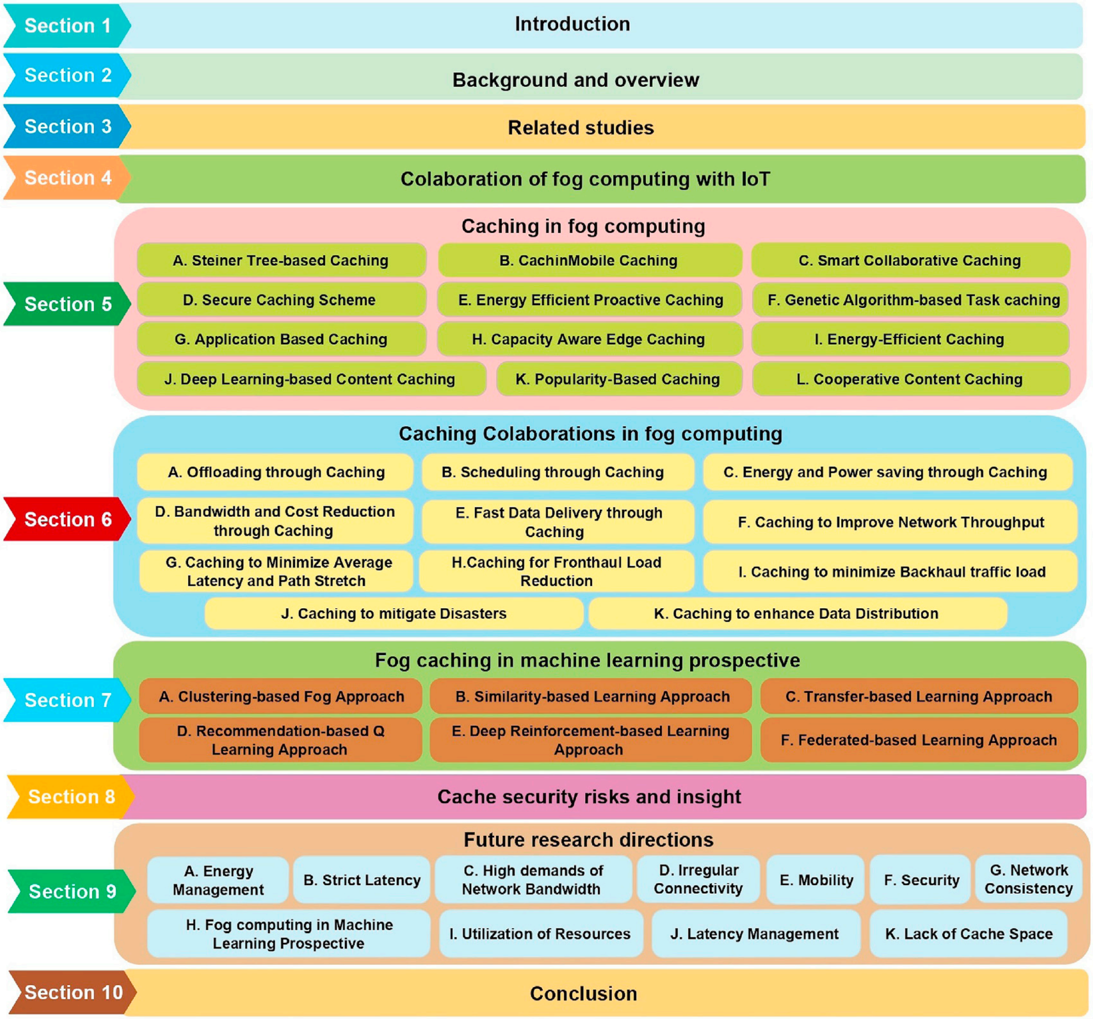

Section 1
Introduction
Section 2
Background and overview
Section 3
Related studies
Section 4
Colaboration of fog computing with IoT
Section 5
Caching in fog computing
A. Steiner Tree-based Caching
B. CachinMobile Caching
C. Smart Collaborative Caching
D. Secure Caching Scheme
E. Energy Efficient Proactive Caching
F. Genetic Algorithm-based Task caching
G. Application Based Caching
H. Capacity Aware Edge Caching
I. Energy-Efficient Caching
J. Deep Learning-based Content Caching
K. Popularity-Based Caching
L. Cooperative Content Caching
Section 6
Caching Colaborations in fog computing
A. Offloading through Caching
B. Scheduling through Caching
C. Energy and Power saving through Caching
D. Bandwidth and Cost Reduction through Caching
E. Fast Data Delivery through Caching
F. Caching to Improve Network Throughput
G. Caching to Minimize Average Latency and Path Stretch
H.Caching for Fronthaul Load Reduction
I. Caching to minimize Backhaul traffic load
J. Caching to mitigate Disasters
K. Caching to enhance Data Distribution
Section 7
Fog caching in machine learning prospective
A. Clustering-based Fog Approach
B. Similarity-based Learning Approach
C. Transfer-based Learning Approach
D. Recommendation-based Q Learning Approach
E. Deep Reinforcement-based Learning Approach
F. Federated-based Learning Approach
Section 8
Cache security risks and insight
Section 9
Future research directions
A. Energy Management
B. Strict Latency
C. High demands of Network Bandwidth
D. Irregular Connectivity
E. Mobility
F. Security
G. Network Consistency
H. Fog computing in Machine Learning Prospective
I. Utilization of Resources
J. Latency Management
K. Lack of Cache Space
Section 10
Conclusion

Fig. 1. Paper taxonomy.

devices because they have limited cache and power capacities [37]. If there is a need to handle multiple IoT-based applications, it will lead to resource contention and increased data processing latency.

Therefore, to address the challenges of cloud computing and cloudlet, fog computing, which is an integrated cloud-centric and network edge device to perform data processing and storage, has recently been designed to enable a more efficient computing architecture. Geographically, it is distributed computing connecting diversified devices at the network edges to provide communication, computation collectively, and storage services [37]. Salvatore J, Stolfo expressed the word fog computing, which was later widely introduced by Cisco, is now accepted worldwide. It is a distributed form of cloud computing that brings out the computing resources (storage, data processing) and services to the network edges [38]. Besides, it allows the applications to run at the end-consumers' devices. As a result, the service latency is reduced, and Quality of Service (QoS) and Quality of Experience (QoE) are enhanced [39]. Fog computing works as a bridge between the cloud and things by providing faster data processing and minimizing overall network costs [23,40]. It supports several applications such as time-critical and Internet of Everything (IoE). Therefore, fog computing can efficiently meet the demands of the latency-sensitive and real-time applications by reducing network resources and bandwidth usage. Table 2 illustrates the summary of different computing architectures with their contributions, benefits, and limitations.

Regarding fog computing structure, an additional resourceful layer is added between the cloud and end-devices to address the present challenges of cloud computing [41]. Besides, it delivers efficient computing services with multiple qualities of high reliability, low latency, mobility, high security, and interoperability. It composes many fog nodes, in which a data management system, multiple edge-devices, and edge data centers are involved in making a complete fog platform [42]. The consumers and devices are connected to fog nodes through wireless mediums, such as 5G, Bluetooth, and WiFi. The data is primarily processed at fog nodes, and then it sent to the cloud data center to perform further processing and storage. Therefore, fog computing plays an important role in providing efficient traditional cloud computing services to network edges and making the computing architecture more affluent [43]. Fig. 2 illustrates the emerging architectures of cloud and fog computing. The end-users and devices are connected to the distributed fog nodes to collect the computing resources and services.

3. Related studies

According to the recent surveys on fog-based computing architecture, the researchers focus on reviewing fog computing definition, related concepts, and representative applications. Besides, they highlight the provided services, virtual functionality, and architecture issues regarding the design and implementation of the fog system. In a survey by Yi et al. [24], a detailed description of fog computing fundamentals is provided. In addition, fog approaches, usage of resources, and services are

Table 1
Notations and acronyms.

Notations	Acronyms	Notations	Acronyms
Personal Digital Assistants	PDAs	Computation Caching Policy	CCP
Quality of Service	QoS	Cache-Based Approach	CBA
Machine Learning	ML	Internet Service Provider	ISP
Mobile Cloud Computing	MCC	Social Aware Edge Caching	SAEC
Quality of Experience	QoE	Cost-Aware	CoA
Internet of Everything	IoE	Efficient Caching Method	ECM
World Wide Web	WWW	Cloud Radio Access Network	CRAN
Wireless Local Area Network	WLAN	Fog Radio Access Network	FRAN
Cyber-Physical System	CPS	Ultra-Dense Network	UDN
Steiner Tree-based Caching	STC	Remote Radio Heads	RRHs
Steiner Tree	ST	Central Processing	CP
Device to Device	D2D	Mobile Social Networks	MSNs
Genetic Algorithm	GA	Most Popular Cache (MPC)	MPC
Smart Collaborative Caching	SCC	Capacity Aware Edge Caching	CAEC
Information-Centric Networking	ICN	Supervised Learning	SL
Secure Caching Scheme	SCS	Un-supervised Learning	UL
Mobile Social Network	MSN	Conventional Neural Network	CNN
Energy Efficient Proactive Caching	EEPC	Bidirectional Deep-recurrent Neural Network	BRNN
User Terminal	UT	Similarity-based Learning Approach	SLA
Access Point	AP	Transfer-based Learning Approach	TLA
Semi Define Relaxation	SDR	Recommendation-based Q Learning	RQL
Mobile User Equipments	MUEs	Popularity Prediction-based Caching Strategies	PPCS
Deep Learning-based Content Caching	ABC	Reinforcement Learning-based Caching Strategies	RLCS
Capacity Aware Edge Caching	CAEC	Reinforcement Learning	RL
Edge Cache Hit Ratio	ECHR	Federated-based Learning	FL
Energy-Efficient Caching	EEC	Application Programming Interfaces	APIs
Application Based Caching	DLCC	Popularity-based Caching Scheme	PCS
Deep Learning	DL	Cooperative Content Caching	CCC
Wireless Sensor Network	WNS	Combinatorial Multi-Armed Bandit	CMAB

Table 2
Computing architectures.

Name	Contributions	Benefits	Limitations
Cloud Computing [20]	It is considered as a resourceful centralized approach to process data due to its flexible storage and power computation characteristics	It provides a centralized platform to process data using robust storage and computational resources	For latency sensitive applications, it has some limitations like latency, bandwidth, and privacy
Cloudlet [21]	It is proposed to provide computing resources that are distributed in proximity to the consumers for local processing and storage	It minimize the cost of transferring data and latency by processing data near the consumers	It can only be accessed through WiFi access points, which typically have a small coverage area and therefore do not provide ubiquitous computing
Mobile Cloud Computing [22]	It is developed to provide cloud-based resources and services to mobile users and make cloud computing advantageous for the end-users	Some tasks are partially performed at the device connected to the network edges in MCC and the latency is minimized	It is considered to be resource-constrained due to constrained oriented devices to handle multiple applications and increase processing latency
Fog Computing [23]	It is an integrated cloud center and network of edge devices to perform data processing and storage to make a more effective computing architecture. Geographically, it is distributed computing, connecting diversified devices in which diversified devices are connected at the network edges to provide communication, storage collectively, and computational services. It works as a bridge between the cloud and things by providing faster data processing and minimizing overall network cost	It brings out the computing resources and services to the network's edges. Moreover, it allows the applications to run at the end-consumers devices to reduce the service latency and improve Quality of Service (QoS). Fog computing can efficiently meet the demands of latency-sensitive and real-tim applications by reducing network resources and bandwidth usage. It supports several applications such as time-critical and Internet of Everything	Regarding fog computing structure, an additional resourceful layer is added between the cloud and end-devices to address the present challenges of cloud computing. It composes many fog nodes, in which a data management system, multiple edge-devices, and edge data centers are involved in making a complete fog platform. It has less storage and resources as compared to cloud computing

summarized to identify the problems such as bandwidth, latency, and energy consumption.

Kitanov et al. [25] a survey in which a review on fog computing, 5G, and their related technologies is presented. Furthermore, this survey describes trustworthy fog computing services for the beyond 5G technology and virtual functionality for fog computing. Hu et al. [26] survey presents a summarized overview of fog computing and its key technologies such as naming, storage, communication, privacy, and security. Moreover, it provides details about the open issues, challenges, and applications of fog computing. Mukherjee et al. [27] present a detailed description of the fundamental of fog computing. Also, fog approaches, resources, and services are summarized to identify the problems such as bandwidth, latency, and energy consumption. In a study by Bellavista et al. [28], the main applications of IoT and their requirements to work with fog computing are described. Furthermore, an overview of the related proposal and integrated platform with fog computing is presented in this survey. Martinez et al. [29] present a survey on the realization and implementation of fog systems in which different aspects of fog computing such as designing, infrastructure resources provisioning of fog computing for IoT-based applications, and resource allocation are described.

Recently a survey by Islam et al. [30] is conducted to explain context-aware scheduling techniques in fog computing comprehensively. Moreover, this survey presents a comparison of scheduling-based techniques based on performance metrics, context-aware, evaluation tools, and case studies. Further, the context-aware parameter analysis, performance metrics, and taxonomy are described in detail. Challenges and issues of fog computing also are defined in this study. The study by Junaid et al. [31] recently is conducted to describe the ML-based techniques for edge networks and determines the factor to improve the caching objectives such as content caching location and replacement. However, based on the current prospective and aim of the present study,

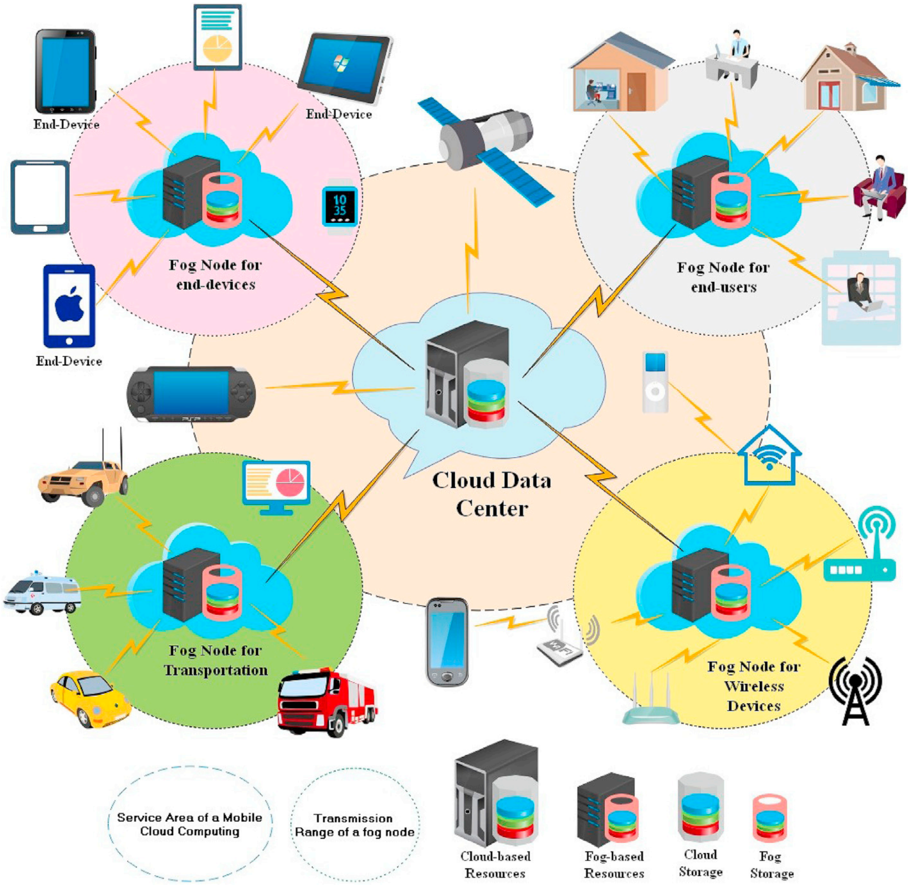


Fig. 2. Computing architecture.

this article does not provide content caching mechanisms and the contributions of caching in fog networks. These surveys neither describes the caching module of fog computing in detail nor discuss its contributions and advantages, as shown in Table 3. However, the present study provides a comprehensive overview of caching modules and their contributions to enhancing the performance of fog computing architecture. Moreover, this study provides knowledge on how caching improves the data dissemination performance of fog nodes. Besides, several caching techniques are described to address the key issues of fog computing, such as data transmission latency, data offloading, bandwidth consumption, link utilization, and energy consumption.

