Review Article

Zahraa A. Jaaz*, Mohd Ezanee Bin Rusli, Nur Azzammudin Rahmat, and Maythem Kamal Abbas Al-Adilee

Latency optimization approaches for healthcare Internet of Things and fog computing: A comprehensive review

https://doi.org/10.1515/jisys-2024-0543
received December 29, 2024; accepted April 05, 2025

Abstract: The Healthcare Internet of Things (H-IoT) has made significant strides in transforming remote diagnostics, real-time monitoring, and data collection. However, achieving low latency remains a challenge, even for systems that critically depend on it, such as emergency response systems and remote surgeries. Fog computing (FC) emerges as a promising solution to mitigate latency issues by bringing computational resources closer to edge devices. This review provides a comprehensive analysis of latency optimization approaches within the H-IoT, framed within the context of FC. The study focuses on architectural enhancements, data communication methods, resource management strategies, and load distribution techniques aimed at reducing latency. Additionally, it evaluates existing challenges related to energy efficiency and trade-offs, security considerations, and scalability potential. Based on an analysis of recent advancements in the field, this study outlines potential development trends and future research directions for building adaptive, fast, and secure H-IoT systems. The study reaffirms the pivotal role of FC in shaping the future of healthcare services, particularly those with stringent low-latency requirements.

Keywords: latency, fog computing, healthcare, healthcare Internet of Things, Internet of Things, artificial intelligence

1 Introduction

The Healthcare Internet of Things (H-IoT) is rapidly transforming traditional healthcare systems by enabling real-time monitoring and diagnosis [1]. H-IoT connects smart devices, wearable sensors, and cloud services to create healthcare ecosystems that enhance patient care and administrative policies. However, if not effectively managed, this technology can introduce latency in data processing and transmission, which is unacceptable for time-sensitive healthcare applications such as emergency response, telemedicine, and remote surgeries [2]. Reducing latency allows healthcare experts more time to provide accurate responses, directly impacting

* **Corresponding author: Zahraa A. Jaaz,** College of Computing and Informatics, Universiti Tenaga Nasional, Selangor 59200, Malaysia; Computer Department-College of Science - Al Nahrain University, Jadriya, Baghdad 10072, Iraq, e-mail: PT21050@student.uniten.edu.my, zahraa.jaaz@nahrainuniv.edu.iq

Mohd Ezanee Bin Rusli: College of Computing and Informatics, Universiti Tenaga Nasional, Selangor 59200, Malaysia, e-mail: ezanee@uniten.edu.my

Nur Azzammudin Rahmat: College of Engineering, Universiti Tenaga Nasional, Selangor 59200, Malaysia, e-mail: azzammudin@uniten.edu.my

Maythem Kamal Abbas Al-Adilee: School of Computing, Asia Pacific University of Technology and Innovation (APU), Kuala Lumpur 57000, Malaysia, e-mail: maythem.abbas@apu.edu.my

patient outcomes. According to the International Data Corporation (IDC), the number of connected devices is estimated to reach between 41.6 billion and 1 trillion by 2025, generating an unprecedented volume of data measured in zettabytes. This trend is driven by the proliferation of IoT devices, the growing use of social networks, mobile applications, a rapidly increasing global population, and a technology-driven lifestyle [3].

In healthcare, IoT devices produce vast amounts of multimedia data that require effective management. Cloud servers worldwide handle these data, including analysis, storage, and preprocessing. The sheer volume of data has significant implications for healthcare, making it crucial to extract meaningful “signals” from the “noise.” Cloud computing plays a vital role in connecting IoT devices to the healthcare industry. Data generated by IoT devices is typically processed, filtered, preprocessed, and aggregated using cloud services. However, cloud computing has limitations, particularly as data transmission rates increase. Issues such as high response times, service latency, and data loss become more pronounced, degrading the quality of service (QoS) for end-users. These challenges are especially critical for IoT systems requiring real-time data processing, such as healthcare and city management systems. Healthcare systems rely on real-time data processing to address urgent situations promptly. The current service model, which requires IoT devices to connect to routers and gateways before reaching the cloud, introduces significant computational latency. The greater the distance between IoT devices and the cloud, or the more routers deployed in the network, the longer the data transmission path and the higher the bandwidth consumption. These challenges highlight the need for solutions that minimize latency and reduce network bandwidth usage to meet the real-time demands of healthcare IoT systems [4–6].

The integration of IoT with fog computing (FC) presents immense potential. To fully realize these possibilities, sufficient networking infrastructure must be provided to maintain low latency for IoT applications. FC plays a critical role in the execution and processing of data from IoT devices. It offers a robust solution to address these challenges by enabling applications to run and data to be processed closer to the IoT devices that generate the data. This approach is particularly effective in overcoming the limitations of centralized cloud computing and enhancing the efficiency of IoT systems. FC emerges as a promising solution to latency issues in Healthcare IoT systems. By decentralizing and distributing computing resources closer to edge devices, FC reduces reliance on centralized cloud services, enabling faster data processing and decision-making. It also helps alleviate network traffic and improves the overall availability of healthcare services. However, integrating FC into H-IoT architectures introduces additional complexities, particularly in resource management, task allocation, and ensuring data privacy while minimizing latency. Addressing these challenges is essential to fully leverage the benefits of FC in H-IoT systems [7–9].

This review systematically identifies and discusses the state-of-the-art approaches regarding the latency optimization of H-IoT systems using FC. It analyzes architectural improvements, resource management strategies, and communications processes aimed at improving system performance. Besides, it enumerates the current challenges and open issues, which include energy consumption, scalability, and the trade-off between latency and security. Based on the synthesis of recent advancements, this work seeks to review and understand latency management in the context of fog supported H-IoT systems to inform future improvement of similar healthcare technologies. The main contributions of this study are:

- This work provides a comprehensive overview of the sources and impacts of latency in H-IoT applications. It highlights the critical importance of low-latency solutions for expanding healthcare use cases, including effective telemedicine, real-time monitoring, and emergency response systems.
- The study explores how FC addresses latency challenges in H-IoT systems by distributing computation and resource-intensive tasks closer to data collection points. Quantitative analyses support qualitative insights, emphasizing the architectural and operational benefits uniquely suited for healthcare environments.
- The work assesses current strategies for task scheduling, resource allocation, and optimized communication protocols in fog-based H-IoT systems, focusing on their effectiveness in reducing latency.
- The review identifies limitations in existing FC approaches for latency minimization, including energy consumption, system capacity, and the latency-security trade-off, providing a foundation for future research.
- By consolidating knowledge from current research, this systematic literature review offers actionable recommendations for designing and implementing latency-sensitive H-IoT systems, aiding healthcare administrators and system designers in their efforts.

The rest of this study is organized as follows: Section 2 provides the requirement of Healthcare IoT Systems. The basic of FC is presented in Section 3. Section 4 provides the related works on latency reduction.

Existing algorithms and techniques for latency reduction in cloud and IoT using FC are discussed in Section 5. Section 6 presents minimizing packet loss and data traffic. Techniques to achieve QoS requirement for IoT is discussed in Section 7. Section 8 provides FC performance tools. The gap and challenges are discussed in Section 9. Finally, the conclusion and future works are presented in Section 10.

2 Requirement of healthcare IoT systems

By 2030, 750.5 zettabytes of data will be created by billions of IoT devices. To efficiently compute and transfer data, reducing latency is essential for improving cloud response and service time [2]. However, end-users may still experience increased latency due to the physical distance and network congestion between devices and the cloud. In this context, latency is routinely measured and analyzed. It refers to the time taken for data to travel from the source (point A) to the destination (point B). Total cloud latency typically comprises three components: processing delay, transmission delay, and queueing delay [3]. The total latency in the cloud is the sum of delay throughout the transmission channel, which includes the time a packet spends on the wire or fiber, as well as processing time at each intermediary node. As a result, healthcare IoT requires minimal network, service, and computing latency for real-time applications. The proliferation of IoT devices has resulted in the creation of massive amounts of data. As a consequence, there are a large number of requests for related services from users via the network. For real-time inquiries, most IoT applications demand low latency and quick responses. The advent of IoT in recent years has created tremendous worry among services that require real-time data [4]. The probability of making a mistake is related to the magnitude of the data transmission. As a result, end-users may suffer poor QoS. End-users faced unacceptably high transmission latency and poor services as a result of the massive volume of data created by IoT devices [5].