4. Collaboration of fog computing with IoT

IoT is an incredible transformation of the connected trend that is broadly accepted throughout the world. Just the World Wide Web (WWW) was developed to connect computers and people to the Internet, the IoTs are used to connect all places. Indeed, it can interconnect the environments, machines, people, and devices. Moreover, it provides Internet services for the QoS, minimizing the complexity of integrating the physical world with the communication systems. Indeed, it extends the communication services of current Internet technologies, connecting the diversified things (devices and objects) and facilitates their dissemination of the required information [44]. In addition, these connected devices can think, learn, understand, and make decisions about the physical and social world. It is expected that the connected things will behave as autonomous agents and will be able to sense and analyze the environments to make intelligent decisions to accomplish the goals [45].

To achieve the communication goals, an intelligent agent needs to deployed that has the ability to sense the heterogeneous environments to learn and cooperatively adopt scalable and secure intelligent services. In these circumstances, fog computing plays a significant role in improving and achieving the IoT goals [46]. It provides a decentralized platform and extends the cloud services to the network edges analysis and processing on smart IoT-based devices, rather than sending data to the traditional cloud. It is the most promising approach for IoT-based applications as it is used to determine incoming data in real time and works within a limited bandwidth. As a result, several benefits can be obtained, such as less resource usage, less complexity, and higher efficient power and lower energy usage. Moreover, fog computing delivers hierarchical-based three-layers architecture for provides efficient computing services to the IoT end devices [47]. The three layers are the terminal layer, fog layer, and cloud layer, as shown in Fig. 3. The terminal layer is responsible for extending cloud computing services for the end devices and is located near the end-consumers physical environment. It comprises several IoT-based smart devices, such as mobile phones, sensors, smart vehicles, readers, and smart cards, which devices are used to sense the data to process and transmit the sensed data to the upper layer.

The fog layer is composed of a large number of distributed fog nodes, where gateways, access points, routers, switches, fog services, and base stations are merged. This layer is located at the network edge, where fog nodes are extensively dispersed among the end-devices and cloud layers, e.g., bus terminals and shopping centers. The end devices are connected to the fog nodes to obtain the computing services [48]. Both (devices and fog nodes) can perform the computing operation, such as process, transmit, and caches the incoming data. Usually, it is static, performing computing operations from a fixed location, and it can be dynamic (mobile), operating from a moving carrier. Indeed, the fog layer provides cloud resources at the network edges and performs real-time analysis to achieve latency-sensitive applications.

The cloud layer comprises several servers and storage devices that offer high-performance computing resources and services to various

Table 3
Goals and limitation of related studies.

Year	References	Main Focus/Goals	Limitations
2015	Yi et al. [24]	In this survey, the fog computing definition and its related concepts and representative applications are discussed. It highlights the issues of architecture regarding the design and implementation of fog systems	provides a general overview of fog computing architecture, and this survey is limited to the design and implementation of fog systems. It does not provide any detail regarding caching and its implementation
2016	Kitanov et al. [25]	A review on fog computing, 5G, and related technologies are provided in this survey. Furthermore, in this survey, trustworthy fog computing services for the beyond 5G technology and virtual functionality for fog computing are also described	It provides a general overview of fog computing trustworthy technologies for the beyond 5G networks. This study is limited to the introduction of fog-based services to make hybrid environments. However, it lacks of caching knowledge
2017	Hu et al. [26]	This survey summarizes the naming, storage, communication, privacy, and security and provides details about the open issues, challenges, and fog computing applications	This survey is limited to detailing the key technologies of fog computing like security, privacy, protection, communication. However, it presents a summary of the storage technology applied in fog computing
2018	Mukherjee et al. [27]	A detailed description of the fundamentals of fog computing is presented. Besides, fog approaches, resources, and services are summarized to determine critical problems like bandwidth and latency	This survey provides state-of-the-art fog computing network application and research aspects. Indeed, this survey is limited to the basic introduction, research trends, and challenges of fog computing
2019	Bellavista et al. [28]	In this survey, the main applications of IoT and requirements to work with fog computing are described. Furthermore, an overview of the related proposal and integrated platform with fog computing is presented in this survey	This survey describes the IoT-based applications and domains integrated with the fog platform. It is limited to emerging technologies of fog computing and IoT environments. It does not provide any description of fog caching modules
2020	Martinez et al. [29]	A detailed description of the realization and implementation of fog systems that includes designing, infrastructure resources provisioning of fog computing for IoT-based applications, and resource allocation are presented in this survey	In this survey, the design of fog computing and its application for IoT-based environments are discussed. However, It lacks caching-based technologies. It is limited to the basic architecture of fog computing and its installations
2021	Islam et al. [30]	This survey presents comparison and performance of scheduling-based techniques on the basis of performance metrics, context-aware, evaluation tools, and case studies	It provides knowledge about fog computing based scheduling techniques as well as comparison, metrics, and tools. It lacks of fog computing based caching and its implementations
2021	Junaid et al. [31]	This survey presents the ML-based caching techniques, the role of NFV, 5G, and SDN in the edge caching system	It describes multiple factors of ML-based techniques and defines the caching objectives such as location and replacement

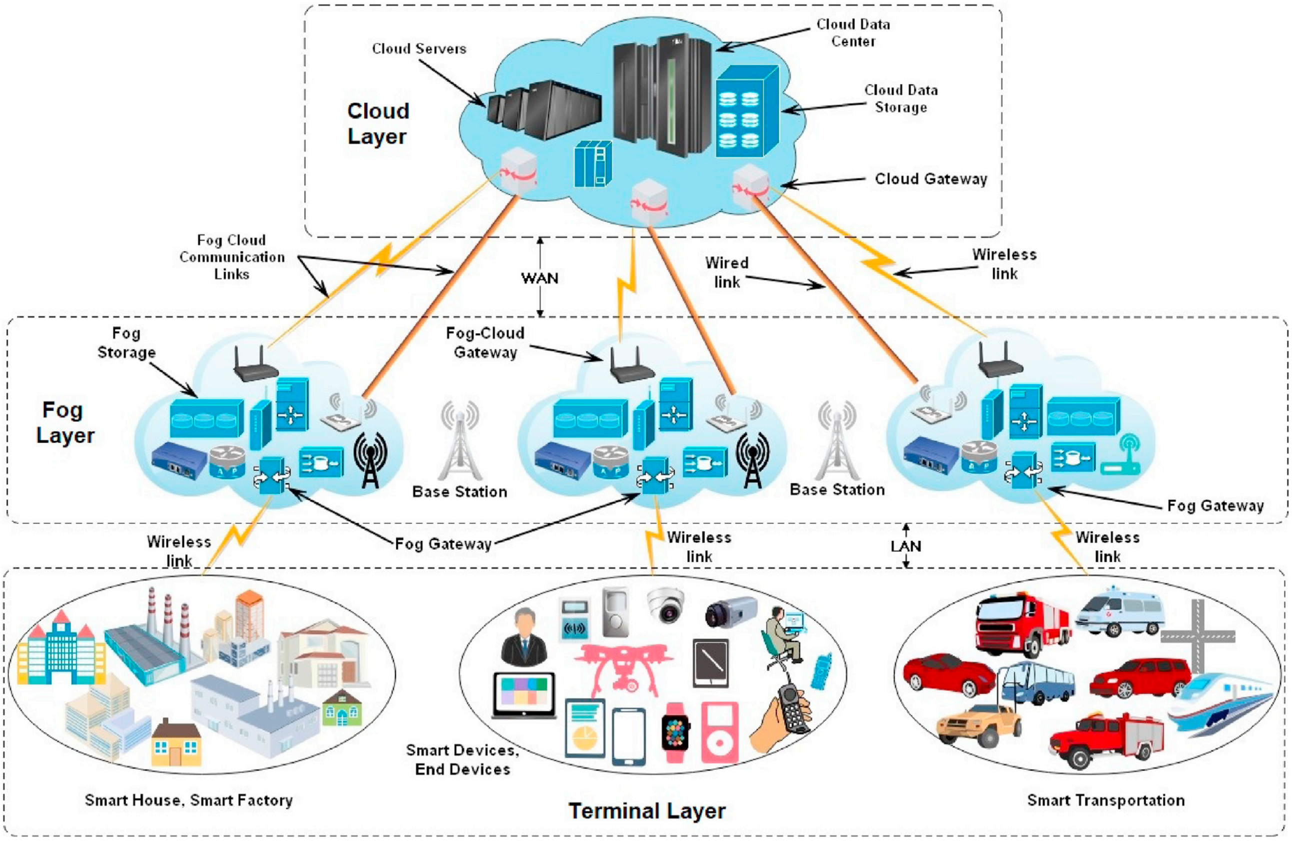


Fig. 3. Three-layer computing architecture.

applications, such as smart transportation, smart home, and smart factories. It has powerful computing resources to perform extensive computation analysis and provides storage to permanently store a huge amount of data [49]. In this computing architecture, each device is connected with a fog node, using a wired connection or wireless mediums, such as WiFi, 4G, 5G, Bluetooth, ZigBee, and Wireless Local Area Network (WLAN). Each fog node can be contented by using wired or wireless connections to the wireless communication technologies (WiFi, 4G, 5G, Bluetooth, ZigBee). It is connected to the cloud through an IP-based network. Moreover, this architecture technically supports the Cyber-Physical System (SPC), IoTs, and mobile networks to provide data storage and processing. Fog computing enhances data dissemination,

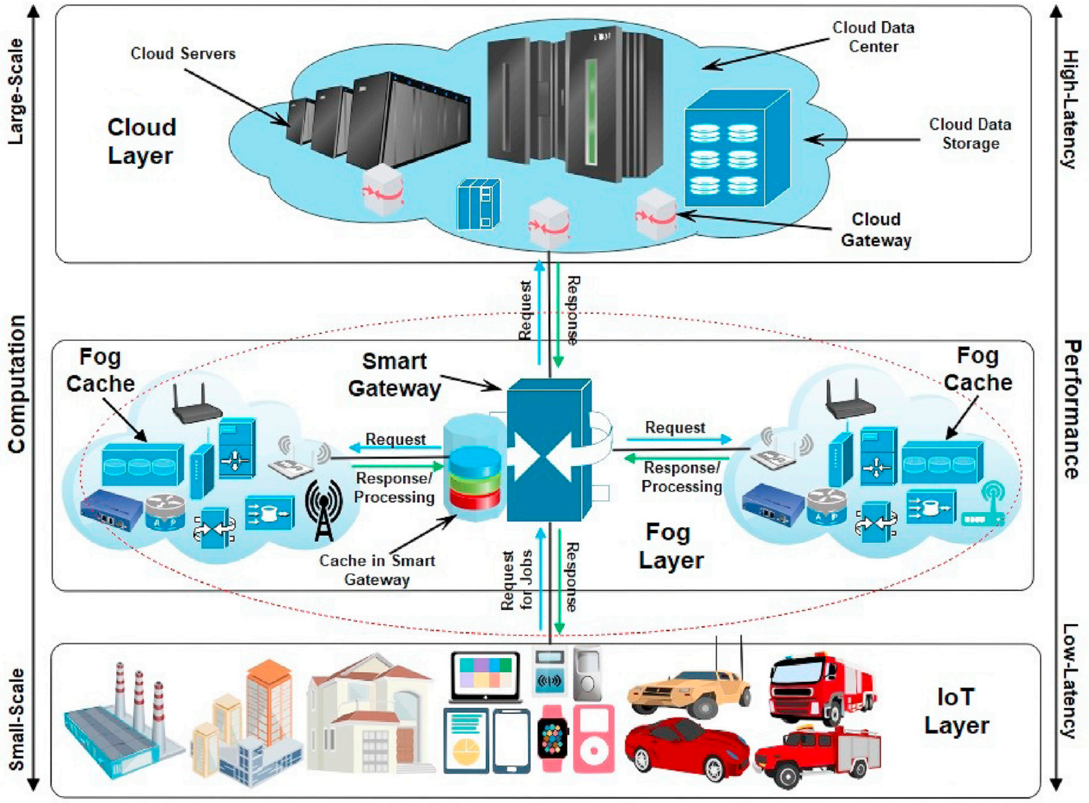


Fig. 4. Caching-based fog computing architecture.

storage efficiency, and QoS in terms of data explosion [50]. Fog computing offers multiple services to the IoT-based environment [51]. For instance, data distribution, scalability, mobility, location, real-time execution, standardization, and, most importantly, temporary storage [44]. In fog computing, temporary storage (cache) plays a significant role in providing enhanced computing services for IoT-based smart environments. The fog-based caching techniques are described and analyzed in the following sections to find the best caching solution for IoT-based environments.

5. Caching in fog computing

Today, an enormous amount of IoT-based data traffic is generated constantly from billions of sensors in heterogeneous devices. These devices can process and store incoming data. However, these devices can not handle such a huge amount of data, even for a short time. Due to IP-based Internet architecture, a large amount of redundant data is created and disseminated which is very difficult to manage, process and store using limited resources (bandwidth, power, energy). To manage such an enormous amount of data in the middle of the network, fog nodes provide cache storage to store data temporarily. Moreover, it also delivers data processing services and increases the availability of desired data to meet the requirements of subsequent future end-users [52]. Furthermore, storage services can analyze, filter, and compress the data to perform efficient data dissemination. It may help to learn about the local information concerning the behavior of a system. Besides, the storage services are capable of enhancing the reliability of a system by providing a proper system behavior to communicate with the end devices [53].

In the fog paradigm, the cloud resources and services are provided at the networks' edges close to the end consumers. Today, it has become an essential requirement to efficiently together and deliver data due to the massive growth in data-generating by smart devices. Therefore, temporary storage (caching) is a promising technique to manage such a huge amount of data. It provides significant approaches to improve the entire network's performance concerning the various aspects, such as response time, reliability, and data retrieval latency, as shown in Fig. 4. In fog-based IoT scenarios, caching is deployed at the fog nodes, significantly reducing the traffic load and computational complexity at the cloud [54]. Thus, several fog-based caching techniques are developed to improve the IoT's productivity and efficiency of fog-based resources and services. In the following sections, different caching mechanisms to efficiently manage fog storage are described.