The difficulty of synchronization between a client's request and the server's answer is exacerbated by high service latency. Furthermore, excessive latency in clouds is caused by network congestion between IoTs and clouds. Communication between IoT devices, the cloud, and end-users are always multi-hop. To be able to forward data packets to end-points, gateway nodes are required. They are determined by the distance between the remote server and the number of routers used [6]. For information bundle sending to end-focuses, the more extended the distance, the more gateway hubs are required. Besides, the method is deferred in the event that the parcel does not stream straight by means of the switch to arrive at the end-point; the bundle might need to go through a few switches. As a rule, switch parcels were deferred by a couple of milliseconds, bringing about a couple of milliseconds for each bundle full circle time. These switch gateway hubs create latency because of the basic information calculation delay. The proficiency of the network and the nature of the steering gear influence reaction times to end-client requests. Information duplication to many cloud server farms in different trans-mainland areas would be deferred too. These deferrals because of high network latency are an immense worry in the IoT-cloud framework, since the network state is continually changing with time [7]. Network latency [7] is the time it takes for a message or communication to move between various locations.

For software applications that must function on a global scale or high-performance software systems that require a very quick response time, network latency accounts for a considerable amount of the necessary reaction time. The unpredictable nature of the network leads latencies to be unpredictable. When data are transported from IoT to cloud servers [8], it undergoes a new life cycle. The quantity of data stored on servers and devices reduces their energy consumption. As a result of the high latency, data traffic, energy, and network consumption, the healthcare IoT-cloud system becomes incapable of providing real-time services to patients and clinicians. Therefore, there is an opportunity and urgency to engage in this area.

3 FC

Whether centralized or decentralized computing models should be used has been argued in recent decades based on two primary sets of difficulties. To begin, consider efficiency vs effectiveness while seeking answers using a rationalistic technique. Second, there are the politics of organizations and resources. At the turn of the century, the centralized cloud computing model was established [9]. To avoid excessive latency, location awareness, and other issues that may arise in a centralized method, it is necessary to move computing from centralized to a unique decentralized technique. FC was coined as a result of the shift from centralized to decentralized computing [10].

FC presented itself as an infrastructure design by Cisco Systems in 2012 [11]. Fog operates across three different levels within network systems according to their definition. The three main characteristics of data collection under FC include sensors and vehicles and ships and roads while also processing the data faster than one second for decision-based measures [12]. FC allows cloud services to approach end-users through distributed compute resources and storage facilities while providing enhanced security together with mobility features and low-latency performance and increased privacy and network bandwidth. Applications that require real-time processing receive significant advantages from using FC platforms [13]. Fog layer nodes possess sufficient processing capabilities which enhance both energy efficiency of the system's sensor nodes along with data transmission performance [1]. A system-level architecture named FC consists of multiple nodes, which form its primary main components that deliver cloud-based features. Different connection systems including both wireless and wired enable fog nodes to form a network. One key feature of FC allows networks to establish connections with the cloud. The OpenFog Consortium Architecture Working Group (2017) states that FC puts mission-critical data-intensive applications ahead in their network infrastructure by moving functions such as communication, storage, computation, and control to data creation locations at network edges. Three main characteristics describing FC are as follows:

- **Low latency and location-awareness:** Quality services significantly benefit from FC because the network edge lacks sufficient support. FC brings low latency behavior and geographical diversity through its network-edge position by bringing cloud control and processing capabilities closer to local devices [14]. FC fulfills the timing requirements that emergency and healthcare services need according to research by Ala'anzy et al. [15] and Baskar et al. [16].
- **Geographical distribution:** FC creates a dispersed network topology through its deployment structure of distributed fog nodes, which differs from centralized cloud distribution. The spread nature of fog layer deployments enables the system to distribute data analytics and processing throughout the network thus enabling it to deliver fundamental services together with locally managed distributed systems that provide excellent quality services [17].
- **Large-scale sensor network:** FC supports large-scale sensor networks through professional management of end devices by adding or subtracting nodes to optimize latency and response time [18].
- **Mobility:** FC uses an architecture design which exhibits built-in support for mobile applications through all-inclusive management of IoT and mobile sensor functions. The solution requires devices to connect directly with fog nodes so that cloud servers become less essential for operation. The primary strength of FC consists of its mobility features since it allows devices to move across networks without demanding reconfiguration procedures. The technology benefits dynamic environments like vehicular networks, smart cities and industrial automation because moving devices require no reconfiguration. The network edge implementation of data processing through FC yields both low-latency performance and high-speed responses that are essential for applications needing real-time decisions. The fast processing of traffic data by vehicles' mobile sensors, which communicate with local fog nodes, allows both better traffic management and decreases vehicle congestion. FC suits applications, which feature mobile devices because its high mobility level provides an optimal framework [19].
- **Heterogeneity:** FC platforms naturally combine diverse elements because they provide various storage and network and computational services. These services undergo virtualization and deployment through a layered system between end devices and the cloud to support diverse environments. Absolute diversity marks fog computing by letting its structure unite different kinds of devices and technology protocols together. The variety of fog nodes encompasses routers as well as gateways and access points and specialized edge devices, which serve applications in different environments ranging from urban to remote industrial locations. The adaptable nature of FC lets it support numerous applications which range from domestic settings to extensive industrial IoT (IIoT) platforms [20].

- **Real-time interaction:** Due to its nature, FC stands out as an excellent choice for time-sensitive applications that need instant processing abilities. Data processed through FC operates within local network regions instead of having to route information through traditional cloud management servers. The processing happens locally at the network edges through fog nodes, which enables both quick responses and low latency that are vital for monitoring systems and emergency response and autonomous vehicles. A healthcare monitoring system makes use of fog nodes to instantly analyze patient data at the edge of the network for early identification of anomalies followed by emergency medical assistance. The implementation of FC in industrial automation helps manage real-time operations of machines for efficient and secure industrial operations [5].
- **Prevalent to wireless access:** FC has been created to support wireless networks thus making it perfectly suited for new mobile and IoT systems. A variety of wireless communication technologies including mobile cellular gateways along with Wi-Fi access points operate as fog nodes for edge-based connectivity and data processing operations. The integration of wireless access stands vital because there are many situations where traditional wired connections become impossible to utilize. This includes rural areas or situations that occur inside or outside buildings and when using mobile systems. In smart agriculture systems wireless fog nodes operate to grab data from field sensors before conducting spatial assessment of soil situations and meteorological elements and crop vitality at real-time intervals. The capability to employ wireless access throughout FC systems allows its deployment across multiple geographic settings that span smart cities to industrial locations in isolated areas [21].
- Interoperability is the fundamental characteristic of FC involving linking different operational domains through service integration, which allows service federation across multiple platforms. The devices together with applications as well as services, which exist in a FC ecosystem, typically originate from different vendors while using varying protocols. FC platforms need capability to speak with various systems for operation continuity. Wide-area data streaming applications such as video surveillance and telemedicine and smart grid management need this feature intensively. The smart city demonstrates the importance of interoperability among fog nodes from departments such as traffic management and public safety and energy distribution because they need shared data to coordinate city functions. The ability of FC to support interoperability enables the development of unified complex systems which can utilize the advantages of various technology platforms [22].

4 Recent works on minimizing latency in IoT and cloud using FC

This section includes a general review for recent research works to minimize the network usage along with latency and response time in IoT and cloud using FC. Furthermore, the section also includes the various factors affecting the latency in IoT and cloud which prompts delay in the services to end-clients. The section is divided into two sub-sections. The various factors which affect the network and service latency in IoT, and cloud are:

- Propagation delay is caused by a standard number of hops on the route to the destination server. (i.e., the amount of time a packet spends traveling across a wire or fiber; the amount of time depends on the actual distance.) [9].
- Large data transfer to the cloud from healthcare IoT devices.
- There is a ton of information transmission between both the cloud storage and IoT devices. Due to the high measure of information sent by IoT gadgets.
- Network congestion causes router queuing delays, which results in packet and path loss.
- Routers have a built-in processing latency.
- The cloud server's response time.
- Replication of data from one cloud data center to another can cause some service delays.
- An abnormally high burden on the cloud data center.

Therefore, real-world implementation research in this field is possible.

Tripathy et al. [23] proposed a new architecture in a cloud computing environment to integrate wireless body area networks (WBAN) with several modeling techniques for storage and optimization of real-time data. These data were further processed for efficient energy usage and low latency. However, they did not discuss the concept of FC. Kaur et al. [24] introduced the fuzzy inference system discipline along with data mining techniques. They discussed a novel technique for a distributed system. Next they applied fuzzy set rules in their proposed algorithm. The algorithm was based on a breadth-first strategy. Their proposed work supports distributed computing rather than centralized computing. In a similar context, Moustafa [25] used an FC architecture in the IoT-cloud. His work was able to process the IoT device data into meaningful information. The filtered data are further sent to end-users in real-time. Nandyala and Kim [26] presented a three-tier FC architecture for the e-healthcare system. The gap of their research was to overcome the real issues associated with high latency; they focused more on FC modeling techniques. Ijaz et al. [27] discussed the FC for smart devices and networks. The FC here acts as a middleware gateway device. This gateway device works like a computational device with CPU storage and processing capabilities. The gateway device was able to bring cloud features close to IoT devices and end-users. Similarly, Natesha and Guddeti [28] proposed an FC-based gateway to reduce network latency at the edge of networks. However, real world implementation is still lacking.

Tuli et al. [29] highlighted the problem of high latency in the medical healthcare system. They used various cloud-based services for the transmission of e-health data to end-users. However, they did not discuss the concept of FC in the healthcare system. Islam et al. [30] proposed an FC-based intelligent gateway, where fog nodes were used for data storage and distribution to several neighboring fog nodes and end-users. the work was able to reduce the latency in IoT-cloud environment. Similarly, Kaur and Aron [31] proposed an FC-based architecture to highlight the problem of load balancing and data replication among neighboring fog nodes. They pointed out the problem of high latency associated with data replication from one node to another. Li and Wang [32] introduced the concept of green cloud using WBAN for services related to healthcare. But they missed out the issues of high service latency and battery draining in WBAN and end-user devices. Darabkh and Alkhader [33] proposed a hybrid model which is a combination of FC and vehicular ad hoc networks (VANET's); their proposed model consists of a VANET "hub." This hub acts like a fog device to deal with large data. This device acts at the edge of networks. Likewise, Aazam and Huh [34] proposed an intelligent gateway that acts as a middleware between body sensors and cloud; their proposed work highlighted the issue of high latency in e-healthcare. The processing, filtering, and mining of healthcare data were done at the gateway. Similarly, Rahmani et al. [35] proposed an enhanced gateway for the healthcare system; they used FC-based technology to distribute the data packets among neighboring fog nodes, which acts as a gateway device. However, their gateway device lacks the decision-making approach.

5 Latency in IoT and cloud reduction techniques

Existing algorithms and techniques for latency reduction in cloud and IoT using FC are discussed in this section.

1. **iFogStor:** High latency is a big issue for time-critical IoT applications, according to Naas et al. [36]. They proposed the iFogStor method. The method was based on the FC principle. The heuristic technique was employed by iFogStor. It performed far better than the cloud, with latency decreased by more than 86 and 60% when compared to FC alternatives. Data placement issue was defined as a generalized assignment problem (GAP) in iFogStor, and two solutions were proposed: (1) an accurate solution based on integer programming, and (2) a heuristic method based on geographical zoning to speed up the process. Using the geographical zoning heuristic, issues with a high number of fog nodes may be solved quickly and effectively, allowing iFogStor to be runtime and scalable. In the combinatorial optimization literature, GAP is a well-known NP-Hard problem. It entails determining the most cost-effective assignment of n tasks to m agents (for example, n jobs to m processors).

   To limit all out storage and administration latency, iFogStor is a haze mindful runtime approach for putting away IoT information in a mist framework, for example, server farms. iFogStor thinks about the entire progressive plan, as well as the variety of gadgets and latencies between hubs, while putting information. They presented a system for recognizing the generated dataset, along with a strategy for fog storage nodes aimed at minimizing overall service latency during data storage and retrieval. Each piece of data may be stored and accessed with a certain delay, depending on its location and characteristics. The authors formulated the data placement problem as an integer programming model with two proposed solutions. The first was a careful reaction in case of truly huge scope applications, while the second was a gap and vanquish heuristic system to lessen critical thinking time. The idea behind iFogStor is to leverage the heterogeneity and geographic distribution of fog nodes to reduce overall latency in storing and retrieving data. Future improvements will require a more accurate model and architecture to support time-sensitive healthcare IoT applications.

2. **FC-cloud fusion:** Shukla et al. [37] examined several issues and research problems related to the fog fusion and cloud for latency reduction, energy consumption, allocation of the resources, the optimization, and RAM utilization in IoT. They cast doubt on the expanding disparate devices number.
3. **Fog-Radio Access Network (F-RAN):** You et al. [38], for lowering IoTs and wireless devices latency, the authors described and recommended a F-RAN architecture. They proposed a latency-driven software algorithm that would enable them to offer services to several end-users with varying resource allocations. A resource may be shared by several users thanks to the algorithm. An F-RAN design that takes use of existing infrastructure, such as small cells and macro base stations, to achieve ultra-low latency via collaborative computing across several F-RAN nodes and near-range communications at the edge. This further defines the tradeoff between communication and computation across a large number of F-RAN nodes. According to their results, F-RAN can supply low latency services using latency-driven cooperative task computing.
4. **Live VM migration:** Osanaiye et al. [39] introduced an audit of distributed computing research articles, as well as a speculative live VM Migration system to further develop administration latency in IoT-haze cloud design. A notional live methodology for VM movement is proposed to guarantee and give assets and administrations to end-clients. As per the report, the system still cannot seems to be coordinated and sent in the real world.
5. **Many-to-one coordinating:** Chiti et al. [40] presented and outlined the test of improving cloudlet choice and limiting latency in IoT and haze networks by checking and managing the responsibility. For some-to-one coordinating IoT hubs with cloudlets, it was concocted to rate system. The matching game among IoT and cloudlet was likewise tackled utilizing a calculation.
6. **FC security service (FCSS):** Wu et al. [41] introduced the requirements for information-centric social networks (ICSN). The ICSN has many needs, including a deployment method, data mobility, low latency, and efficient end-node communication. Computational duties, resources, and intelligence are moved from remote and distant servers to the network's edge when FC is used in ICSN. The services of ICSN are protected by FC-based content-aware filtering. The authors deployed a FC security solution to protect ICSN. The authors did not address the problem of excessive computational delay induced by the fog nodes' enormous number of computational workloads. Large data transmission at the network's edge results in a lot of data traffic and network congestion, as well as a lot of network delay.
7. **Cost-effective schema:** Yadav et al. [42] discussed issues such as latency, hardware failure, resource limits in FC, and software failure, all of which are considered significant in network function virtualization (NFV). In fog and cloud networks, the authors provided a cost-effective IoT service deployment architecture. The suggested schema assesses the availability of virtual network functions with the potential to enhance software function chaining. Their work is the most cost-effective and scalable, according to them. However, despite the fact that the authors stated that high latency is a key problem for IoT deployment, no effort was done to reduce service function chaining latency via cloud servers and fog nodes.
8. **Service popularity-based smart-resources partitioning (SPSRP):** Li et al. [43] raised the figuring and asset proficiency issues. On haze hubs, there was an issue with coordination between computational proficiency and asset productivity. Thus, there is a ton of traffic and network clog. The authors fostered a SPSRP (administration fame based savvy assets dividing) method for IoTs and FC. In IoTs and mist servers, their proposed concentrate on exertion diminishes defer time, reaction time, and adaptation to internal failure. The authors, then again, did not resolve the issue of extreme network and handling delay in IoTs.

9. **Cloud-based service:** Haghgoo et al. [44] discussed the challenges of high latency, huge data transfer, and high data traffic caused by a large number of heterogeneous applications. For healthcare applications, cloud services are unable to fulfill the required QoS. As a result, the authors offered cloud-fog-based services for healthcare applications, as well as a reference architecture. The findings were analyzed for data transmission optimization, latency minimization, and power consumption reduction. Cost efficiency, energy usage, and network latency were improved as a consequence of the study. However, there has not been much research into reducing excessive computational delay in healthcare applications.
10. **FC-based system with a central component:** Almaiah et al. presented a centric FC-based method for cloud storage [45] to preserve and secure data inside the cloud environment; they highlighted numerous concerns connected to data security, data transmission latency, and cloud privacy. In cloud servers, the privacy of users' data is paramount. To safeguard the data from unwanted access and harmful assaults, a XoR combination was utilized. In terms of data packet processing time, the findings were confirmed. To identify data change with the highest likelihood, the scientists employed a novel approach based on a hash algorithm.
11. **Software-defined network (SDN):** Lakhan et al. [46] proposed an SDN for IIoTs: A technique for task execution based on task priority using a computing mode system. Real-time performance is accomplished with FC when the task priority is used. They were able to minimize IoTs latency, fog, end-users, and cloud using their suggested strategy. However, there are no computing services for mobile devices in the study effort.

5.1 AI techniques for latency minimization

Current machine learning approaches and algorithms for minimizing high latency in the IoT and cloud are discussed in this section.