5.1. *Steiner tree-based caching*

In fog networks, the computing resources are shared or cached using the fog clusters to provide the mobile devices' services. However, it is difficult to share or cache the fog-based resources among the fog servers at a minimum cost. Therefore, in this study, to improve the performance of the fog network, a Steiner Tree-based Caching (STC) is proposed [55]. In STC, the Steiner Tree (ST) is produced to reduce the total cost for a path in such a way as to minimize the usage of fog servers and caching resources. To implement the ST, the graph theory is used to determine the connected tree with the following characteristics: The set of nodes have some predefined nodes, the selected weighted edges should be the original graph, and the sum of weighed edges should be the smallest. Thus, the ST is proposed to analyze caching or resource sharing among the fog clusters. The STC is compared with the traditional shortest path techniques. The results show that the STC can reduce the cost of resource sharing and content caching.

5.2. *CachinMobile caching*

CachinMobile [56] is proposed to improve the energy consumption in fog-based networks. In this technique, the cache of edge nodes is enhanced to increase the responses for the incoming requests using Device to Device (D2D) communication and social network. The requests from the end-users are sent to the caching-based edge nodes via D2D communication. The content is sent to the end-users if it is found at the edge node. Otherwise, the incoming request is forwarded to the neighbor nodes to download the required content. For content caching optimal content placement scheme among the caching nodes is deployed in such a way as to minimize the transmission energy. Moreover, a Genetic Algorithm (GA) is used to find the location for content placement. A group of influential social users are recommended as an edge node for the caching of popular content. The proposed caching scheme is compared with random caching and no caching schemes. The outcomes show that the CachinMobile enhances the content placement by reducing the transmission energy consumption.

5.3. Smart collaborative caching

The IoT data through fog computing is significant to enhance the cacheable nodes in fog-based networks. It is hard to meet the latest demands due to the limited resources and caching capabilities of fog computing. Therefore, a Smart Collaborative Caching (SCC) [57] is proposed by Information-Centric Networking (ICN) rules to improve the caching performance of IoT-based fog networks. The SCC enhances the procedure of content caching, resource pooling, and node locating. The SCC focuses on IoT nodes and data distribution via the ICN and fog networks. Moreover, due to resource-constrained IoT nodes, the data caching approach stores the popular contents based on ICN. During the data caching procedure, the caching node is selected regarding the connecting user and network centrality. The basic contribution of SCC is to enable ICN caching capabilities in IoT to improve the network connection among the devices and provide cooperative caching in the fog network. In addition, content processing, cluster composition, and position management are considered in SCC to enhance the overall performance of fog networks. More specifically, content finding, joining, caching, and leaving were enhanced. A topology with 100 publishers, 100 subscribers, and N clusters is selected to validate SCC. The comparison among IP-based network ICN SCC, ICN flooding, and ICN SCC-Enhanced is made for the performance evaluation. Therefore, the ICN SCC and ICN SCC-E outperform in terms of transmission latency.

5.4. Secure caching scheme

Caching the data at the edges of the fog networks is a promising approach to mitigate the burden on the network links and enhance the quality of experience for mobile users. However, network nodes may be attacked by malicious users, so secure caching becomes a challenge for fog computing. Thus, a Secure Caching Scheme (SCS) [58] is proposed to mitigate the disasters in fog computing-based Mobile Social Network (MSN). The scrambling and partitioning methods are formulated to encrypt data items without increasing size and combined them with additional information to develop a disaster backup caching scheme. These data items are encrypted and sent to the user, while copies of that encrypted data are cached locally on different nodes. Based on the goals of data transmission latency and recovery time, an auction game model is developed to find the best computation to maximize the average utility of the node. The outcomes show that the SCS performs better in terms of improving resource efficiency and data security.

5.5. Energy efficient proactive caching

Energy Efficient Proactive Caching (EEPC) [59] is proposed to improve the caching and data distribution services between fog Access Point (AP) and User Terminal (UT). Therefore, proactive caching is designed for a fog network that consists of an AP and one UT over the finite-time horizon. The correlation among the consecutive delay-sensitive tasks is cached at the current slot to facilitate subsequent computing. A long-term weighted sum energy minimization problem is formulated to predict the length of task input bits. The minimization problem has three slots and inadequate task input prediction, and together enhance the caching decisions and computation offloading. Therefore, an offline solution that depends on Semi Define Relaxation (SDR) is provided to serve as a performance upper bound. Further, a sliding window based on an online algorithm is used to predict error into account. To evaluate the benefits of caching, the proposed algorithm is evaluated by comparing it with the several benchmarks. The outcomes show that the online algorithm with a small window displays exceptional robustness against prediction error.

5.6. Genetic algorithm-based task caching

The Mobile User Equipments (MUEs) have the significant ability to cache the requested tasks to improve the gain using D2D networks. Therefore, the task caching in fog networks can be expressed as follows: requests from multiple MUEs can be satisfied by caching a task locally at neighboring MUEs for multiple MUE requests, or by establishing a distributed task cache, to obtain the maximize gains. Thus, in this study, a task caching scheme [60] is proposed for D2D based fog computing. A task caching optimization is established to improve the gain by caching tasks at the D2D network. In this study, task offloading, MUE association, resource allocation, and task caching are considered to optimize for maximizing the MUEs' utility to reduce the process energy consumption and serving delay. The task caching is used in off-peak time, while the offloading occurs in different periods. Moreover, the caching of tasks is done regarding the requests history where the subsequent requests are unknown. The task caching decision scheme depends on the MUS’ preference. However, task offloading and optimization depend on task caching, such as the current channel condition. The basic goal of caching scheme is to enhance the D2D fog network performance when the subsequent requests and the end-user are unknown. A task optimization problem is formulated to maximize the gains using D2D sharing and local caching. With the help of stochastic theory, a task cache optimization problem is built and solved using a GA-based algorithm to improve the average utility through task caching. Thus, a near-optimal algorithm for task caching is developed based on GA to solve this problem. Furthermore, the task caching benefits for D2D based fog networks are explored. Thus, the result shows that the task caching scheme outperforms well in improving the total utility of the system.

5.7. Application based caching

Application Based Caching (ABC) [16] is recently being developed to improve the caching performance in fog computing. As an emerging paradigm, fog computing significantly increases the demand for efficient caching schemes to maximize the caching gain, and these schemes should be compatible with IoT-based heterogeneous applications. The low-quality caching schemes may sometimes increase the networks’ burden and consume extra resources due to the high miss ratio. To cope with such challenges, ABC is developed to provide caching prediction criteria. Moreover, the variations in IoT-based applications are considered to make caching decisions rather than content popularity, spatial of the temporal locality to determine what to cache. Besides, the IoT devices demand high-performance caching schemes because these devices are combined with limited storage, leading to cache and evicting data contents more frequently. Therefore, it requires a short delay in response to making content retrieval transparent for the end-users. Thus, ABC is designed to implement the data prefetching based on the specified application. The evaluation for ABC is established in the NS-2 simulator in terms of response time, hit rate, and bandwidth consumption. The outcomes show that the ABC has achieves efficient caching performance.

5.8. Capacity aware edge caching

Capacity Aware Edge Caching (CAEC) [61] schemes are developed in which capacity-aware edge caching is designed by considering two factors such as the limited capacity of fog cache and connectivity capacity of the Base Stations (BSs). In CAEC, the edge data distribution is measured as a multiclass processor queuing process and formulates CAEC as an optimization problem in average download time. To find the best performance of CAEC using different cache capacities in fog nodes and last-mile connectivity capacities of BSs, an algorithm is proposed an alternating direction method of multipliers. CAEC improves the data delivery and cache usage in fog networks. In CAEC, the optimal caching place is determined to store the data items so that the data hit ratio is maximized, and the average downloaded time gets reduced. Moreover, it improves traffic allocation balance and enhances the connectivity and cache capacity in fog networks. The experiments are performed between traditional Edge Cache Hit Ratio (ECHR) and CAEC to evaluate CAEC. The results show that it is helpful to use the available cache capacity of the fog nodes to maximize the ECHR, while the connectivity capacity of the BS is sufficient.

5.9. *Energy-efficient caching*

Energy-Efficient Caching (EEC) [62] scheme is proposed to improve fog-based networks' delay and energy efficiency. Several studies are presented to elaborate on the problem of high energy consumption and designed several techniques to reduce the delay and energy consumption in fog computing. In addition, a few studies focus on how to implement the node-based techniques. The cache placement is a difficult problem in the fog-based networks, it is a challenge to achieve efficient data distribution to the end users. It demands the simultaneous determination of several factors like the quality of network connections, content, and user activities. Therefore, to enhance the fog network performance energy-aware scheme, load balancing and content filtration are implemented. In EEC, the frequently requested content is determined through random distribution, and the active fog node is selected based on energy level, number of neighbors, and operational power. Therefore, the chosen content is cached on the fog-based active nodes using the filtration method, and the load balancing method is used to maximize the system efficiency of the cached-based fog network. The EEC is evaluated with a simple caching scheme without caching, and the outcomes show that the EEC achieves better performance in terms of energy efficiency and delay.

5.10. *Deep learning-based content caching*

Deep Learning-based Content Caching (DLCC) [63] is proposed to improve the performance of cache in Fog-Access Point (F-AP). Recently, the proactive caching of frequently accessed content in F-AP-based cache is considered as a promising approach to reduce delay-related problems caused by varying requirements of multimedia data traffic. Due to dynamic user preference, it is difficult to determine the users' content to cache efficiently at F-Aps. Several studies implement Deep Learning (DL) to predict the future popular contents that may solve the content placement problems in the F-AP cache. Therefore, DLCC helps to cache the future predicted content in fog-based networks proactively. In DLCC, a 2D conventional neural network-based method is formulated to optimize the caching model. The performance of DLCC is compared using different Zipf probability distributions with randomized replacement and transfer learning cooperative caches and evaluated. The results show that the DLCC has greater prediction accuracy than the benchmark. Moreover, DLCC achieves high cache hit performance and significantly reduces the delay.

5.11. *Popularity-based caching*

The emergence of the Internet of Everything (IoE) has put several challenges for fog computing to deliver user-requested data efficiently. As the fog nodes provide data storage and processing at the geographically distributed IoT heterogeneous devices. The latency becomes a crucial issue for the continuous increase of time-sensitive applications in fog computing. Therefore, to increase the system efficiency and QoS, a content Popularity-based Caching Scheme (PCS) [56] is proposed for fog computing. The incoming requests are measured to identify the popular data item based on IoT user interests to implement the popularity-based data caching. The users are grouped into clusters based on their interests, and each cluster is attached to the nearest fog nodes. The popular data items are cached at the fog nodes attached to the clusters according to the PCS. Thus the overall service delay is reduced, and the throughput of the system is maximized. However, to further minimize the delay, the D2D communication is performed in case of a cache miss. Moreover, the association rules are formulated for the prediction of future demands. Thus, PCS displays better outcomes to reduce latency and increase the cache hit ratio by caching the popular data items at fog nodes close to the IoT devices.

5.12. *Cooperative content caching*

The latency demands and increasing data traffic requirements by emerging IoT-based applications have continuously posed new challenges for efficient data delivery. Therefore, Cooperative Content Caching (CCC) [64] is proposed to cope with these challenges. The CCC proactively caches the data items at the node to reduce the overall data retrieval latency in fog computing. Moreover, it provides an allocation power scheme that dynamically enables users to fetch their desired data items from the nearest F-APs, or the data can be retrieved from the UEs using D2D communication. Furthermore, the CCC efficiently allocates transmission power to improve the transmission rate for the D2D based UEs. The CCC problem is expressed as a Combinatorial Multi-Armed Bandit (CMAB) framework. A multi-agent reinforcement learning-based algorithm is proposed by considering popularity prediction and user preference to deploy an optimal caching strategy. To reduce the data retrieval latency for each UE, the power allocation issue is designed to increase the sum of data rates of end-users. Thus, a Q-learning power allocation scheme iss developed. In evaluation, both the power allocation strategy and CCC performed better in latency and cache hit ratio. Table 4 shows the summary of the above caching strategies with their goals and achievements.

6. Caching contributions in fog computing

Caching plays an important role in fog computing to improve the overall IoT-based fog networks. In this section, the fog-based caching contributions and appropriate caching techniques are determined to find the optimal solution for improving fog-based IoT networks' performance.

6.1. *Offloading*

Data offloading is one of the most prominent methods that outsource low power devices (smart wearable, smartphones) to perform processing for a given task with higher capabilities and resources. Offloading will not be beneficial if the computational requirements are not as much of a communication cost [65]. Therefore, a device can perform offloading, while offloading benefits are important compared to execution costs [66]. Thus, offloading's ultimate objective is to minimize the overall processing for a device to increase the battery life. Today, the rapid growth of IoTs has been leading to explosive data generation in both engineering and commercial fields. Besides, many heterogeneous smart devices, objects, and sensors are integrating by using IoT platforms. These physical devices, such as smart devices and sensors are linked together through Internet connections and make a Network [67]. In addition, it is challenging to find the appropriate resources for the execution in offloading, and the task cannot be executed efficiently if the resources are insufficient. Thus, the overall network performance decreases. Besides, the allocation of computing resources to compute the dynamic entries is another offloading problem in IoT-based scenarios [68].

Now, scaling in offloading is the most difficult task to perform in large

Table 4
Goals and achievements of existing caching techniques for fog computing.