1. **Trendy person:** Nishtala et al. proposed Hipster as a method to address the issues of QoS in end-client necessities [47]. A blend of heuristic and support learning was utilized in this strategy. In distributed computing, the coordinated AI impact controls latency for time-touchy applications. It supports asset advancement and the interpretation of latency-serious tasks to clump jobs. While allowing bunch handling, the proposed strategy increments throughput, asset ampleness, and reaction time. Trendy person's Heuristic Policy made it feasible for latency-requesting and group jobs to coincide in shared server farms. On the opposite side, the authors neglected to resolve the issue of the significant communication defer that exists between IoT gadgets, cloud servers, and end-clients.
2. **Hermes:** Kao et al. [48] introduced a strategy for lessening latency in portable figuring for time-basic circumstances. Hermes was the name of the system. The proposed method's essential job is to advance work tasks for gadgets with restricted assets. The proposed research project was focused on offloading figuring exercises. The authors introduced the plan of a NP-difficult issue system to additionally lessen latency utilizing this AI innovation. The authors did not resolve the issue of full circle time delay, which is brought about by lengthy communication delays between cloud servers, portable clients, and IoT gadgets. In any case, when contrasted with past flow research in this field, the authors had the option to lessen high calculation latency and high network by a greater number of rates.
3. **Essential square offloading:** Alam et al. [49] proposed a decentralized fundamental square offloading method for sending portable codes on a geologically scattered haze versatile network. The squares in the conveyed multi-specialist framework were moved utilizing a support learning approach. As a result, high latency and handling time for IoT have been decreased. Nonetheless, none of the exploration depicted above has gone into profundity on the issue of extreme latency. Thus, lessening extreme latency between medical care IoTs and the cloud is of huge interest.
4. **Data set pattern and dynamic metadata:** Waqar et al. [50] introduced a system to shield and get clients' information in the cloud from undesirable interruptions. The system was based on the idea of data set diagram and dynamic metadata. Following that, the unique metadata were reproduced for security, latency decrease, and other ongoing applications. To change the data set diagram, a few cryptographic techniques

were performed. The execution of the proposed system using support learning methods is a theme for future review.

5. **Fluffy based model:** Soleymani et al. fostered a fluffy based system to assemble exact and allowed information from vehicle impromptu networks [51] Preferred Vehicular ad hoc network (VANETs). In VANETs, the proposed worldview had the option to lessen latency. The VANET vehicles need exact data. Wrong information would cause disturbance and framework disappointment. The authors utilized fluffy rationale to make ongoing decisions while making rules for the VANET's trust model. Their proposition depended on past work that involved conveyed haze hubs to survey occasion accuracy in VANETs. The proposed model still cannot seem to be approved in the real world, as per the review.
6. **Mixture bio-roused calculation:** Rafique et al. [52] introduced a crossover bio-motivated calculation to lessen response and execution times in the IoT-haze cloud setting. Molecule multitude and Cat swarm enhancement were joined in the mixture method. The method was changed to more readily oversee asset accessibility and occupation planning for haze hubs. In the IoT–fog environment, future work will require the use of reinforcement learning techniques for resource management, including efficient allocation, scheduling, and utilization of fog computing resources.

5.2 Using traditional technologies to minimize latency

This section contains a full explanation, discussion, and analysis of current traditional latency minimization techniques utilized in cloud and IoT.

1. **Multipath transmission control protocol (MPTCP):** Grinnemo and Brunstrom [53] described the role of MPTCP in reducing latency and improving QoS for cloud and end-users. MPTCP delivers a decreased IoT applications latency, according to the findings. MPTCP, on the other hand, has a difficulty with packet loss. For cloud-based intensive traffic applications, their suggested efforts exhibit gains in terms of latency when compared to regular TCP.
2. **Binary asynchronous transfer mode (ATM):** Sambyo and Bhunia [54] utilized the paired ATM to lessen latency, which likewise settled the issue of parcel mistake in the cloud. Their discoveries proposed that double ATM beats customary methodologies. Aside from that, when the steady bit rate part is equivalent to or not exactly factor bit rate, hub handling delays, which are subject to various parcels, are decreased in paired ATMs variable bit rate.
3. **Change data capture (CDC):** Eccles et al. [55] established the CDC approach. The CDC assists in reducing latency. This suggested approach is utilized in data warehousing to reduce latency by eliminating the need to repeatedly load the datastore over time. Batch processing and CDC techniques are supported. The CDC keeps track of data changes in its system.
4. **Line:** Pan and McElhannon [56] depicted the idea of CORD for diminishing latency, edge virtualization, and offering start to finish types of assistance utilizing SDN and NFV. It joins NFV, SDN, and cloud advances. Line is a server farm that utilizes the virtualization approach. Line empowers IoT gadgets to move around and conveys SDN at the edge. They likewise talked about the NEBULA idea for IoT latency decrease and area mindfulness. It is used in MapReduce and other conveyed information serious applications. Cloud works at the network's edge. It is not, nonetheless, proper for a unified PC framework.
5. **Femto cloud:** The Femto cloud approach was proposed by Habak et al. [57]. A bunch of cell phones is utilized to make a variable, self-configurable, and multi-gadget portable cloud. Numerous cell phones might be arranged in a coordinated distributed computing administration utilizing this system. To help the accessible measurements while controlling gadget beating, the scheduler should allot liabilities using booking calculation open devices.
6. **Content delivery network (CDN):** Sajithabanu and Balasundaram mentioned the CDN in their study [58], which is used to reduce latency, response time, and network traffic. The CDN performs intelligent caching and distributes the data. It offers static data, while edge servers retain information on their own computers. A CDN caches its material in several places. Each PoP has many cache servers that are in charge of content delivery. The CDN is not designed to handle dynamic data. It can only be used with static data.

7. **Home cloud:** Lee et al. [59] employed the home cloud approach for delivery services of IoT applications, device automation, latency reduction, and automated orchestration. SDN and virtualization are supported by the home cloud, as well as many applications on the same network functionality and infrastructure. It is a cloud that cuts down on transmission time of the data.

6 Minimizing packet loss and data traffic

The problem of excessive packet loss and data traffic in the cloud and IoT is highlighted in this section. An itemized conversation and examination of flow research work are embraced to packet parcel loss and information traffic utilizing FC and regular methods.

1. **Offloading calculation:** Cao and Cai [60] laid out a system for asset enhancement and portion. Dividing the information parcels into smidgens may be utilized to convey assets. These bits are then conveyed to different hubs in the network. The work is scattered to other sub mists called cloudlets in view of the size of the errand and the quantity of assignments holding up in a line, a cycle known as figure offloading. This procedure will aid in node load balancing. This was the best choice for unloading. The suggested technique determines the resource capabilities of nodes with the goal of reducing total data traffic rate. They presented about how cloud computing may be used to offload computation for numerous users in a cloudlet context. The user has the option of offloading the data based on his or her own set of criteria and information. The scientists also presented a machine-learning system for mobile devices and apps that would cut energy use, bandwidth consumption, and network utilization.
2. **Cloudlet:** A cloudlet was suggested by Cavalcante et al. [61] to reduce the amount of data transmission between the edge device and the cloud. Various virtual machines make up the cloudlet. The devices were built with the intention of moving from one cloudlet to the next. According to the authors, this option allows the program to run quicker. The authors, however, confronted difficulties as a result of the application's long reaction time.
3. **FC framework:** Liu et al. [62] presented an FC framework. They presented a latency-reducing design and explored many problems and optimization options. They also spoke about resource allocation approaches for reducing load and network traffic while employing FC. However, there is no real-world application of the study findings.
4. **Asset portion:** Name et al. introduced and proposed a strategy [63] to deal with the asset allotment issue in IoT-mist cloud design. The method joins a portable IPV6 handover system and booking calculations to limit network use, information traffic, response time, latency in IoTs, and much more. However, their study does not include any work on reducing the amount of RAM used by processes.
5. **IoT and cloud interconnection:** Al-Fuqaha et al. [64] utilized FC to lead a profound and complete examination of the connections between distributed computing, edge figuring, and IoTs. They likewise featured on how current and impending advances could lessen latency and information traffic between cloud, end clients, and IoT gadgets. The dependability of crisis reaction application administrations, then again, is a main issue, bringing about huge deferrals and information misfortune.
6. **Communication that is cooperative:** Masri et al. [65] depicted how to involve FC to lessen network traffic and latency in an IoT-cloud situation. They contrived a methodology for haze hubs to cooperatively convey. In IoT-haze cloud framework, this assists with lessening information traffic and start to finish latency. Their review needs true application to check the worldview they offer.
7. **FC design and computation offloading:** Meng et al. [66] proposed a mixture system to lessen network traffic and energy utilization in IoTs using FC. Naha et al. [67] investigated the FC plan for limiting network traffic and latency in IoTs. They likewise fostered a scientific categorization for FC with regards to IoT.
8. **IoT system:** Yousefpour et al. introduced the IoT-mist cloud structure [68]. Also, they gave a logical way to deal with lessening IoT administration latency. The authors proceeded to make sense of how the FC method might be utilized to apply the idea of stretching out cloud administrations to the edge of networks

to increment framework execution and asset proficiency. The proposed design limits network transfer speed use as well as information traffic.

9. **FC review and examination:** Mukherjee et al. [69] distributed a total report on FC to lessen network traffic and latency in IoT. Also, the authors featured network applications, research issues, and FC establishments according to IoT and cloud. Besides, FC can possibly further develop IoT gadget execution and proficiency. Also, Mouradian et al. [70] led a careful assessment of the FC research issues. The study incorporates an FC scheme for latency reduction and a network traffic management algorithm. However, the findings of the study have not yet been applied in a real-world implementation.
10. **Virtual haze system:** A virtual mist structure was introduced by Li et al. [71]. To depict the virtual haze in the IoT framework, a NFV was proposed. This virtual methodology in FC had the option to lessen jitter, latency, and information traffic. In any case, in the proposed task, asset use is a vital issue.
11. **FogBus:** Tuli et al. [72] proposed a FC scheme for latency reduction, along with a network traffic management algorithm. However, the study's findings have not yet been validated through real-world implementation. Their proposed arrangement utilizes a blockchain-based innovation to lessen weighty information traffic while likewise shielding private IoT information from outside interlopers and programmers. To protect sensitive activities involving IoT data, the system employs multiple encryption techniques. Additionally, the proposed architecture enhances the integration of the IoT–Fog (Edge)–Cloud framework. FogBus gives stage skeptic execution and connection interfaces for IoT applications and figure cases. It assists designers with making applications, yet it likewise helps buyers in running numerous applications immediately and specialist co-ops in dealing with their assets.

7 Techniques to achieve QoS requirement for IoT

In the medical care industry, only a couple of uses need ongoing information. In such conditions, FC-based innovation is vital for meeting the medical care IoT's QoS needs. The IoT device might send information continuously utilizing this method. By lessening information transmission, FC makes communication between haze hubs and end gadgets simpler. Bluetooth and Zigbee are utilized to impart among these gadgets [73]. Painted health data (PHD) and important bodily functions are recorded by FC at the network's edge. These clinical devices are normally used in far off areas. The FC fulfills the QoS rules for medical care time-touchy applications when contrasted with the cloud. FC is important for a scattered neighborhood local area network. Cloud, then again, is more unified and utilizes a wide area network (WAN). The FC contraption is a savvy gadget. Due to its nearness to IoT gadgets, haze gadgets might execute an assortment of time-touchy medical care applications [74]. The broad sending of savvy edge gadgets and applications that need ongoing information handling has without a doubt required the expansion of distributed computing to the edge, otherwise called mist or edge figuring. FC is planned to work in pair with distributed computing, considering a new, more extravagant design that can take utilization of and incorporate both haze and cloud assets.

1. **FogPlan:** Yousefpour et al. [75] introduced a system called FogPlan to fulfill the QoS rule for low latency. The system was based on a unique help that gave mist hub applications. In their proposed study, they utilized an eager calculation. The convention design for administrations used between haze hubs is absent from the proposed work. There are also no solutions available for managing traffic between fog nodes that rely on learning. FogPlan is a platform designed to dynamically provision fog services while maintaining Quality of Service (QoS) through quality of dynamic fog service provision (QDFSP). QDFSP focuses on the adaptive deployment of application services on fog nodes, or the reallocation of currently deployed services, in order to meet the low latency and QoS requirements of applications, while simultaneously minimizing costs.
2. **FogTorch:** Brogi and Forti [76] proposed a worldview for latency and transfer speed improvement in IoT and cloud in view of a Java program named FogTorch. The model follows IoT QoS necessities. The model portrays functional fundamental elements of accessible framework (latency and transfer speed), programming part and thing connections, and business rules. The calculations proposed here handle these issues by adaptively figuring out where a part ought to be sent along the Cloud-to-Things continuum. To limit the hunt

space and observe a solitary qualified sending, the authors utilized a pre-handling and backtracking method utilizing a heuristic system. The proposed model and methods are executed in a proof-of-idea Java program called FogTorch to show their innovative suitability. FogTorch takes the framework and application details, as well as the pertinent things restricting and sending strategy, and creates reasonable arrangement plans. The proposed model might be utilized during runtime, planning, and sending. The methodology deliberately sticks to no norm for characterizing equipment and programming abilities. Programming abilities included the accessible operating systems (Linux, Windows, and so on) and introduced stages or systems, while equipment details stressed both consumable (e.g., RAM, storage) and non-consumable assets (e.g., design, CPU centers) (.NET, JDK, Hadoop MapReduce, and so on.). Besides, the authors laid out a haze framework in their model, yet they did not zero in on the model's constancy, mist hub sending timing, or cost.

3. **E-medical care needs examination and correlation:** Skorin-Kapov and Matijasevic investigated the QoS necessities for e-medical care [77]. This remembers necessities for ongoing information transmission for medical care IoT and telemedicine. Negligible full circle time deferral, or least help delay, is a rule for IoT gadgets and telemedicine tasks. The authors led a careful examination and correlation of e-medical care administration QoS necessities.
4. **FC and savvy medical care:** The importance of IoTs in shrewd medical care applications was featured by Baker et al. [78]. Different advances, troubles, and potential outcomes connected with medical care IoT, e-medical care, and telemedicine are examined. The authors proceeded to investigate how FC might assist with lessening latency and accomplish QoS necessities for time-basic applications in savvy medical care. They introduced a unique system for following patient wellbeing. The authors then went through the various wearable gadgets that might be utilized to screen and record patient crucial signs. In their proposed framework, they applied an AI method.
5. **Healthfog:** Tuli et al. [29] proposed a structure for a savvy medical care framework that coordinates IoT with FC to identify patient heart issues continuously utilizing profound learning-based approaches. The proposed design had the option to satisfy the medical care IoT QoS necessity. The system was made in view of latency-touchy applications, for example, clinical checking and flight control. The IoT in medical care makes a huge amount of information known as Big information. This information requires a huge estimation, after which the information is moved to data sets and from data sets to cloud server farms, bringing about a decline in framework execution. Thus, the proposed haze empowered cloud design fulfills the QoS for medical care IoT regarding power utilization, jitter, and coronary illness analytic forecast precision.

8 FC performance tools

The various simulation tools for evaluating the performance of the FC technique are covered in this section. This section also includes a full comparison of current tools. Framework (i.e., gadgets and networks), stage (which incorporates assets, administrations, and their organization), and applications are immensely significant parts of FC frameworks (which execute in the mist and have specific necessities). Resource management, scalability, and flexibility are all essential aspects of FC modeling and simulation [79]. FC, however, represents a complex environment in which users both consume services and execute their applications [80]. The allocation of services or resources depends on the roles of the participating nodes, whether as providers or consumers.

The FC worldview fits a wide assortment of utilizations with low latency necessities, time awareness, and assumptions for openness or proficiency from portable clients. Medical care, savvy environmental factors, augmented reality, intelligent transportation systems, public security, smart grid, and Industry 4.0 are only a couple of the areas where these applications might be found. To help the displaying of numerous sorts of uses, recreation devices incorporate choices for physical or virtual assets, network design, control systems, and information of the board [79]. FC is the most recent cloud computing expansion [80]. Although there is a large variety of cloud computing simulation tools, they cannot be utilized as-is for FC research; hence, they have been customized to fit the new demands. Simultaneously, new simulation tools, particularly for fog, have been suggested and created. This section does a survey study of all such instruments in the FC region.

The available simulation tools are summarized in Table 1. The first column refers to the issue that the different simulators are attempting to solve. The second column contains the metrics supplied by each simulator, which emphasizes the simulator's major goals. The list of simulators is included in the third column. iFogSim seems to be the most popular tool among researchers based on these data. The majority of simulators are designed to help with a particular FC issue.

Table 1: Comparison of simulation tools

Ref.	Simulation tools	Problem analysis	Metrics used
[81]	A Java tool and iFogSim as FogTorch	QoS in IoT networks	Processing delay, service delay, transmission delays, and propagation
[29]	iFogSim	Latency source in healthcare system	Communication latency
[60]	CloudSim	Best distribution of processing load between the cloud and the fog	Processing costs for every unit time
[76]	FogTorch and HealthFog	Resource allocation	Time cost and price
[63]	FogNetSim++	Allocation of FC resources under QoS constraints	System loss rate, CPU utilization, system throughput, system response time, and number of messages
[72]	FogBus	Modeling a classic healthcare monitoring approach	Response time and computing cost
[80]	iFogSim	FC performance for IoT applications	Energy consumption and latency
[80]	iFogSim	Services migration from the edge to the cloud	Reconfiguration cost, routing costs, transmission cost, and operational cost
[66]	CloudSim and ModFogSim	Offloading process optimization	Storage costs and processing

Table 2 presents complete details about different FC simulators that include their implementation platforms with fundamental metrics alongside their objectives for use. The field of FC needs crucial information about simulation tools to allow researchers and practitioners to make appropriate tool selections according to their distinct requirements. The table functions as a key reference by identifying the specific focus areas among different simulators, which include QoS, energy consumption, latency, and resource management capabilities. The decision-making process for FC system design becomes simpler through this tool especially regarding healthcare applications that need low latency and resource-efficient management. Also, Table 2 demonstrates how FC simulators possess diverse specializations, which align with the complexity of FC difficulties. Both iFogSim and MyiFogSim focus on resource management and latency since healthcare applications need real-time responsiveness yet other