Year	Reference	Aim/Goal	Achievements
2015	Su et al. [55]	Steiner Tree-based Caching (STC) is proposed to reduce the total cost for a path so as to minimize the usage of fog server and caching resources	It minimize the cost of resource sharing and reduces the cost of content caching at fog nodes
2016	Wang et al. [56]	CachinMobile provides optimal content caching between nodes and uses GA-based algorithm to minimize transmission energy. A group of influential social users are recommended as an edge node for the caching of popular content	It improves the content placement and reduces energy consumption in data transmission
2017	Song et al. [57]	Smart Collaborative Caching (SCC) is proposed ICN rules to improve content caching, resource pooling, and node locating. Due to resource-constrained IoT nodes, the data caching approach stores the popular contents based on network centrality	SCC implements the ICN content caching rules in fog computing and reduces the data transmission latency
2018	Su et al. [58]	Secure Caching Scheme (SCS) is proposed to develop a disaster backup caching scheme. The scrambling and partitioning methods are formulated to encrypt data items without increasing size and combined them with additional information. The encrypted is sent and cached in SCS	SCS reduces disasters by encrypting data in fog computing, improves resource efficiency, and increase data security
2019	Xing et al. [59]	Energy Efficient Proactive Caching (EEPC) is proposed to improve the caching and data distribution services between fog Access Point (AP) and User Terminal (UT)	EEPC enhances the striking robustness against prediction error
2019	Lan et al. [60]	In task caching, task offloading, MUE association, resource allocation, and task caching are considered to optimize for maximizing the MUEs' utility to reduce the process energy consumption and serving delay	It Caches the data of requested tasks to improve the gain using D2D networks and Improve system utility
2020	Almobaideen et al. [16]	Application Based Caching (ABC) is developed to provide caching prediction criteria by considering variations in IoT-based applications to make caching decisions. More specifically, ABC implements the data prefetching based on the specified application	ABC improves bandwidth consumption. It reduces content retrieval latency. It increases the overall cache hit ratio
2020	Li et al. [61]	Capacity Aware Edge Caching (CAEC) schemes are developed to develop capacity-aware edge caching by considering two factors, such as the limited capacity of fog cache and connectivity capacity of the Base Stations (BSs)	CAEC improves data delivery and cache usage and the Improve usage of cache capacity and connectivity capacity of BS
2020	Shahid et al. [62]	In Energy-Efficient Caching (EEC), the frequently requested content is determined for caching through random distribution. The active fog node is selected based on energy level, number of neighbors, and operational power	EEC achieves better energy efficiency and minimizes the data retrieval delay
2021	Bhandari et al. [63]	In Deep Learning-based Content Caching (DLCC) a 2D conventional neural network-based method is formulated to optimize the caching model that helps to cache the future predicted content in fog-based networks proactively	DLCC achieves a high cache hit rate and reduces the data retrieval delay
2021	Gupta et al. [56]	Caching Scheme (PCS) implements popularity-based data caching. The incoming requests are measured to identify the popular data item on the basis of IoT user interests, and the popular data is cached at the fog nodes that are attached to the clusters	PCS reduces the overall service delay, improves the system throughput, and increases the cache hit ratio
2021	Jiang et al. [64]	Cooperative Content Caching (CCC) provides an allocation power scheme that enables the users to fetch their desired data items from the nearest F-APs dynamically, or the data can be retrieved from the UEs using D2D communication	CCC reduces the overall data retrieval latency in fog computing and enhances the overall cache hit ratio

scale IoT-based scenarios. The reason is that available IoT-based projects are combined with approaches that are usually ad hoc in nature [69]. These approaches cannot be migrated or scaled in a new environment [70]. Besides, communication between the cloud and the IoT becomes challenging due to the lack of IoT standard platforms that can be used for low-power devices for offloading in large-scale IoT-based environments. Hence, if there are a large number of resources available for offloading, the task will not be exploited. Conversely, if fewer resources are available, extensive downloading will be required to complete the offloading process [71].

Fog caching is considered to be the most promising approach to address these challenges and perform offloading efficiently. Fog caching offers distributed storage to manage the large amount of data generated by IoT devices, reducing the load on the entire network and edge devices. Therefore, distributed fog cache works with the end devices to perform offloading and improve the process's overall efficiency. Moreover, it reduces the time of execution for a task [72].The limitations of IoT devices to perform offloading can be minimized by performing the computational offloading at fog cache instead of low power devices. Therefore, it is an impressive way to offload the computational tasks using distributed cache-based fog approaches to reduce the computation overhead significantly. Also, the IoT devices will reduce the latency, response time, and energy consumption by task offloading through caching [73]. Consequently, the data is cached at the network edges near the end devices, and the offloading is performed with short latency.

Many fog-caching offloading techniques are recently developed [74], such as an energy consumption oriented offloading algorithm for fog computing. With this technique, an offloading algorithm is developed to reduce energy consumption [75]. A caching scheme named GA-based task caching mechanism is proposed by Lan et al. [48] to support offloading in D2D networks. In this scheme, caching is used to enhance the offloading performance of fog in terms of average caching utility, the utility of multiple mobile user equipment, and the total utility. Computation Caching Policy (CCP) [76] is proposed to enhance the offloading capabilities by measuring three primitives: the popularity of task, input, and output sizes. CCP reduces the uplink traffic between User Equipment (UE) and serving small cells. Moreover, it decreases the computation and communication cost and increases the capacity to offload more tasks. Besides, it reduces the usage of computational resources and offloading latencies.

6.2. Scheduling

Today, the IoT ecosystem's swift development is associated with billions of devices that connect and communicate. However, these devices are imposing stringent latency, power consumption, processing delay, and execution time requirements. Fog computing technology is proposed to process data at the gateway or provide device-level processing to cope with these acute requirements. Fog nodes are connected collaboratively to provide elastic computation resources and services, such as storage and processing [77]. However, fog computing offers promising features. It is still facing high latency problems due to a lack of appropriate resources and scheduling algorithms. Indeed, it depends on several factors that make resource allocation a challenging task in fog computing, such as resource scarcity, geographic restrictions, heterogeneity, and varying demands of the resources [78].

The primary objective of resource allocation and scheduling is to maximize the efficiency of using the resources, increases the profit of fog nodes and IoT devices, and meet the Quality of Services (QoS) requirements. Most of the fog techniques focus on minimizing the performance of execution time and latency minimization. However, there is a significant gap in these techniques to perform resource re-allocation to the jobs. Therefore, the overall efficient resource allocation and scheduling are reduced [79]. Since the beginning of IoT and fog computing paradigm become an enormous evolution in the field of the connected world, a huge number of applications, such as online interactive gaming, augmented reality, face recognition, and natural language, are integrating and attracting many researchers to explore procedures or methods that can efficiently perform device-level computing [80]. However, these kinds of applications are computing incentive or data incentive that needs huge resources and usually consumes extensive energy. Hence, caching is a promising approach to perform efficient tasks' execution requested by end-devices in addressing job scheduling issues. Indeed, the cache is equipped within the gateway to perform scheduling.

Caching algorithms are deployed in the network to minimize the propagation delay, internal processing time, and execution time for the jobs [81,82]. The cache-based job scheduling algorithms deliver several advantages over traditional fog computing, such as lower latency, shorter execution time, lower power consumption, and less processing delay. Caching is one of the most flexible technologies to speed up data retrieval and improves system efficiency. Usually, caching offers storage to absorb data traffic by caching the frequently requested data items near the edge devices, provides low processing cost, and eliminate the single point of failure [83]. Therefore, the combination of fog computing and caching enables the fog node to identify the user requirements and select the most appropriate data item to be cached at intermediate fog nodes [84]. Caching not only decreases the latency and execution time via providing fast processing but also minimizes the burden on network, power, and energy consumption [85].

Recently, a cache-enabled fog computing job scheduling technique named Cache-Based Approach (CBA) is proposed in Ref. [86]. This technique integrates caching and the concept of a smart gateway to solving the resource re-allocation problem in fog computing by providing a CBA scheduling algorithm. CBA incorporates the cache module into the smart gateway to store the job and information. Therefore, CBA significantly increases the overall scheduling performance by reducing the latency, execution time, power consumption, and internal processing delay.

6.3. Energy and power saving

The demand for energy/power is exponentially increasing due to the immense usage of advanced mobile applications. Fog computing is a novel technique with the extra potential to deliver the desired data close to the end-users. It enables future technologies to provide efficient data dissemination services to end devices and users. It consumes less energy than cloud computing to perform task offloading, computation, and data delivery. It delivers energy-efficient techniques to provide computing services near the end-users. It reduces the energy consumption of the cloud server as it is closed to the network edges. However, energy efficiency becomes a critical issue with the continuous expansion of fog networks on a large scale.

Many types of research are conducted to minimize the energy consumption in fog networks in which the researchers keenly focus on job scheduling to reduce the energy consumption. In addition, the node-based techniques neglect to improve energy efficiency [86]. Moreover, if the fog network becomes energy efficient, the overall performance of the network will be improved. In addition, fog computing delivers Access Points (APs) at the edge of the network with storage and computing capabilities that provide computing resources and services to the low power edge devices. These wirelessly connected devices can find the computing services from the nearby cached-based APs for task offloading. In this way, the energy-saving computation can perform in real-time. Indeed, edge caching alleviates the air traffic and saves the edge servers from performing repeated computation by caching the popular data items at the network edges [87].

Now, fog computing is seen as the most promising paradigm that uses high energy to support IoT-based applications. Fog nodes are equipped with small batteries that can recharge by renewable energy resources such as wind turbines and solar panels. Several techniques are developed to reduce the energy consumption in fog computing. The energy consumption can be reduced by improving the data transmitting power, network size, and remote radio unit density. However, these kinds of mechanisms can reduce energy consumption while the network load is very small. Moreover, these mechanisms do not involve fog features and caching capabilities. Furthermore, caching is the most significant approach to deliver fog services during peak traffic demands. It can meet the user requirements by sending the cached data items during off-peak traffic [88].

Caching plays a significant role in coping with energy efficiency by caching popular contents at a network node, which improves the quality of energy consumption. To improve the energy efficiency in fog networks, a popularity-based caching technique is developed in Ref. [62]. With this technique, energy-aware mechanisms such as load balancing and content filtration are implemented. The load balancing mechanism is selected to improve the system efficiency in the cached-based fog network. However, the filtration mechanism is used to cache the popular data items on active nodes. Therefore, the experiments show that the proposed caching technique consumes 92.6% less energy than a network without caching. In another study by Xing et al. [59], dynamic-based computation caching is proposed to reduce the computation burden. Moreover, it reduces the overall energy consumption.

6.4. Bandwidth and cost reduction

Cloud computing is a platform where the system offers services and resources over the Internet. It provides highly scalable and on-demand computing capabilities to enhance the overall data dissemination for the end-users. The reason for moving computation tasks and storage to fog computing is to provide extra capacity and power compared to the limited resources of the end devices. Data traffic is growing exponentially due to massive use of IoT-based applications and devices, which are connected with the fog networks causing bottlenecks and congection due o bandwidth constraints [89]. Therefore, the research community is trying to find an optimal approach that can reduce bandwidth consumption.

However, caching is considered as a promising technique that gains popularity to deliver computing resources and storage to perform tasks at the network edges. It reduces the data delivery distance and the traffic load on network channels. Therefore, a large volume of data is processed at the network edges as an alternative to the cloud. Thus, a very small number of tasks remain to perform at cloud servers, which significantly reduces bandwidth consumption. Hence, a large amount of bandwidth could be saved in caching based networks [90]. Besides, reduction in data traffic and operational cost for the Internet Service Provider (IPS) is a significant aspect of cache-able networks. Consequently, the overall load on the provider links and transit becomes minimized. It shows that caching can reduce inter-domain costs, and inter-domain costs are the most crucial issue, as inter-domain traffic grows 60% in a faster year compared to the cost reduction offered by the current technologies [91]. The inter-domain cost can be minimized by increasing the cache space. However, it will increase capital expenditures. This trade-off can be tackled by deploying an optimal caching scheme. Currently, caching is implemented to provide benefits from the user perspective and reduce the ISP cost. In caching, the content retrieval cost is considered in-network operations to make the network profitable for ISPs.

Social Aware Edge Caching (SAEC) [92] is proposed to minimize the bandwidth consumption in fog computing. According to SAEC, the most requested data items are cached near the end-users at the network's edges. Therefore, the stretch between the user and the locally stored data is reduced. The bandwidth is significantly improved because there remain fewer requests to forward the remote cloud server. In Cost-Aware (CoA) [93] caching, cost reduction is considered as a basic goal for the benefit of ISPs. CoA considers the operational cost to reduce the load on network channels.

6.5. Data availability

Today, the rapid growth of technologies and IoT applications, such as automotive electronics, home appliances, sensors, and actuators, are incorporated in a sole hub known as the Internet of Everything (IoE). These devices facilitate humans' daily lives by providing intelligent services to various applications, such as smart cars, smart transportation, smart home, e-health care, and augmented reality [94]. Moreover, the IoT sensors have limited computing resources that cannot provide efficient QoS requirements to IoT-based applications. Therefore, task and data offloading are considered to be flexible approaches to earn profit for IoT-based applications [95]. For the remote execution, these devices are used to offload intensive data to the centralized cloud server.

Indeed, the cloud provides storage and processing resources to improve overall cloud-IoT communication. Consequently, the cloud executes processing on the data received from the heterogeneous devices and sends them back. Accordingly, the cloud paradigm delivers standardization for the communication of diversified IoT devices. However, the centralized cloud cannot satisfy the high demands of IoT-based time-sensitive applications such as augmented reality, voice and face recognition, image and video processing [96]. Since the distance between the cloud and end-device is extremely large, which increases the service latency during data dissemination. A standard approach is required to perform efficient communication for such heterogeneous devices.

To cope with these challenges, researchers suggest using cache-based fog computing approaches that leverage the processing and caching facilities near the IoT devices to perform intensive operations on data [97]. It delivers distributed resources and fetches the cloud computing services at a single hop distance from the end devices. Consequently, the latency-sensitive applications can be computed at cacheable fog nodes, and there is no need to send the applications to the cloud for execution. Since the fog nodes are deployed at a single hop stretch, it provides agile computation with reduced data retrieval latency for the end-users and IoT devices. However, fog storage capacity is much smaller compared to the data generated by IoE device, which means that there are key challenges in achieving efficient data dissemination performance. As a result, a limited amount of data can be stored at fog nodes.

To increase the storage performance in terms of throughput and latency, caching provides several techniques to cache popular data and enhance the overall network performance. In fog-caching, the popular data items are cached near the end devices to meet the subsequent requests. In addition, the incoming requests are sent to the cloud when the required data item is not available at the fog node, and hence, the overall latency is increased. Therefore, cache maximizes the availability of popular and desired data and reduces the average latency and execution time. Moreover, it decreases the computational cost, processing, and network load. Besides, it reduces the access latency and improves the overall throughput of the fog nodes.

Recently, a popularity-based caching scheme named Efficient Caching Method (ECM) by Riya et al. [64] is proposed to improve fog computing-based IoE environments. In this caching scheme, users' interest is measured proactively and form group clusters of users with similar interest, and the cluster is mapped within the fog node. ECM maximizes the data hit rate and network throughput. Moreover, it reduces the average latency.

6.6. Throughput

With the rapid production of various types of advance mobile applications and a massive amount of connected devices, fog computing are facing unprecedented data traffic. Although, cloud computing can provide stable and reliable services the end-devices, data traffic's continuous proliferation implies inconceivable pressure on Cloud Radio Access Network (C-RAN) due to the limited capacity of front-haul and back-haul links. This may cause interruption or congestion in data dissemination, especially at off-peak periods. Since some social applications are becoming more popular, and the redundant data traffic over the network channels is frequently transferred, which increases the network overhead. In this case, an efficient solution is required to transfer the cloud resources to the edges of the networks and provides the facility to perform computation and caching of popular data. As a result, fog computing technology moves the computation of the cloud towards the network edges. It can overcome the core issues of data explosion in IoT-based environments. Instead, it is now processed at the network edges to send the raw data toward the cloud. However, a large amount of raw data is still being processed, which creates massive communication congestion in the whole network [98]. Therefore, caching is considered to be the most favorable approach to overcome such problems.

To improve system performance, cache memory is distributed with fog nodes to perform caching of popular data items close to the IoT devices an approach that minimize communication overhead and latency. Moreover, caching of the frequently fetched data items near the IoT devices increases the data hit rate, and hence, the overall throughput of the network is improved. Now, the Fog Radio Access Network (FRAN) provides caching facilities that can effectively minimize the front-haul congestion by caching the popular data items near the edge devices. It distributes edge caching, which is a key component to improve the overall throughput of fog computing. Recently, Jiang et al. [99] proposed a popularity-based caching scheme to improve the fog architecture. In this scheme, the frequently requested data items are selected through popularity prediction and locally cached the items chosen near the end devices in real-time. Therefore, the subsequent requests are accomplished there, which increases the overall data hit performance.