Table 2: Simulator tools for FC

Simulator	Details
Edge-Fog [84]	Implemented using Python, focusing on QoS and energy consumption to distribute task processing across cloud resources
FogTorch [23]	Developed in Java, evaluating QoS, reliability of links and nodes, power consumption, security, and monetary costs to identify suitable application deployments over fog infrastructure
FogTorch II [46]	An extension of FogTorch, emphasizing resource utilization and QoS accuracy, with additional QoS profiles based on probability distributions
iFogSim [45]	Built as an extension of CloudSim using Java and JSON, analyzing energy consumption, network congestion, and operational costs to assess resource management policies
MyiFogSim [9]	An extension of iFogSim, specifically targeting latency for resource allocation purposes
FogNetSim++ [100]	Based on OMNeT++, incorporating energy modules, scheduling algorithms, and pricing models for general fog environment simulations
Fogbus [47]	Utilizes Java and iFogSim, measuring latency, energy, network, and CPU usage
FogTorch [23]	Based on CloudSim, focusing on computational and network costs for IoT resource management in FC

tools like FogTorch and FogNetSim++ tackle general aspects including reliability and security and power efficiency. Researchers can focus on solving particular fog computing problems due to the availability of various simulators, which tackle both performance optimization and energy-efficient operations. The table integration allows users to choose suitable tools and exposes development opportunities which guide the creation of advanced fog computing simulators for emerging application requirements. A technical analysis of the simulation tools is presented in Table 2 based on their capabilities. As observed, current simulators address only a portion of the simulation requirements in fog computing (FC). While most of them support multiple devices and components for fog infrastructure development, they vary in terms of the features and interactions they provide. To evaluate the suitability and effectiveness of each tool in addressing specific challenges in FC and IoT, a comprehensive assessment of existing simulation platforms has been conducted.

9 Identified research gaps

The most current study does not provide any recommendations for reducing latency. This is due to the fact that current data analytic techniques are built to handle enormous amounts of data but not real-time data processing and dispatching. Having millions of objects generate data and then sending it all to the cloud is neither scalable nor ideal for real-time decision making. Time-dependent IoT environments together with instantaneous requirements continue to grow at a rapid pace. Due to the processing ability of edge devices, it becomes necessary to implement a real-time fall detection system. The detection workload separation between edge devices linked to users and cloud servers enables this system to use the FC paradigm for analytics distribution throughout the network. FC paradigm originated from cisco as an answer to provide QoS requirements for IoT applications. Different authors described how FC fulfills the (network speed and system response time requirements) that IoT technology demands. FC ensures all latency reduction factors to make it an excellent choice for real-time data transmission between IoT devices and cloud systems. The main barrier resides in developing edge device resource selection protocols to determine which analytics applications will be delivered per device as a way to reduce delay and increase data speed. The assessment platform needs development for measuring how resource management rules execute on both IoT and FC infrastructure through repeatable testing methods. The key focus of this project exists in advancing IoT-based systems through the introduction of both smart FC architecture and paradigm. The IoT device performance and efficiency benefits from the implementation of forwarding capability algorithms. This framework ends cloud services to form a wider system that enables connected objects to communicate via wireless radio waves. Analysis and comparison of methodologies proved that researchers failed to dedicate sufficient work to reduce complete latency, which includes compute delay together with communication latency and network latency between IoTs and cloud. A new technique needs development to lower substantial latency amounts required for time-sensitive healthcare IoT applications.

The research of latency optimization for H-IoT systems in FC requires additional work to resolve multiple outstanding gaps before establishing improved solutions. The following essential deficiencies exist in latency optimization approaches for H-IoT platforms in FC context:

- Multiple strategies for latency reduction (such as architectural improvements and communication protocols and load balancing) remain unconnected through comprehensive overall frameworks within many existing approaches despite their focus on specific aspects (such as task scheduling or resource allocation).
- The analysis of latency optimization methods lacks confirmation in practical medical centers through extensive testing. Proposed solutions that scientists test in artificial conditions might fail to demonstrate accurate results in healthcare facilities with their real-world complexities and unpredictable scenarios.
- The process of enhancing performance through latency optimization leads to increased energy usage during execution. The H-IoT community requires additional research to develop strategies, which reduce latency yet maintain high energy efficiency to achieve both performance speed and sustainability.
- The performance of latency optimization methods remains unstable when healthcare systems grow because most solutions were designed primarily for small environments. The processing of expanding H-IoT systems

presents a major problem because it requires effective solutions to handle increasing devices and data streams.

- The enhancement methods for reducing latency need to ensure both security requirements and privacy standards since they should protect medically sensitive patient information. The integration between fast solution deployment systems and secure infrastructure remains a problem that researchers need to address.
- The healthcare environment operates in a dynamic state because it deals with fluctuating demands together with shifting operational conditions. A gap exists in current approaches because they fail to support immediate adaptation according to live network changes and device availability and data flow patterns.
- Different FC platforms and H-IoT devices work less effectively when standardized protocols and interoperability between these systems are absent. Standardization along with universal guidelines remains essential to achieve perfect performance and system integration.

9.1 Challenges and comparative analysis of techniques in IoT and cloud

In this part, the challenges and comparative analysis of approaches for latency minimization in IoT and cloud are covered in depth. The limits of the present approaches are recognized in the IoT-fog-cloud scenario. Besides, network, compute, network utilization, and communication latency are stated to be excessive and infeasible for healthcare IoTs. Excessive latency in healthcare IoTs causes delays in PHD delivery to end-users [1,6]. Table 3 lists the issues associated with the stated research activity.

Table 3: Challenges identification (Ci, *i* = 1, 2, 3 … *n*)

Reference	Challenges identification
[1,3,5,10,14]	C1: High network latency
[2,4,29,31]	C2: High communication latency
[5,18,32,73]	C3: High computation latency
[1,3,5,33,72,75]	C4: High data traffic
[8,13,22,35,46]	C5: High bandwidth consumption
[1,5,24,30]	C6: Large volume of data
[1,11,24,36,45,51]	C7: Real-time data transmission
[4,5,16,33,75]	C8: Load balancing and data replication
[1,3,5,7,12,18,69]	C9: High energy consumption
[1,1,20,31,42,51,60,73,75]	C10: Coordination between IoT, Fog, and cloud

Multiple problems appear during IoT integration with cloud computing because their foundational designs and functional needs remain incompatible. The main difficulty in this system is the time delay and data volume restrictions that occur when transferring large IoT device information to cloud platforms. The timing delays caused by this situation become dangerous to real-time healthcare operations because minor delays could cause major issues. The centralization of cloud systems creates performance bottlenecks which reduce operational speed and maintains longer processing delays. Sensitivity of data passing between IoT devices and cloud servers creates major security and privacy threats because these exchanges are vulnerable to cyber-attacks and security breaches. Large-scale IoT deployments face economic and sustainability challenges because of the high costs of cloud services and excessive energy usage by IoT devices. Technical examination of IoT procedures alongside cloud technology systems demonstrates how performance directly competes against operational expenses while affecting system scalability. Cloud computing performs data processing at high levels and serves applications that need extensive processing power. Edge and FC systems work as supplemental solutions for solving the downsides of cloud computing. Many organizations bring their computing power from cloud servers to nearer locations, which then decreases data latency as well as bandwidth consumption. Research teams develop resource allocation strategies and efficient task scheduling methods and edge-preprocessing approaches to enhance performance results. The selection of technique relies on particular application demands since there exists no standard solution. Research efforts focus on creating hybrid models, which unite the best elements of IoT technology and cloud systems and efficiently solve their identified limitations. The FC communication, network latency, and minimization are presented in Table 4.

Table 4: FC techniques for computation minimization, network latency, and communication

Technique ID	Details
T1	FC-based content-aware filtering and FC security service (FCSS) on edge networks. Enhances security and reduces latency by processing data closer to the source [90]
T2	Cost-effective deployment schema for IoT using FC. Optimizes resource allocation to minimize costs while maintaining performance [91]
T3	SPSRP for IoT and FC nodes. Dynamically allocates resources based on service demand to improve efficiency [92]
T4	Reinforcement learning (RL) algorithm and basic block offloading mechanism on fog mobile. Reduces computation load and latency by offloading tasks intelligently [32]
T5	Hermes: Mobile computing framework for efficient resource utilization. Focuses on minimizing latency and improving communication in mobile environments [42]
T6	Hipster: Heuristic and RL-based optimization in cloud computing. Balances workload and reduces latency through intelligent task scheduling [27]
T7	iFogStor: Heuristic approach for storage and computation in FC. Optimizes data storage and processing to minimize latency and resource usage [44]
T8	Future edge cloud, cloudlet, FC, and mobile edge computing. Integrates multiple edge technologies to reduce latency and improve communication [40]
T9	Computation offloading for multiple mobile users in a cloud computing environment. Reduces computation load on mobile devices by offloading tasks to the cloud [82]
T10	FC model for QoS deployment infrastructure in IoTs. Ensures reliable and low-latency communication for IoT applications [23]
T11	Cloud-fog-based services for healthcare applications. Combines cloud and FC to minimize latency and enhance real-time data processing in healthcare [93]
T12	A hybrid bio-inspired algorithm for optimizing computation and communication. Uses nature-inspired techniques to improve efficiency and reduce latency [98]
T13	A fog-centric cloud storage scheme for efficient data management. Reduces latency by storing and processing data closer to the edge [94]
T14	Integration of FC-based techniques with neural network (NN) algorithms. Enhances computation and communication efficiency through machine learning [45]
T15	FC analytical model and hybrid machine learning algorithm for healthcare IoTs. Proposes a novel approach to optimize computation and reduce latency in healthcare applications (Our study)

Table 4 shows complete details of different FC methods, which fight against key computational problems and network delay problems while optimizing communication performance. The distinctive nature of each technique specifically improves IoT and edge computing system efficiency within operational requirements that demand low-latency performance and optimized resource consumption in applications such as healthcare, real-time monitoring and mobile operations. The techniques T1 (FCSS) and T8 (edge cloud integration) work to decrease latency by localizing data processing near its source and also T4 (RL-based offloading) and T12 (bio-inspired algorithms) reduce computation time through intelligent distribution of tasks and resources. Advanced technologies including heuristic approaches (T7 and T6) along with machine learning (T14 and T15) are integrated in the table to enhance system adaptability and performance. The systematic organization of these techniques within the table provides researchers and practitioners with an essential reference when they want to implement efficient FC solutions. Table 4 demonstrates its significance through its presentation of the wide range of innovative FC strategies, which enhance healthcare IoT applications that need low latency. The continuous growth of IoT technology along with increasing system complexity creates challenges for traditional cloud models to meet requirements for real-time processing and minimal latency. This gap in performance is resolved through FC, which operates at the edge and the included techniques show effective ways to implement such solutions. The combination of T11 and T15 illustrates that FC technology provides vital advantages for healthcare application scenarios that require immediate responses because delays could lead to life-threatening situations. These techniques help develop IoT systems that are reliable and scalable through their solutions for energy efficiency issues and resource allocation problems and security concerns. The table compiles current advancements and indicates that further innovative development is necessary to fulfill modern IoT application requirements. The comparative analysis for minimization are presented in Table 5.

Table 5: Comparative analysis for minimization of (C_L), (C_{PL}), and (N_L)

Reference	T_i	(C_L)	(C_{PL})	(N_L)
[41]	T1	×		
[82]	T2	×		
[43]	T3	×		
[49]	T4	×	×	
[48]	T5	×		
[47]	T6		×	×
[36]	T7		×	×
[56]	T8			×
[60]	T9		×	
[76]	T10	×		
[83]	T11	×		×
[52]	T12	×	×	
[84]	T13		×	
[29]	T14		×	
Our proposed approach	T15	×	×	×

Different strategies, procedures, and their research efforts are used as benchmarks for comparison and analysis with the proposed approach. Among cloud and IoT solutions, the referenced methods aim to reduce network usage, excessive latency, and RAM utilization. These solutions employ a standard FC framework as middleware gateways for data transmission between end-users, IoT devices, and cloud servers. In healthcare IoT, the proposed systems play a critical role. However, most prior research (e.g., network latency, computation latency, and communication latency between cloud servers, IoT devices, and fog nodes) overlooked the practical application of total latency reduction. As a result, these existing approaches are used to contrast with the proposed study. As demonstrated in previous studies [29,45], middleware gateways and conventional cloud computing solutions failed to meet the QoS and latency requirements of healthcare IoT. To date, no comprehensive study on healthcare IoT has been conducted to reduce the end-to-end time delay between end-users, IoT devices, and the cloud. To address high latency, a novel hybrid fuzzy RL algorithm based on NN architecture is proposed, along with an analytical model. This study aims to minimize energy consumption, network usage, latency, and RAM utilization across healthcare IoT, cloud, and end-users. The proposed analytical model and algorithm meet the QoS requirements for healthcare IoT. A comprehensive comparative analysis of prior work on latency reduction in IoT can be found in Table 6.

9.2 Future directions and suggestions

Future research on low-latency Healthcare IoT and FC should concentrate on uniting advanced technologies such as 5G and beyond because they provide ultralow latency combined with high reliability. Healthcare IoT system performance improves substantially when healthcare organizations deploy 5G networks because these networks provide accelerated data transmission as well as instant device-to-device connectivity. The

Table 6: Comparative analysis of previous research works for minimization of latency in IoT

Authors / Year	Challenges	The proposed approach	Advantages	Limitations
Hassan et al., 2017 [73].	There is an issue how to send real-time data or information	They proposed networking modeling	The proposed approach was able to send in real-time modes, thereby reducing the packet loss	Minimizing the computation latency was not complete in this work
Brogi and Forti 2017 [76]	High bandwidth and high latency	They proposed model based on FogTorch and Java tool	The proposed approach was able to reduce high bandwidth and high latency	The high deployment cost associated with ensuring fog node reliability is not adequately addressed
Masri et al., 2017 [65]	High latency	They proposed Fog-to-Fog approach	The proposed approach was able to reduce the high latency	There is no validation for the proposed approach with real case study
Pang et al., 2017 [85]	High latency driven for scheduling process	They proposed Fog Radio access networks	Reduce the latency ratio	There is no explanation on how to reduce the high latency between fog nodes and wireless devices
Name et al., 2017 [63]	High response time and high latency	Algorithm for FC	Minimize the response time	The computation time was not considered
Ali et al., 2018 [86]	High latency	The cloudlet selection based on the problem optimization	The proposed clouldlet selection and IoT nodes reduce the latency ratio	They did not implement the network traffic stage
Yousefpour et al., 2019 [75]	How to identify the QoS requirements and low latency	They proposed Fogplan model for QoS requirements	The proposed model met the QoS requirements for latency issue	Fog services identification issues and VM migration were not considered
Kraemer et al., 2017 [80]	High latency and communication in healthcare applications	They proposed Fog gateway scheme	Tasks time in the Fog gateway was more than that of the cloud	Although the system involves large data transmission, the issue of high network latency is not discussed
Tuli et al. [29]	High latency	They proposed deep learning methods	They reduced service latency	Computation latency and high network
Shukla et al., 2023 [87]	Requires future research work for improving the latency minimization techniques and technologies	Presented some techniques and technologies for improving latency minimization	Determined the future research direction	Handle high-latency
Wu et al., 2023 [88]	Unmanned aerial vehicle (UAV)-enabled mobile edge computing (MEC) system	Algorithm can significantly reduce the system's computation latency compared to the benchmark schemes	Optimize the UAV's horizontal location, the UAV's altitude, and the offloading bandwidths and computing CPU frequencies, respectively	Minimize high latency
Ahmed et al., 2023 [89]	Enhance fuel efficiency and reduce traffic congestion	Decentralized platooning model	Minimizing latency and energy consumption	Mixed-integer linear programs formulation captures
Wang et al., 2024 [90]	Reduced processing time	Whitetail deer farmers of Ohio-VANET, improves energy consumption and minimizes the communication costs of VANET	Created a probability-based hybrid Whale -Dragonfly Optimization (p–H-WDFOA) edge computing model for smart urban vehicle transportation	Handle requests from vehicles in a shorter amount of time

(*Continued*)