6.7. Latency and path stretch

Currently, latency is becoming a challenge for data transmission in modern applications such as virtual reality augmented reality, ultra-low latency dissemination, and computation [100]. The efficient latency requirements for such types of applications cannot be achieved through the traditional cloud because it needs a round trip delay for the successful transmission [101]. To cope with such delay-based issues, fog computing is offered in a distributed cloud manner in which fog nodes provide facilities to perform key functions (data computation and storing) at the network edges near the end-devices [102]. Besides, fog nodes are distributed in physical proximity, bringing the cloud resources at the network edges to provide low latency computations. However, fog computing still cannot meet the satisfactory level of efficient latency requirements and face delay-related challenges. For example, the limited capacity of backhaul and forward links leads to delays in data transmission. In these cases, the caching function plays an important role in improving the overall data transmission. It also plays an in important role in improving the system throughput by reducing the data transmission distance and latency. For example, fog caching in a vehicular network can minimize the distance covered by the data in a network.

In a study by Mohammed et al. [100], a popularity-based caching strategy is proposed to reduce the content retrieval latency. The tasks are categorized according to the requested frequency and determined by the popular tasks in the given caching strategy. The popular tasks are proactively cached using cloudlet. Consequently, the computation latency is minimized, and the cost of computing delay is reduced. Thus, according to Mohammed et al. [100], the proposed caching scheme can minimize computational latency up to 91%. In another study by Lee et al. [103], a caching scheme is proposed to improve fog networks' latency performance. An Online Computational Caching (OCC) is implemented in a fog network that optimizes the input intermediate computational results on the arrival of users' operations. It reduces computational latency and transmission latency and can minimize the total latency by up to 27%.

6.8. Fronthaul link load

Due to the exponential demand for mobile applications and smart devices, the development of mobile networks and the phenomenal growth of data traffic has created significant problems for the entire communication architecture. Currently, Ultra-Dense Network (UDN) is considered as a promising approach for mobile networks to meet the demands of explosive data traffic. However, such a huge volume of data traffic causes trouble for the communications through fronthaul and produces high congestion on network links. Cloud Radio Access (CRAN) and Fog Radio Access Network (FRAN) are considered as promising techniques to alleviate the high demands of fronthaul link capacity by deploying the BaseBand Units (BBUs) close to the fog nodes and near the end-users. FRAN is a promising technique that can significantly improve the 5G cellular networks' spectral efficiency via fog computing and CRAN. To this end, in the FRAN system, the User Equipments (UEs) and Remote Radio Heads (RRHs) are capable of performing cooperative ratio resource management, signal processing, and caching [104].

Nevertheless, the distributed edge caching plays an imperative role in reducing the load on fronthaul channels effectively. Since the sharing of popular data items using social applications produces a huge amount of mobile data traffic. However, in the caching system, multiple users' demands for similar data items will be satisfied without duplicate transmission if the most requested data item is cached at the edge node. The core objective of caching in FRAN is to minimize the heavy burden on fronthaul and radio access networks. Recently, in Ref. [105] a caching scheme is proposed in which popular data items are cached within the FRAN network to reduce the dissemination rate under the limited capacity of fronthaul.

In another study [106], a comprehensive analysis is presented to enhance the total data delivery delay using fronthaul. Moreover, in this study, the analysis is established to find the optimal caching scheme to improve the performance of fronthaul.

6.9. Backhaul traffic load

Fog Radio Access network (F-RAN) is recently proposed to enhance a fog computing network's overall performance. In F-RAN, the Radio Units (RUs) are equipped with cache storage that helps to store frequently accessed data items. The user requests are served from the cluster of RUs, and the local cache of F-RAN serves multiple requests for the same data items, or it can be fetched from the remote data center via backhauls links [105]. In this way, caching of the frequently fetched data items can significantly minimize the load on backhaul links. C-RAN is considered as a promising approach to enhance interference management because of Central Processing (CP) [107] to optimize the spectral efficiency in data channels. A simple way to improve spectral efficiency is to minimize the distance between the sender and the receiver. A number of low-cost RUs are densely deployed to cope with this, and these low-cost RUs are connected to CP through backhaul links [108]. Therefore, the backhaul links between RUs and CP become congested due to limited capacities of backhaul, and thus, overall network performance is reduced. Offering local cache within RUs is shown as the most flexible approach to mitigate congestion in backhaul links.

Thus, F-RAN provides cache space for popular and frequently requested data items to be cached at the network edges without the need to retrieve them from the remote CPs via backhaul links [109]. Therefore, at higher spectral efficiency, the overhead is minimized and the overall content retrieval latency is reduced, so the overall bottleneck in backhaul link is minimized and area spectral efficiency is improved. A portion of multimedia data show significant growth in data traffic, such as sports matches and movies. To handle such a huge amount of multimedia data, caching plays a necessary role by caching popular data files near the end-users that can significantly reduce the burden on backhaul channels and minimize the latency for a large number of end-users. To this end, two caching schemes were proposed, namely coded caching and un-coded caching.

In coded caching, the data items are cached in terms of parity bits at different locations using fountain code, while, in un-coded caching, the whole data item or object is cached. In an un-coded caching strategy, the frequently accessed data items are cached at each RU until full cache space. All the RUs behave cooperatively to serve the incoming users' requests using the backhaul links [110]. These strategies are developed to improve the joint design of dynamic clustering, multicast beamforming, and backhaul traffic balancing. The dynamic clustering and beamforming are cooperatively enhanced to reduce power consumption. However, the traffic on each backhaul channel is optimized according to their link capacity.

6.10. Caching to mitigate disasters

Data dissemination over the Mobile Social Networks (MSNs) is considered as a promising approach for mobile users to share their information. Recently, research shows that the number of mobile devices already exceeds the number of people on earth [113]. The immense growth of mobile users has produced an exponential increment in mobile data traffic. Consequently, it needs the extra capacity of backhaul links to disseminate such a huge amount of mobile traffic [111].

Now the MSNs face data delivery related issues in which delay is critical regarding mobile users' perspective. Therefore, addressing these issues is important to deliver large amounts of mobile data in the core network. To cope with this, fog-based caching plays an imperative role in enhancing the overall data transmission over the MSNs. Indeed, caching can efficiently deliver diverse data items by minimizing the huge amount of redundant transmission. As a result, the traffic burden and usage of backhaul links are reduced. Moreover, it minimizes the latency to deliver data items in MSNs. Caching can contribute to MSNs in multiple ways. The popular data items are cached at the network edges near the mobile users that mitigate the delay in transmission and improve the Quality of Experience (QoE) [114]. Moreover, caching reduces the duplicate transmission over the backhaul in core MSNs.

According to disaster backup, caching plays a significant role in minimizing data loss. For instance, popular and critical data items will be lost if a network node's operating system stops working or when malicious users attacks spread the virus. Therefore, fog caching is a promising approach to alleviate disaster-related problems. Caching also plays an important role in natural disasters, such as earthquakes and fires, where fires can cause network nodes to lose power and potential for losing important data is high. In these situations, the popular and important data items are cached at multiple nodes as the backup to use during a node's failure [112].

Secure Caching Scheme (SCS) is proposed by Su et al. [58] to enhance the MSNs in fog computing and minimize the data loss during disasters. In this scheme, data is encrypted with partitioning and scrambling methods to improve privacy, and these data items are delivered to multiple locations to be cached. The proposed caching scheme outperforms in terms of improving resource efficiency and security.

6.11. Data distribution

In fog computing, data processing and storage are executed at the central unit. However, data storing at the central cloud is not suitable because it may cause a delay in response, high storage cost, and congestion on network channels due to high data traffic between cloud and fog nodes. At the fog layer, the data distribution depends on the geographic producer's location and end-users. However, data storage consists of different factors such as distribution, dissemination, and replication. In large networks, where many devices are interconnected, data dissemination techniques need to distribute data to all nodes fairly. However, caching is considered as a promising approach to fairly distributing data among the nodes in fog computing. Thus, caching schemes are used to handle a large amounts of information in a fog-based caching system that sends data items to different network nodes and caches them close to the end-users. As a result, delay and latency in data propagation is reduced. It also improves the availability of desired data.

Recently, Most Popular Cache (MPC) [112] and Capacity Aware Edge Caching (CAEC) [61] schemes are developed to enhance the data distribution in fog computing. The MPC only caches the most frequently requested data items to avoid load on a network node and fog storage capabilities. Moreover, it minimizes the delay and improves the packet delivery ratio in a fog-based caching network. Besides, it increases the data hit ratio and minimize caching operations. However, CAEC improves the data delivery and cache usage in fog networks.

In CAEC, the optimal caching place is determined to store the data items. In this way, the data hit ratio is maximized, and the average downloaded time is reduced. Moreover, it improves traffic allocation balance and enhances the connectivity and cache capacity in fog networks. Table 5 shows the present challenges of fog computing and their corresponding caching schemes as solutions with basic goals and advantages.

7. Fog caching in machine learning prospective

Cache module plays a significant role in improving the fog computing networks' performance. Due to the continuous development and growth of fog computing at a large scale network, cache management becomes a hurdle in improving the overall network performance. Moreover, cache

Table 5
Fog computing challenges and caching schemes with their aims and advantages.

Aspects	Challenges	Techniques	Aim/Goal	Advantages
Offloading [67]	Limited resources Scaling in offloading Inadequacy of standard platform Allocation of computing resources	GA based task caching [75] CCP [76]	This caching is used to enhance the offloading performance of fog computing. CCP is proposed to enhance the offloading capabilities by measuring three primitives: task popularity, input, and output sizes	Improve the utility of the system Reduce the uplink traffic, computation and communication cost CPPReduce offloading latencies
Scheduling [80]	Latency Heterogeneity Execution time Resource scarcity Power consumption Resource allocation Geographic restrictions Internal processing time	CBA [86]	This technique integrates caching and the concept of a smart gateway to solve the resource re-allocation problem in fog computing by providing a CBA scheduling algorithm. CBA incorporates the cache module into a smart gateway to store the job and information	Minimize latency Increase the overall Scheduling performanceI mprove execution time Reduce power consumption Enhance internal processing delay
Energy and Power Saving [86]	Network size Data delivery Task offloading Energy efficiency Energy consumption Transmitting power Mobile applications	PC [62] DCC [59]	PC provides load balancing to improve the system efficiency and filtration to cache popular content on active nodes. DCC provides caching space to perform a task in caching-based fog computing	PC consume 92% less energy than they would without caching networkDCC reduces the overall energy consumptionDCC reduces the computation burden
Bandwidth and Cost Reduction [90]	Bottleneck Congesting Inter-domain cost Bandwidth limitations Content retrieval cost Operational cost for ISP	SAEC [92] CoA [93]	SAEC is proposed to minimize the bandwidth consumption in fog computing. CoA reduces the cost for the benefit of ISPs. It considers the operational cost to reduce the load on network channels	Reduce storage costs Reduce average delay Reduce the link load Minimize operational cost for ISPs Minimize bandwidth cost and consumption
Data Availability [94]	Throughput Data traffic Service latency Remote execution Limited amount of computing resources	ECM [64]	In ECM, the users' interest is measured proactively and forms a group like clusters of the users with similar interest and the cluster is mapped within fog node	Reduces the average latency Cache the popular files effectively Improve the hit rate and network throughput
Throughput [98]	Unprecedented traffic Congestion at off-peak periods Limited capacity of front and backhaul Redundant data traffic and network overhead	PC [99]	PC is proposed to improve fog caching by selecting the frequently requested content through popularity prediction, and these content are locally cached near the end-users in real-time	Enhance the overall network throughput Increase the overall data hit performance Subsequent requests are accomplished in less time
Latency and Stretch [100]	Delay in successful transmission Limited capacities of back and front-haul links Ultra-low latency dissemination and computation	PCS [100] OCC [103]	In PCS, tasks are categorized based on the request frequency. Popular tasks cached using cloudlet. OCC optimizes the input intermediate computational results on the arrival of users' operations	PCS reduce the costPCS minimize the delayOCC minimizes Transmission latencyPCS and OCC reduce the computational latency
Fronthaul link load [105]	Heavy burden Spectral efficiency Duplicate transmission Huge volume of data traffic Incredible growth of mobile applications and smart devices	PC [105] CA [106]	In PC, popular data items are cached to reduce the dissemination rate under the capacity of fronthaul. In CA, the analysis is established to find the optimal caching techniques to improve the performance of fronthaul	PC reduces the dissemination rate Caching enhances the delivery delay Improve the performance of fronthaul.
Backhaul traffic load [108]	Load on backhaul links Content retrieval latency The bottleneck in backhaul links Spectral efficiency in data channels Congested Backhaul links between RUs and CP	Coded, un-coded caching [110]	In coded caching, the data items are cached in terms of parity bits at different locations using fountain code while, in un-coded caching, the whole object is cached each RU until the cache space is full. All the RUs behave cooperatively to serve the incoming requests	Power consumptionD ownlink efficiency On-demand data broadcasting Improve dynamic clustering and beamforming Enhance backhaul channel according to link capacity
Caching in Disaster [111]	Delay Malicious users attack Data loss during system failure Earthquake and fire can make power failure	SCS [58]	SCS minimizes data loss during disasters which improves privacy by encrypting the contents using partitioning and scrambling methods, and these contents are cached at multiple locations	Improve privacy Enhance security Reduce the data loss Maximize data delivery Improve resource efficiency
DataDistribution [61]	Delay in response Fairly distribution of data High storage cost due to high Data traffic The distribution of data stored at the fog layer depends on the geographic producer's location	MPC [112] CAEC [61]	MPC caches the most frequently requested data items. In CAEC, the optimal caching location is determined to store the data items. It improves the balance of traffic allocation, and enhances the connectivity and cache capacity	MPC minimizes delayMPC improves packet deliveryCAEC improves the data deliveryCAEC reduces the average downloaded timeCAEC enhances the connectivity and cache capacity

placement is a significant problem in fog-based IoT networks to distribute highly requested contents to end-users efficiently. We mention in earlier sections the caching plays an imperative role in improving the performance of fog networks [115]. However, it is facing several critical problems due to the limited cache size. Fog computing is a novel approach to bring the resources, data, and services at the edges of the network near the end consumers to reduce transmission delay and energy consumption [116]. However, it is challenging to perform caching and processing operations for all incoming data items due to the limited size of available cache at fog nodes. Therefore, it is most important to know about the exact data for caching prior [117,118].

Data caching within the fog nodes is dependent on multiple factors such as network topology, change locations, and the varying nature of different IoT-based end users. Thus, subsequent requests for different data items are highly unidentified before taking caching decisions [119]. To this end, Machine Learning (ML) proposes flexible techniques to provide a centralized location to process raw data. Besides, ML-based schemes facilitate and maximize each fog node's performance to make the right decisions by caching the right data items. In addition, ML is the most suitable approach to predict the user demands and map the users' inputs with the outputs actions [120]. Moreover, it is used to improve the overall caching performance of a network by identifying the end-users requirements to discover early information from a large number of content streams. Furthermore, in ML, a large number of contents are exploited to identify the popularity of a data item, is a suitable approach to filter the data and information [121]. As a result, the subsequent processing is easy to analyze the connection between the features corresponding outputs of the data [122]. ML techniques are categorized into supervised learning (SL) and Un-supervised Learning (UL).

In SL, the system provides learning-based algorithms with known quantities that help in making future decisions. However, in UL, the system provides algorithms with unlabeled data to make decisions without any prerequisite information or guidance. ML can emerge with fog computing at the terminal, fog, or cloud layer. ML is used for data sensing at the terminal layer, and diverse schemes and methods are available for data sensing. The complex features of datasets such as vibrations, videos, and model reading from the IoT-based devices can be identified using the ML methods like Conventional Neural Network (CNN) [123]. At the fog layer, ML is responsible for resource management and data caching (storage). For this purpose, ML-based algorithms are used to sample and compressed the data received from the IoT devices and aggregated the compressed data at fog nodes for additional processing. Therefore, to address the different challenges of fog computing, many ML-based caching techniques are developed to enhance fog networks. For example, in Refs. [124,125], a ML-based caching scheme named online proactive caching is developed to predict time-series requests for contents and updated the network edge caching. In this scheme, Bidirectional Deep-recurrent Neural Network (BRNN) and convolution neural network models are used to predict content popularity and reduce computational costs. Later, a fully connected neural network is integrated to predict and learn samples from BRNN. The experiment results show that proposed scheme improves prediction accuracy and maximizes the data hit rate for end devices.

In a study by Lan et al. [126], an intelligent computation offloading technique with caching to improve the offloading in fog computing is proposed. It reduces long-term energy consumption and task processing time. In this scheme, a Deep Reinforcement Learning (DRL) based algorithm is proposed to solve complex optimization problems. Thus, the proposed technique performs better in terms of energy consumption and computation latency. The following sections describe diverse ML-based techniques that improve cache management in fog computing. The techniques like clustering of fog servers are explained in the following sections.

7.1. *Clustering-based fog approach*

Clustering works under UL, where the information is not guided. In this approach, the fog-based nodes are combined into clusters to satisfy the demands of IoT-based end devices [127]. Data items are cached at different fog nodes after being coded into segments. Whenever a request is received from the user, it sends to the fog clusters and is satisfied based on cached data items. The user of IoT devices can fetch the data items if the data items are cached at clusters of fog nodes; otherwise, the requests forward to the cloud server to download the required data items. Moreover, the caching performance changes with the size of clusters, which means that if the cluster includes more fog nodes, the user can get data from the nearest nodes. Therefore, cache diversity is improved, and efficient caching performance can be achieved. Thus, to balance the trade-off, the size of the cluster should be optimal.

7.2. *Similarity-based learning approach*

In the Similarity-based Learning Approach (SLA), the fog nodes are combined with a pair of similar IoT-based devices and less similar IoT-based devices. In SLA, the intelligent fog node is used to determine the similarity function, or distance function among the given set of similar IoT-based devices through learning about the different features of devices [19]. Besides, parameters like physical location (link quality) and common interest are measured to identify the similarity between IoT-based devices. With the help of a one-to-one function, the intelligent node determines the new devices are similar or not to meet the future demands of new devices.

7.3. *Transfer-based learning approach*

In the Transfer-based Learning Approach (TLA), the intelligent node has pre-knowledge to solve the problems and cache the data items [128]. In TLA, fog nodes use the source domain and target domain to complete the given task. The source domain is responsible for getting the information from the interaction of IoT-based devices. In contrast, the target domain is responsible for following the requests pattern made by IoT devices. The source domain is the combination of IoT devices and the data that was requested earlier. Moreover, it knows the popularity matrix, where the popularity of each content is measured by taking the sum of received requests for content. The TLA gets the user-data relationship from the source domain to create a link between the source domain and target domain to find similar content. Therefore, both source domain and target domain are combined to determine the popularity matrix [131]. Thus, the content showing a higher popularity matrix is cached at the fog nodes near the IoT devices.

7.4. *Recommendation-based Q learning approach*

In local caching-based systems, the end-users do not have any information about the cached data items, and they do not know where to send the requests to fetch the desired data items. Therefore, Recommendation-based Q Learning (RQL) algorithms are most suitable to enhance system efficiency [129]. To improve caching efficiency of the system, fog nodes broadcast messages to the end-users to provide information about the cached data. The broadcast message introduces content and its rank about popularity value (how many requests the content has received). The value of the message influences the decision making by the user to send requests for fetching the content even though the requesting rate for content, arrival, and departure rate of end-devices are unknown. A Q-based learning scheme is considered the appropriate approach to enhance the system performance by minimizing delay and improving the system throughput. Thus, it shows promising accuracy to determine the subsequent requirements of the fog nodes by considering the Q values. The ratio for an i-th content depends on the number of IoT-based end devices connected with the system and the number of devices connected within a particular period. Consequently, the unknown requests for i-th content depend on the caching action within the previous and current time interval. Therefore, the Q value is assigned to the state action pair in learning schemes, increasing the reward. The number of IoT-based devices is selected for taking action within a particular period. Thus, the reward for the subsequent action is determined by observing the previous actions, and a new learning value is measured [129]. RQL maximizes the long-term reward for a system to improve the overall caching performance.

7.5. Deep reinforcement-based learning approach

This approach intelligently observes the environment automatically to learn about the caching strategy regarding the history [132,133]. The integration of deep neural networks is a promising approach to learn about raw and high-dimensional data automatically. However, learning-based caching strategies are divided into Popularity Prediction-based Caching Strategies (PPCS) and Reinforcement Learning-based Caching Strategies (RLCS). In the PPCS, the popularity of content is measured and then caching policy is developed based on the popularity prediction [119]. Several parameters like context information, user content relation, and traffic patterns are selected to determine (predict) the popularity of a content [134,135]. The RECS emerges the content popularity and content placement to work as a single entity. In this approach, Reinforcement Learning (RL) agents are trained on raw and high dimensional observations. Therefore, the fog nodes observe the environment to obtain the state of the environment, and then, according to the caching policy, the corresponding state is obtained [136]. The basic goal of deep reinforcement learning is to maximize the reward during the agent taking actions at a particular state.

7.6. Federated-based learning approach

The conventional ML techniques depend on the central entity for the data processing and caching. However, it is not possible to fetch the data from a private server. It increases the communication overhead to disseminate the data from a large number of IoT devices to the central ML processors [130]. To cope with this, Federated-based Learning (FL) provides decentralized-based ML techniques to keep the data at the origin (where the data was generated), and locally trained models are disseminated to the central processor. These techniques significantly reduce network bandwidth and energy consumption by disseminating the features rather than the entire data stream. Moreover, FL-based approaches also reduce the delay in real-time responses [137]. In addition, these approaches increase the processing power of the devices. They use private data to perform model training in a distributed manner to keep data at the origin (place of data generation). Furthermore, the content popularity prediction can be computed with the help of FL approaches [138]. Table 6 summarizes the goals and benefits of ML-based techniques that are the most promising approaches to enhance the overall caching performance of fog networks.

8. Cache security risks and insight

The fog system is still facing security-related problems. To optimize web services, the fog computing has some security related issues, for example, applications are vulnerable to the code injection attacks if user input is not properly valid. In SQL injection, for example, user-supplied SQL code is automatically executed to help unauthorized users access and modify data. Consequently, the whole fog system will compromise and forward the unauthorized (modified) data to the central server. Likewise, insecure Application Programming Interfaces (APIs) attacks such as cookie hijacking and sessions are insecure direct object references for unauthorized data, malicious redirections, drive-by attacks, and illegal information can force fog computing to expose the system and its authorized users. Moreover, web attacks also target the other applications within the same fog system by integrating the scripts and damage sensitive information. Furthermore, cache-based side-channel attacks are also challenging security risks for the cache management module in the fog system. For example, exposing the cryptographic hash functions or keys can result in the leaking of sensitive information. In practical implementation, preventing cache-based attacks is expensive. Recent studies show that the cache-interferences are challenging for both software and hardware modifications.

From the present study, it is clear that fog-based networks face several issues such as offloading, scheduling, energy consumption, the high burden on front-haul, backhaul links, high bandwidth cost, latency, data availability, and data distribution. To improve the performance of fog-based networks, caching seems like a promising method to cope with the fog paradigm's existing issues. Caching has several benefits over traditional fog-based networks. Most researchers show their interest in

Table 6
Machine learning techniques with aim/goal and benefits.

Techniques	Reference	Aim/Goal	Benefits
Clustering-based Fog Approach [127]	Alghamdi et al. [127]	In this approach, fog-based nodes are combined into clusters to satisfy the demands of end devices. Data items are cached at diverse fog nodes after being coded into segments to achieve better performance	Provide coded-based caching Provide load balancing Improves cache diversity mechanisms to improve system efficiency
Similarity-based Learning Approach [19]	Xu et al. [19]	In SLA, the intelligent fog node is used to determine the similarity function or distance function among the given set of similar devices by learning their features	Use only previous knowledge to find the similarity Efficient if similar device arrives
Transfer-based Learning Approach [128]	Li et al. [128]	In TLA, the intelligent node has pre-knowledge to solve the problems. It caches the data items using the popularity matrix, and fog nodes use the source domain and target domain to complete the given task	Provide a popularity matrix to cache the content efficiently and solve the issues by manipulating existing knowledge
Recommendation-based Q Learning Approach [129]	Guo et al. [129]	In local caching-based systems, the end-users do not have any information about the cached data items, and they do not know where to send the requests to fetch the desired data items. Therefore, RQL algorithms are most suitable to enhance system efficiency. RQL maximizes the long-term reward and performance	In this approach, the end-devices can identify the data to be cached to which node using data priorities. RQL maximizes the long-term reward for a system to improve the overall caching performance
Deep Reinforcement-based Learning Approach [119]	Zhu et al. [119]	In PPCS, the popularity of a content is measured and caches the content. The basic goal of deep reinforcement learning is to maximize the reward during the action taking by an agent at a particular state	Reduce the long-term cost of downloading content. Overlapping coverage of fog nodes is considered
Federated-based Learning Approach [130]	Li et al. [130]	The conventional ML techniques depend on the central entity for the data processing and caching. Furthermore, the content popularity prediction can be computed with the help of FL approaches	Use the private contents to minimize communication overhead, bandwidth, and energy consumption

exploring its capabilities that will enhance the fog computing infrastructure. It is one of the most suitable techniques to improve data distribution and dissemination in fog-based environments. Thereupon, a number of flexible caching schemes are developed to enhance the different fog environments. Most of these caching schemes are appropriate to meet fog computing requirements. For instance, caching can reduce the huge amount of duplicate transmission.

The massive growth in data traffic and multiple dissemination of similar content continues to pose a number of challenges, the most important of which are inefficient resources utilization, redundant transmission, network congestion, energy, bandwidth, and cost consumption, and which are not easily solved by the current fog-based caching technologies. Moreover, cache management becomes a critical issue because of its limited capacity to handle or accommodate the huge amount of transmitted data. Consequently, it needs to enhance the fog-based caching infrastructure regarding user perspectives. For example, the end-users are keenly interested in downloading their desired data within short latency. Moreover, cache placement is a significant problem in fog-based IoT networks to distribute highly requested contents to end-users efficiently. Besides, fog computing has limited capacity and resources (cache) to efficiently allocate resources to the end-users. Efficient content caching is the core part of fog computing. Therefore, low-quality-based caching schemes can increase the burden on the network and consumes more resources. Moreover, the caching schemes should be compatible with IoT-based applications [16]. ML is considered as a promising solution to cope with these challenges, and recently, it has gained significant attention from the research community. ML uses stochastic gradient descent that can determine the optimal solution for complex problems [139].

ML-based data handling and caching techniques play an imperative role in providing security and popular data items with less delay and caching these data items near the end-users. Hence, the subsequent requests can fetch these cached data items within a short time. ML provides diverse techniques to handle the data by managing the cache efficiently, as described in previous sections.

9. Future research directions

Regardless of the promising view of cached-based fog computing, several issues still needed to be addressed, as given in the following.

9.1. Energy management

As the fog networks consist of distributed nodes that consume more energy compared with the cloud counterparts. Thus, it needs much effort to propose and optimize an efficient energy-saving caching scheme for fog networks. For instance, an optimal caching scheme needs to be designed to use resources to minimize energy consumption efficiently.

9.2. Strict latency

It is one of the significant issues of fog and cache-based fog computing paradigms. Industrial systems such as manufacturing, goods packing, gas, and oil systems generally measure the end-to-end latency within a few milliseconds. However, potential applications, such as drone flight control, vehicle to roadside communication, and virtual reality, need tens of milliseconds. In such systems and applications, strict latency is required to accomplish the tasks. It creates a issue for cached-based fog computing.

9.3. High demands of network bandwidth

The number of mobile devices is growing very fast and generates a gigantic amount of data traffic. For instance, currently, the generated data from an autonomous vehicle is predicted as one gigabyte per second. The Google data traffic is estimated as one petabyte per month. To transmit such a massive amount of data (cloud to fog nodes), computing architecture needs more network bandwidth and extensive cost. Therefore, the integrated cache-based fog network will be required to enhance fog infrastructure for efficient data processing and storage to provide data dissemination with efficient bandwidth consumption.

9.4. Irregular connectivity

Currently, a large number of devices are connected with the Internet through wireless channels, and some devices may face from the fluctuating wireless signals and suffer from irregular connectivity to fog nodes. For instance, moving devices usually suffer from such kinds of issues in which drones, vehicles, or a mobile terminal of a cellular system. However, communication services such as data gathering, analytics, and control are highly recommended. Therefore, fog computing architecture is highly desirable to design an efficient caching scheme that gives consistent service availabilities under fluctuating and irregular connectivity conditions to serve urgent demands.

9.5. Mobility

A number of content providers, computing nodes, and caching nodes can be mobile devices. It is very hard to find an appropriate computing and caching node in a network in these circumstances. Route failure and data requests failure depend on the mobility of a network node. Thus, the overall network performance is reduced during irregular mobility. Consequently, it is essential to address the mobility-related issues by deploying an efficient caching scheme having mobility features to improve the fog network effectiveness.

9.6. Security and privacy

Cloud servers offer obvious protection and are more secure against security threats compared to fog nodes. However, in fog computing, the caching facilities and services are provided in a distributed manner. Therefore, such distributed systems usually have fewer safety features and are vulnerable to attacks. As a result, cache-based fog computing architecture is facing security-related issues. Likewise, cached-based computing has limited resources (limited sizes) to identify the threats and protect the distributed system from attacks. Thus, there is a need to integrate security features within caching schemes to secure communication and data caching.

9.7. Network consistency

Most caching and computing systems are generally organized in distributed manners that may cause oscillation, divergence, and inconsistency for the global computing network. For instance, this may happen when an un-organized mobile system is combined with a virtual pool of unpredictable shared resources. Consistency is considered as a typical issue of distributed networks, and several distributed systems such as analytics and stream mining require extra demand to address this issue. Therefore, it can handle by providing consistent data dissemination through using appropriate caching and computing resources.

9.8. Fog computing in machine learning perspective

Despite the broad usage of robots services and IoT-based industrial applications, real-time applications, there are still some problems with real-time applications, such as low efficiency in completing services and tasks. Therefore, providing a collaborative environment for the real-time processing of a smart and industrial application requires emerging ML-based intelligence for fog-based caching systems that can respond quickly and decide on real-time processes.

9.9. Utilization of resources

Fog computing offers an efficient platform for diversified technologies to provide several services to the IoT-based end-devices and users. However, the connection to and use of resources is huge challenge that needs to be addressed effectively. Besides, it is imperative to develop an efficient caching technique for task scheduling to utilize resources properly.

9.10. Latency management

It is crucial to consider the latency to ensure the efficient level of QoS in fog computing scenarios. The research on fog-based caching is still in its early stage regarding the latency for service delivery to ensure efficient QoS in the whole system.

9.11. Lack of cache space

To deploy an ML-based system, it is important to have adequate data within the learning system to complete the learning process. However, in fog computing, the nodes have limited cache capacity and do not ensure the learning process to complete efficiently. Therefore, the selection of appropriate data during the learning process by developing an effective ML-based caching technique is an important issue that needs to be addressed.

10. Conclusion

Cache-based fog computing is a promising high potential communication architecture that is broadly acceptable due to the considerable development of mobile Internet and IoT. Fog computing provides extra services and caching space at the network edges to efficiently use edge devices. It significantly reduces transmission latency and energy consumption. It can efficiently meet the requirements of latency-sensitive and real-time applications. This survey presents a comprehensive overview of fog computing and its relation to IoT-based environments. A summary of existing surveys is described with their contributions and limitations. In addition, caching in fog computing, caching mechanisms, and its contributions regarding different perspectives to enhance fog networks' overall performance are thoroughly explained. Moreover, an acute survey on the challenges of fog computing is also presented. The ML-based caching techniques are summarized to determine how to support the cache module in fog infrastructure. Therefore, ML cache-able fog nodes can serve more effectively to improve the overall system performance. In the end, open issues and future research directions like strict latency, intermittent connectivity, mobility, security, consistency are presented.

Declaration of competing interest

The authors declare no conflict of interest.

Acknowledgements

Provincial key platforms and major scientific research projects of universities in Guangdong Province, Peoples R China under Grant No. 2017GXJK116.

References

[1] J.A. Cunningham, J. Whalley, Internet of things: promises and complexities, in: The Internet of Things Entrepreneurial Ecosystems, Springer, 2020, pp. 1–11.

[2] F. Tao, Y. Wang, Y. Zuo, H. Yang, M. Zhang, Internet of things in product life-cycle energy management, J. Indust. Info. Intergrate. 1 (2016) 26–39.

[3] S.A. Chaudhry, A. Irshad, J. Nebhen, A.K. Bashir, N. Moustafa, Y.D. Al-Otaibi, Y.B. Zikria, An anonymous device to device access control based on secure certificate for internet of medical things systems, Sustain. Cities Soc. 75 (2021) 103322.

[4] S.A. Chaudhry, A. Irshad, M.A. Khan, S.A. Khan, S. Nosheen, A.A. AlZubi, Y.B. Zikria, A lightweight authentication scheme for 6g-IoT enabled maritime transport system, IEEE Trans. Intell. Transport. Syst. (2021) 1–10.

[5] M.W. Akram, A.K. Bashir, S. Shamshad, M.A. Saleem, A.A. AlZubi, S.A. Chaudhry, B.A. Alzahrani, Y.B. Zikria, A secure and lightweight drones-access protocol for smart city surveillance, IEEE Trans. Intell. Transport. Syst. (2021) 1–10.

[6] I. Lee, K. Lee, The internet of things (IoT): applications, investments, and challenges for enterprises, Bus. Horiz. 58 (4) (2015) 431–440.

[7] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, A. Wahab, A review of intrusion detection systems using machine and deep learning in internet of things: challenges, solutions and future directions, Electronics 9 (7) (2020) 1177.

[8] S. Sankaranarayanan, J.J. Rodrigues, V. Sugumaran, S. Kozlov, et al., Data flow and distributed deep neural network based low latency IoT-edge computation model for big data environment, Eng. Appl. Artif. Intell. 94 (2020) 103785.

[9] J. Ren, D. Zhang, S. He, Y. Zhang, T. Li, A survey on end-edge-cloud orchestrated network computing paradigms: transparent computing, mobile edge computing, fog computing, and cloudlet, ACM Comput. Surv. 52 (6) (2019) 1–36.

[10] F.B. Alhasawi, J.V. Milanovic, Techno-economic contribution of facts devices to the operation of power systems with high level of wind power integration, IEEE Trans. Power Syst. 27 (3) (2012) 1414–1421.

[11] J. Moura, D. Hutchison, Fog computing systems: state of the art, research issues and future trends, with a focus on resilience, J. Netw. Comput. Appl. (2020) 102784.

[12] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A.K. Bashir, A. Jolfaei, N. Kumar, Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications, IEEE Trans. Ind. Inf. 17 (7) (2020) 5068–5076.

[13] Y. Li, H. Ma, L. Wang, S. Mao, G. Wang, Optimized content caching and user association for edge computing in densely deployed heterogeneous networks, IEEE Trans. Mobile Comput., 21 (6) (2022) 2130 - 2142.

[14] C. Mouradian, D. Naboulsi, S. Yangui, R.H. Glitho, M.J. Morrow, P.A. Polakos, A comprehensive survey on fog computing: state-of-the-art and research challenges, IEEE Commun. Survey Tutorial. 20 (1) (2017) 416–464.

[15] Y. Meng, M.A. Naeem, M. Sohail, A.K. Bashir, R. Ali, Y.B. Zikria, Elastic caching solutions for content dissemination services of ip-based internet technologies prospective, Multimed. Tool. Appl. 80 (11) (2021) 16997–17022.

[16] W.A. Almobaideen, O.M. Malkawi, Application based caching in fog computing to improve quality of service, in: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), IEEE, 2020, pp. 20–27.

[17] S. Khan, S. Parkinson, Y. Qin, Fog computing security: a review of current applications and security solutions, J. Cloud Comput. 6 (1) (2017) 1–22.

[18] E. Di Pascale, H. Ahmadi, L. Doyle, I. Macaluso, Toward scalable user-deployed ultra-dense networks: blockchain-enabled small cells as a service, IEEE Commun. Mag. 58 (8) (2020) 82–88.

[19] Z. Chang, L. Lei, Z. Zhou, S. Mao, T. Ristaniemi, Learn to cache: machine learning for network edge caching in the big data era, IEEE Wireless Commun. 25 (3) (2018) 28–35.

[20] K. Ujjwal, S. Garg, J. Hilton, J. Aryal, N. Forbes-Smith, Cloud computing in natural hazard modeling systems: current research trends and future directions, Int. J. Disaster Risk Reduc. 38 (2019) 101188.

[21] D.C. Nguyen, P.N. Pathirana, M. Ding, A. Seneviratne, Integration of blockchain and cloud of things: architecture, applications and challenges, IEEE Commun. Survey Tutorial. 22 (4) (2020) 2521–2549.

[22] R. Casadei, M. Viroli, G. Audrito, D. Pianini, F. Damiani, Engineering collective intelligence at the edge with aggregate processes, Eng. Appl. Artif. Intell. 97 (2021) 104081.

[23] H.F. Atlam, R.J. Walters, G.B. Wills, Fog computing and the internet of things: a review, Big Data Cognitive Comput. 2 (2) (2018) 10.

[24] S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in: Proceedings of the 2015 Workshop on Mobile Big Data, 2015, pp. 37–42.

[25] S. Kitanov, E. Monteiro, T. Janevski, 5g and the fog—survey of related technologies and research directions, in: 2016 18th Mediterranean Electrotechnical Conference (MELECON), IEEE, 2016, pp. 1–6.

[26] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42.

[27] M. Mukherjee, L. Shu, D. Wang, Survey of fog computing: fundamental, network applications, and research challenges, IEEE Commun. Survey Tutorial. 20 (3) (2018) 1826–1857.

[28] P. Bellavista, J. Berrocal, A. Corradi, S.K. Das, L. Foschini, A. Zanni, A survey on fog computing for the internet of things, Pervasive Mob. Comput. 52 (2019) 71–99.

[29] I. Martinez, A.S. Hafid, A. Jarray, Design, resource management and evaluation of fog computing systems: a survey, IEEE Internet Things J. 8 (4) (2021) 2494–2516.

[30] M.S.U. Islam, A. Kumar, Y.-C. Hu, Context-aware scheduling in fog computing: a survey, taxonomy, challenges and future directions, J. Netw. Comput. Appl. (2021) 103008.

[31] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, E. Alanazi, Applying machine learning techniques for caching in next-generation edge networks: a comprehensive survey, J. Netw. Comput. Appl. (2021) 103005.

[32] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: architecture, enabling technologies, security and privacy, and applications, IEEE Internet Things J. 4 (5) (2017) 1125–1142.

[33] M.A. Naeem, R. Ali, B.-S. Kim, S.A. Nor, S. Hassan, A periodic caching strategy solution for the smart city in information-centric internet of things, Sustainability 10 (7) (2018) 2550–2576.

[34] Y. Wu, Cloud-Edge Orchestration for the Internet of Things: Architecture and AI-Powered Data Processing, IEEE Internet Things J., vol. 8, no. 16, pp. 12792-12805, 15 Aug, 2021.

[35] Y. Li, S. Xia, m. zheng, b. cao, q. liu, Lyapunov optimization based trade-off policy for mobile cloud offloading in heterogeneous wireless networks, IEEE Trans. Cloud Comput. (2019) 1, 1.

[36] Y. Li, J. Liu, B. Cao, C. Wang, Joint optimization of radio and virtual machine resources with uncertain user demands in mobile cloud computing, IEEE Trans. Multimed. 20 (9) (2018) 2427–2438.

[37] A. Heidari, M.A. Jabraeil Jamali, N. Jafari Navimipour, S. Akbarpour, Internet of things offloading: ongoing issues, opportunities, and future challenges, Int. J. Commun. Syst. 33 (14) (2020) e4474.

[38] A. Agasthian, R. Pamula, L. Kumaraswamidhas, Defense model for preserving the wind turbine records in cloud using fog computing with coupling based cryptography, Peer to Peer Network Appl. 13 (6) (2020) 2155–2165.

[39] N. Piovesan, A.F. Gambin, M. Miozzo, M. Rossi, P. Dini, Energy sustainable paradigms and methods for future mobile networks: a survey, Comput. Commun. 119 (2018) 101–117.

[40] Z. Ali, S.A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, Y.B. Zikria, A clogging resistant secure authentication scheme for fog computing services, Comput. Network. 185 (2021) 107731.

[41] M.N. Khan, A. Rao, S. Camtepe, Lightweight cryptographic protocols for IoT constrained devices: a survey, IEEE Internet Things J. 8 (6) (2020) 4132–4156.

[42] T. Goethals, F. De Turck, B. Volckaert, Near real-time optimization of fog service placement for responsive edge computing, J. Cloud Comput. 9 (1) (2020) 1–17.

[43] S. Shekhar, A. Chhokra, H. Sun, A. Gokhale, A. Dubey, X. Koutsoukos, G. Karsai, Urmila: dynamically trading-off fog and edge resources for performance and mobility-aware IoT services, J. Syst. Architect. 107 (2020) 101710.

[44] M.A. Naeem, T.N. Nguyen, R. Ali, K. Cengiz, Y. Meng, T. Khurshaid, Hybrid cache management in IoT-based named data networking, IEEE Internet Things J. 9 (1) (2021) 7140–7150.

[45] S. Madakam, V. Lake, V. Lake, V. Lake, et al., Internet of things (IoT): a literature review, J. Comput. Commun. 3 (5) (2015) 164.

[46] J. Hou, L. Qu, W. Shi, A survey on internet of things security from data perspectives, Comput. Network. 148 (2019) 295–306.

[47] J. Ni, K. Zhang, X. Lin, X. Shen, Securing fog computing for internet of things applications: challenges and solutions, IEEE Commun. Survey Tutorial. 20 (1) (2017) 601–628.

[48] M. Aazam, S. Zeadally, K.A. Harras, Fog computing architecture, evaluation, and future research directions, IEEE Commun. Mag. 56 (5) (2018) 46–52.

[49] S. Sarkar, S. Misra, Theoretical modelling of fog computing: a green computing paradigm to support IoT applications, IET Netw. 5 (2) (2016) 23–29.

[50] M.A. Naeem, R. Ali, M. Alazab, Y. Meng, Y.B. Zikria, Enabling the content dissemination through caching in the state-of-the-art sustainable information and communication technologies, Sustain. Cities Soc. 61 (2020) 102291.

[51] Y. Meng, M.A. Naeem, A.O. Almagrabi, R. Ali, H.S. Kim, Advancing the state of the fog computing to enable 5g network technologies, Sensors 20 (6) (2020) 1754.

[52] T.N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, Fog computing in healthcare internet of things: a case study on ecg feature extraction, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 356–363.

[53] S. Sarkar, S. Chatterjee, S. Misra, Assessment of the suitability of fog computing in the context of internet of things, IEEE Trans. Cloud Comput. 6 (1) (2015) 46–59.

[54] E. Balevi, R.D. Gitlin, Optimizing the number of fog nodes for cloud-fog-thing networks, IEEE Access 6 (2018) 11173–11183.

[55] J. Su, F. Lin, X. Zhou, X. Lu, Steiner tree based optimal resource caching scheme in fog computing, China Commun. 12 (8) (2015) 161–168.

[56] S. Wang, X. Huang, Y. Liu, R. Yu, Cachinmobile: an energy-efficient users caching scheme for fog computing, in: 2016 IEEE/CIC International Conference on Communications in China (ICCC), IEEE, 2016, pp. 1–6.

[57] F. Song, Z.-Y. Ai, J.-J. Li, G. Pau, M. Collotta, I. You, H.-K. Zhang, Smart collaborative caching for information-centric IoT in fog computing, Sensors 17 (11) (2017) 2512.

[58] Z. Su, Q. Xu, J. Luo, H. Pu, Y. Peng, R. Lu, A secure content caching scheme for disaster backup in fog computing enabled mobile social networks, IEEE Trans. Ind. Inf. 14 (10) (2018) 4579–4589.

[59] H. Xing, J. Cui, Y. Deng, A. Nallanathan, Energy-efficient proactive caching for fog computing with correlated task arrivals, in: 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2019, pp. 1–5.

[60] Y. Lan, X. Wang, D. Wang, Z. Liu, Y. Zhang, Task caching, offloading, and resource allocation in d2d-aided fog computing networks, IEEE Access 7 (2019) 104876–104891.

[61] Q. Li, Y. Zhang, Y. Li, Y. Xiao, X. Ge, Capacity-aware edge caching in fog computing networks, IEEE Trans. Veh. Technol. 69 (8) (2020) 9244–9248.

[62] M.H. Shahid, A.R. Hameed, S. ul Islam, H.A. Khattak, I.U. Din, J.J. Rodrigues, Energy and delay efficient fog computing using caching mechanism, Comput. Commun. 154 (2020) 534–541.

[63] S. Bhandari, N. Ranjan, P. Khan, H. Kim, Y.-S. Hong, Deep learning-based content caching in the fog access points, Electronics 10 (4) (2021) 512.

[64] N. Gupta, S.K. Dhurandher, et al., Efficient caching method in fog computing for internet of everything, Peer to Peer Network Appl. 14 (1) (2021) 439–452.

[65] M. Aazam, S. Zeadally, K.A. Harras, Offloading in fog computing for IoT: review, enabling technologies, and research opportunities, Future Generat. Comput. Syst. 87 (2018) 278–289.

[66] L. Liu, Z. Chang, X. Guo, S. Mao, T. Ristaniemi, Multiobjective optimization for computation offloading in fog computing, IEEE Internet Things J. 5 (1) (2017) 283–294.

[67] S. Chen, X. Zhu, H. Zhang, C. Zhao, G. Yang, K. Wang, Efficient privacy preserving data collection and computation offloading for fog-assisted IoT, IEEE Trans. Sustain. Compute. 5 (4) (2020) 526–540.

[68] S. Mu, Z. Zhong, D. Zhao, M. Ni, Joint job partitioning and collaborative computation offloading for internet of things, IEEE Internet Things J. 6 (1) (2018) 1046–1059.

[69] S. Xia, Z. Yao, Y. Li, S. Mao, Online distributed offloading and computing resource management with energy harvesting for heterogeneous mec-enabled IoT, IEEE Trans. Wireless Commun. 20 (10) (2021) 6743–6757.

[70] H. Flores, S. Srirama, Adaptive code offloading for mobile cloud applications: exploiting fuzzy sets and evidence-based learning, in: Proceeding of the Fourth ACM Workshop on Mobile Cloud Computing and Services, 2013, pp. 9–16.

[71] Z. Ning, P. Dong, X. Kong, F. Xia, A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things, IEEE Internet Things J. 6 (3) (2018) 4804–4814.

[72] Z. Chang, L. Liu, X. Guo, Q. Sheng, Dynamic resource allocation and computation offloading for IoT fog computing system, IEEE Trans. Ind. Inf. 17 (5) (2021) 3348–3357.

[73] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X.S. Shen, Energy efficient dynamic offloading in mobile edge computing for internet of things, IEEE Trans. Cloud Comput. 9 (3) (2021) 1050–1060.

[74] F. Chiti, R. Fantacci, B. Picano, A matching theory framework for tasks offloading in fog computing for IoT systems, IEEE Internet Things J. 5 (6) (2018) 5089–5096.

[75] Y. Liu, F.R. Yu, X. Li, H. Ji, V.C. Leung, Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access, IEEE Trans. Veh. Technol. 67 (12) (2018) 12137–12151.

[76] N. Di Pietro, E.C. Strinati, An optimal low-complexity policy for cache-aided computation offloading, IEEE Access 7 (2019) 182499–182514.

[77] Y. Xiao, M. Krunz, Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.

[78] A.-C. Pang, W.-H. Chung, T.-C. Chiu, J. Zhang, Latency-driven cooperative task computing in multi-user fog-radio access networks, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2017, pp. 615–624.

[79] H. Gupta, A. Vahid Dastjerdi, S.K. Ghosh, R. Buyya, ifogsim, A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments, Software Pract. Ex. 47 (9) (2017) 1275–1296.

[80] S. Akbar, S.U.R. Malik, S.U. Khan, R. Choo, A. Anjum, N. Ahmad, A game-based thermal-aware resource allocation strategy for data centers, IEEE Trans. Cloud Comput. 9 (3) (2021) 845–853.

[81] U. Idachaba, F. Wang, A community-based cloud computing caching service, in: 2015 IEEE International Congress on Big Data, IEEE, 2015, pp. 559–566.

[82] M. Aazam, E.-N. Huh, Fog computing and smart gateway based communication for cloud of things, in: 2014 International Conference on Future Internet of Things and Cloud, IEEE, 2014, pp. 464–470.

[83] G. Jia, G. Han, H. Wang, F. Wang, Cost aware cache replacement policy in shared last-level cache for hybrid memory based fog computing, Enterprise Inf. Syst. 12 (4) (2018) 435–451.

[84] S. Safavat, N.N. Sapavath, D.B. Rawat, Recent advances in mobile edge computing and content caching, Digital Commun. Network. 6 (2) (2020) 189–194.

[85] B. Assila, A. Kobbane, M. El Koutbi, A many-to-one matching game approach to achieve low-latency exploiting fogs and caching, in: 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), IEEE, 2018, pp. 1–2.

[86] O. A. Khan, S. U. Malik, F. M. Baig, S. U. Islam, H. Pervaiz, H. Malik, S. H. Ahmed, A Cache-Based Approach toward Improved Scheduling in Fog Computing, Software: Practice and Experience.

[87] B. Assila, A. Kobbane, A. Walid, M. El Koutbi, Achieving low-energy consumption in fog computing environment: a matching game approach, in: 2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON), IEEE, 2018, pp. 213–218.

[88] J. Xu, K. Ota, M. Dong, Saving energy on the edge: in-memory caching for multi-tier heterogeneous networks, IEEE Commun. Mag. 56 (5) (2018) 102–107.

[89] S. Alonso-Monsalve, F. García-Carballeira, A. Calderón, Fog computing through public-resource computing and storage, in: 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), IEEE, 2017, pp. 81–87.

[90] J. Qadir, B. Sainz-De-Abajo, A. Khan, B. García-Zapirain, I. De La Torre-Díez, H. Mahmood, Towards mobile edge computing: taxonomy, challenges, applications and future realms, IEEE Access 8 (2020) 189129–189162.

[91] P.K. Dey, M. Yuksel, An economic analysis of cloud-assisted routing for wider area sdn, IEEE Trans. Network Service Manage. 17 (1) (2019) 445–458.

[92] X. Wang, S. Leng, K. Yang, Social-aware edge caching in fog radio access networks, IEEE Access 5 (2017) 8492–8501.

[93] A. Araldo, D. Rossi, F. Martignon, Cost-aware caching: caching more (costly items) for less (isps operational expenditures), IEEE Trans. Parallel Distr. Syst. 27 (5) (2015) 1316–1330.

[94] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A survey on IoT security: application areas, security threats, and solution architectures, IEEE Access 7 (2019) 82721–82743.

[95] A. Ghosh, O. Khalid, R.N. Rais, A. Rehman, S.U. Malik, I.A. Khan, Data offloading in IoT environments: modeling, analysis, and verification, EURASIP J. Wirel. Commun. Netw. (1) (2019) 1–23, 2019.

[96] R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: a taxonomy, survey and future directions, in: Internet of Everything, Springer, 2018, pp. 103–130.

[97] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: a survey, Future Generat. Comput. Syst. 97 (2019) 219–235.

[98] Z.M. Nayeri, T. Ghafarian, B. Javadi, Application placement in fog computing with AI approach: taxonomy and a state of the art survey, J. Netw. Comput. Appl. (2021) 103078.

[99] Y. Jiang, H. Feng, F.-C. Zheng, D. Niyato, X. You, Deep learning-based edge caching in fog radio access networks, IEEE Trans. Wireless Commun. 19 (12) (2020) 8442–8454.

[100] M.S. Elbamby, M. Bennis, W. Saad, Proactive edge computing in latency-constrained fog networks, in: 2017 European Conference on Networks and Communications (EuCNC), IEEE, 2017, pp. 1–6.

[101] M. Chen, W. Saad, C. Yin, Resource management for wireless virtual reality: machine learning meets multi-attribute utility, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–7.

[102] M. Chiang, T. Zhang, Fog and IoT: an overview of research opportunities, IEEE Internet Things J. 3 (6) (2016) 854–864.

[103] G. Lee, W. Saad, M. Bennis, Online optimization for low-latency computational caching in fog networks, in: 2017 IEEE Fog World Congress (FWC), IEEE, 2017, pp. 1–6.

[104] X. Huang, G. Xue, R. Yu, S. Leng, Joint scheduling and beamforming coordination in cloud radio access networks with qos guarantees, IEEE Trans. Veh. Technol. 65 (7) (2015) 5449–5460.

[105] S.-H. Park, O. Simeone, S.S. Shitz, Joint optimization of cloud and edge processing for fog radio access networks, IEEE Trans. Wireless Commun. 15 (11) (2016) 7621–7632.

[106] R. Tandon, O. Simeone, Cloud-aided wireless networks with edge caching: fundamental latency trade-offs in fog radio access networks, in: 2016 IEEE International Symposium on Information Theory (ISIT), IEEE, 2016, pp. 2029–2033.

[107] H. Zhou, M. Tao, E. Chen, W. Yu, Content-centric multicast beamforming in cache-enabled cloud radio access networks, in: 2015 IEEE Global Communications Conference (GLOBECOM), IEEE, 2015, pp. 1–6.

[108] Y. Wei, F.R. Yu, M. Song, Z. Han, Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor–critic deep reinforcement learning, IEEE Internet Things J. 6 (2) (2018) 2061–2073.

[109] Y. Ugur, Z.H. Awan, A. Sezgin, Cloud radio access networks with coded caching, in: WSA 2016; 20th International ITG Workshop on Smart Antennas, VDE, 2016, pp. 1–5.

[110] D. Chen, S. Schedler, V. Kuehn, Backhaul traffic balancing and dynamic content-centric clustering for the downlink of fog radio access network, in: 2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2016, pp. 1–5.

[111] D. Wang, H. Cheng, D. He, P. Wang, On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices, IEEE Syst. J. 12 (1) (2016) 916–925.

[112] F. Alghamdi, S. Mahfoudh, A. Barnawi, A Novel Fog Computing Based Architecture to Improve the Performance in Content Delivery Networks, Wireless Communications and Mobile Computing, 2019.

[113] Q. Xu, Z. Su, S. Guo, A game theoretical incentive scheme for relay selection services in mobile social networks, IEEE Trans. Veh. Technol. 65 (8) (2015) 6692–6702.

[114] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu, Y. Jia, An edge caching scheme to distribute content in vehicular networks, IEEE Trans. Veh. Technol. 67 (6) (2018) 5346–5356.

[115] I.F. Siddiqui, S.U.-J. Lee, A. Abbas, A.K. Bashir, Optimizing lifespan and energy consumption by smart meters in green-cloud-based smart grids, IEEE Access 5 (2017) 20934–20945.

[116] R. Ali, I. Ashraf, A.K. Bashir, Y.B. Zikria, Reinforcement-learning-enabled massive internet of things for 6g wireless communications, IEEE Commun. Standard Magazine. 5 (2) (2021) 126–131.

[117] R. Tapwal, N. Gupta, Q. Xin, Data Caching at Fog Nodes under IoT Networks: Review of Machine Learning Approaches.

[118] A. Akbar, M. Ibrar, M.A. Jan, A.K. Bashir, L. Wang, Sdn-enabled adaptive and reliable communication in IoT-fog environment using machine learning and multiobjective optimization, IEEE Internet Things J. 8 (5) (2020) 3057–3065.

[119] H. Zhu, Y. Cao, W. Wang, T. Jiang, S. Jin, Deep reinforcement learning for mobile edge caching: review, new features, and open issues, IEEE Network 32 (6) (2018) 50–57.

[120] C. Sang, J. Wu, J. Li, A.K. Bashir, F. Luo, R. Kharel, Ralaas: resource-aware learning-as-a-service in edge-cloud collaborative smart connected communities, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.

[121] M. Habib ur Rehman, P.P. Jayaraman, S.U.R. Malik, A.U.R. Khan, M. Medhat Gaber, Rededge, A novel architecture for big data processing in mobile edge computing environments, J. Sens. Actuator Netw. 6 (3) (2017) 17.

[122] M. Chen, U. Challita, W. Saad, C. Yin, M. Debbah, Artificial neural networks-based machine learning for wireless networks: a tutorial, IEEE Commun. Survey Tutorial. 21 (4) (2019) 3039–3071.

[123] K. Katevas, I. Leontiadis, M. Pielot, J. Serrà, Practical processing of mobile sensor data for continual deep learning predictions, in: Proceedings of the 1st International Workshop on Deep Learning for Mobile Systems and Applications, 2017, pp. 19–24.

[124] L. Ale, N. Zhang, H. Wu, D. Chen, T. Han, Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network, IEEE Internet Things J. 6 (3) (2019) 5520–5530.

[125] A.K. Bashir, R. Arul, S. Basheer, G. Raja, R. Jayaraman, N.M.F. Qureshi, An optimal multitier resource allocation of cloud ran in 5g using machine learning, Trans. Emerge Telecommun. Technol. 30 (8) (2019) e3627.

[126] D. Lan, A. Taherkordi, F. Eliassen, L. Liu, Deep reinforcement learning for computation offloading and caching in fog-based vehicular networks, in: 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), IEEE, 2020, pp. 622–630.

[127] S. Yao, S. Hu, Y. Zhao, A. Zhang, T. Abdelzaher, Deepsense: a unified deep learning framework for time-series mobile sensing data processing, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 351–360.

[128] B. Bharath, K.G. Nagananda, H.V. Poor, A learning-based approach to caching in heterogenous small cell networks, IEEE Trans. Commun. 64 (4) (2016) 1674–1686.

[129] K. Guo, C. Yang, T. Liu, Caching in base station with recommendation via q-learning, in: 2017 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2017, pp. 1–6.

[130] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60.

[131] E. Baştuğ, M. Bennis, M. Debbah, A transfer learning approach for cache-enabled wireless networks, in: 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), IEEE, 2015, pp. 161–166.

[132] A. Sadeghi, G. Wang, G.B. Giannakis, Deep reinforcement learning for adaptive caching in hierarchical content delivery networks, IEEE Trans. Cognitive Commun. Network. 5 (4) (2019) 1024–1033.

[133] R. Ali, Y.B. Zikria, B.-S. Kim, S.W. Kim, Deep reinforcement learning paradigm for dense wireless networks in smart cities, in: Smart Cities Performability, Cognition, & Security, Springer, 2020, pp. 43–70.

[134] S. Li, J. Xu, M. van der Schaar, W. Li, Trend-aware video caching through online learning, IEEE Trans. Multimed. 18 (12) (2016) 2503–2516.

[135] Y.B. Zikria, S.A. Malik, H. Ahmed, S. Nosheen, N.Z. Azeemi, S.A. Khan, Video transport over heterogeneous networks using sctp and dccp, in: Wireless Networks, Information Processing and Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 180–190.

[136] X. Zhang, Y. Li, Y. Zhang, J. Zhang, H. Li, S. Wang, D. Wang, Information caching strategy for cyber social computing based wireless networks, IEEE Trans. Emerge Topic. Compute. 5 (3) (2017) 391–402.

[137] R. Ali, Y.B. Zikria, S. Garg, A.K. Bashir, M.S. Obaidat, H.S. Kim, A federated reinforcement learning framework for incumbent technologies in beyond 5g networks, IEEE Network 35 (4) (2021) 152–159.

[138] S. Niknam, H.S. Dhillon, J.H. Reed, Federated learning for wireless communications: motivation, opportunities, and challenges, IEEE Commun. Mag. 58 (6) (2020) 46–51.

[139] N.C. Luong, Y. Jiao, P. Wang, D. Niyato, D.I. Kim, Z. Han, A machine-learning-based auction for resource trading in fog computing, IEEE Commun. Mag. 58 (3) (2020) 82–88.