Table 6: *Continued*

Authors / Year	Challenges	The proposed approach	Advantages	Limitations
Adhistian and Wibowo 2024 [91]	Strategy of QoS class identifiers 3 (QCI 3) has greatly improved user experience by reducing average latency and jitter	Proposed optimization strategy effectively enhances the user experience by significantly reducing average latency and jitter	Enhance the user experience by reducing average latency and latency variations (jitter)	Limited QCIs (QoS class identifiers) restrict the handling different service types with varying quality requirements
Wang et al., 2024 [92]	Trade-off in inference accuracy	Latency minimized parallel inference on CPU (L-PIC) – a framework	Reduce the inference latency of multi-DNN	IoT edge devices rely on CPUs.
Patel and Choudhury 2025 [93]	Effective latency reduction strategies that enable cloud-based networks	Techniques aimed at minimizing latency in cloud networking infrastructures	Use of CDN for caching	
Lai et al., 2023 [94]	Determine the blocklength and the size of subtasks	Transmit-while-compute scheme to reduce the average latency	Very close to the computing time of the task	Partial offloading
Su et al., 2023 [95]	Existing HFL research works are insufficient to tackle	A multi-employee self service framework	Reduce the cumulative computation and communication time	Extensive experiments on real-world data sets
Siefert et al., 2023 [96]	Accessibility of benchmark data, system heterogeneity, inter-node performance	Providing a collection of latency and bandwidth microbenchmark results for US Department of Energy systems ranked above 150 in June 2023	Comprehensive benchmarking, focus on node-level performance, use of established benchmarks	Limited Scope, intra-node focus, reproducibility challenges, outdated results
Tang et al., 2023 [97]	Network latency strategic manipulation peer churn and transaction activity	Strategic peering (Peri Algorithm) hybrid approach theoretical model	Effective latency reduction and cost-effective	Knowledge of network topology computational hardness
Zheng et al., 2023 [98]	Energy and latency constraints	Wireless power transfer (WPT) and MEC integration	Latency reduction energy efficiency	Limited battery capacity dependence on channel conditions
Zhang et al., 2023 [99]	Latency minimization privacy concerns	Multi-objective optimization block coordinate descent framework	Improved spectral efficiency low latency	Complexity of implementation
Agrawal et al., 2024 [100]	Low compute utilization in decodes throughput-latency tradeoff	Stall-free scheduling	Scalability, flexibility, improved throughput, and latency	Complexity in token budget selection, limited feature parity
Desai and Patil 2023 [101]	Traffic spikes and latency	RL and large language models (LLMs)	Improved latency and throughput	Complexity in model deployment
Hu et al., 2024 [102]	Large data volume collaboration security	Collaborative perception framework	Improved perception accuracy and reduced latency	Dependency on communication infrastructure
Warrier et al., 2024 [103]	High latency and service disruption	Graph theory-based handover algorithm	Improved connectivity, reduced latency, and enhanced QoS	Complexity in implementation of hardware requirements
Oliveira et al., 2024 [104]	Dynamic and heterogeneous environment latency and communication costs	Minimizing response time and reducing network traffic	Improved latency reduced network traffic	Dependency on network topology, no privacy considerations
Agrawal and Gupta 2024 [105]	Latency and bandwidth trade-off	Latency-optimized request allocation algorithm	Cost efficiency, latency guarantee	Static bandwidth and latency requirements

development of 6G networks which anticipate substantial bandwidth and minimal latency features will produce transformative healthcare applications. Research groups must assess how quantum computing technology optimizes edge computational procedures to speed up essential medical operations. Intelligent algorithms including federated learning and edge AI enable distributed decision-making structures which allow time-critical healthcare tasks to run efficiently without centralizing cloud resources.

The future development of FC in healthcare IoT requires standardization efforts for both protocols and frameworks. Standardized approaches are missing today because they limit both interoperability and scalability thus blocking general healthcare industry adoption. The development of future analysis needs to build standardized frameworks which link FC solutions to healthcare infrastructure and maintain full compliance with regulatory provisions including health insurance portability and accountability act and general data protection regulation. To validate their effectiveness in real-world environments, researchers require testing through digital twin and simulation platforms that operate at a high level of robustness. Research must identify ethical aspects of improved latency performance specifically related to patient data security and protection from unauthorized exposure. International cooperation between healthcare workers along with computer scientists and network engineers represents the crucial pathway for addressing distinct FC and healthcare IoT latency optimization requirements. The summary of future directions with technical suggestions are as follows:

1. **Advanced networks**
   - 5G/6G Integration provides healthcare applications with real-time capabilities through low latency systems that offer up to 1 ms latency along with reliable network connections, which support applications such as remote surgery and telemedicine and continuous patient monitoring. Research work in the coming years must determine the potential benefits of 6G networks because they will deliver superior speed levels using terahertz frequencies and automated network management systems controlled by AI. These technological advancements will establish perfect data transmission channels between IoT devices and fog nodes thereby delivering prompt data processing services.
   - Quantum algorithms employ quantum computing to process large edge-based healthcare analytics data in real time. Quantum computing strengthens encryption security by developing new methods for secure and fast communications between devices. Quantum key distribution serves as a security method that protects the data transfer between devices and fog nodes.
   - The 5G infrastructure supports network slicing which produces specific virtual networks for healthcare operations through dedicated 5G infrastructure that provides both low-latency performance and high reliability. The application-specific design of network slicing enables proper configuration for critical systems such as emergency response and remote diagnostics to expedite the delivery of important data.
2. **Intelligent algorithms**
   - Federated learning: This allows multiple edge devices to jointly train the malfunctioning software without the need to exchange raw data, thus not compromising patient privacy nor increasing latency in the process. Federated learning is especially beneficial for applications such as predictive diagnostics, where incorporating multiple sources of data enhances model performance while avoiding the risks of centralizing sensitive data.
   - Edge AI: Adopting low-intensive AI models on fog nodes. This process enables real-time decision-making without the need to send data into the cloud. AI models, for instance, can identify anomalies in patient vitals or initiate automated emergency responses, enabling prompt interventions.
   - Predictive analytics: Machine learning algorithms can forecast network congestion and dynamically allocate resources or offload tasks to reduce latency. Predictive analytics, for example, can proactively route data along less congested network paths or dynamically shift computational loads based on real-time traffic patterns, resource availability, and latency forecasts.
3. **Robust testing environments**
   - Researchers create digital twins of healthcare IoT systems for implementing latency optimization technique analysis through controlled simulation and testing systems. The use of digital twins helps to recognize possible system blockages, which enables testing solutions before actual implementation thus decreasing the need to fix problems after deployment.

- Research groups should use NS-3 and OMNeT++ simulation frameworks for evaluating the operational characteristics of FC structures under both high data throughput and network failure scenarios. System behavior becomes more conspicuous through these tools, which also enables an optimization of latency measurements.
- The implementation of real-world trials in hospitals together with clinics gives researchers a chance to validate their latency optimization strategies under authentic clinical conditions. Practical information from pilots demonstrates the features and implementation scalability of solutions so health services and their patients receive solutions that work effectively.

4. **Ethical and security considerations**
   - Real-time healthcare data processing remains private through homomorphic encryption protocols that execute advanced encryption methods for edge-based operations. Homomorphic encryption allows operations to take place on encrypted medical data, which protects confidentiality.
   - Implementing a zero-trust security architecture requires that every device and user be authenticated and authorized before gaining access to the network. Allowing only trusted entities to participate ensures low system latency while significantly reducing the risk of data breaches.
   - The healthcare IoT system benefits from blockchain technology to maintain secure and decentralized transaction logging mechanisms and data exchange tracking. Blockchain technology enables transparent data protection along with the capability to track transactions, which lowers the opportunity for unapproved manipulation of system data.

10 Conclusion

A review on latency reduction in healthcare IoT and cloud computing has been conducted. The limitations of the current systems and technologies have been identified, revealing that several performance criteria remain unaddressed. Additionally, processing delays, network latency, and communication latency were found to be significant and often impractical for real-world applications. As a result, various latency reduction approaches and technologies have been explored. By reviewing systems, tools, challenges, methodologies, and technologies, this study aims to bridge the existing gap. This study delves into the systems and technologies used in IoT and cloud computing to reduce latency. Furthermore, the findings are expected to serve as a foundation for future research. This study intentionally selected and thoroughly analyzed relevant articles for the review, synthesizing data from these articles to address a set of research questions outlined in this study, which guided this investigation. The objective of this study is to provide an in-depth examination of the latency issue in IoT and cloud computing, as well as to propose a novel solution and compare it with other latency reduction strategies currently in use. The goal is to encourage future research toward developing a latency-aware model for healthcare IoT. The contributions of this study can be summarized as follows:

1. **Classification of technologies and approaches:** A systematic categorization of the technologies and approaches discussed in the literature.
2. **Quantitative analysis:** Evaluation of the volume of distributed works analyzed and search trends in Google Scholar to identify patterns in latency reduction techniques and technologies.
3. **Review of latency reduction strategies:** A comprehensive review of various latency reduction strategies and technologies.
4. **Identification of research gaps:** Recognition of research needs in the areas of IoT, FC, and cloud latency reduction.
5. **Addressing limitations:** Highlighting the limitations of existing studies and unresolved challenges related to latency requirements for time-sensitive applications.

From this review, researchers and industry professionals will gain a clearer understanding of IoT and cloud computing requirements for time-sensitive applications, along with a better grasp of latency reduction approaches and technologies. The investigation presents different techniques which use FC to decrease healthcare IoT system latencies. The time it takes to process and respond to information remains a decisive aspect of healthcare applications when they need to provide effective patient care through remote monitoring and emergency situations and telemedicine systems. Cloud computing proves inappropriate for time-sensitive healthcare applications because it presents two major limitations through high latency together with bandwidth constraints that impact its performance. The authors introduce FC as a potential solution because it positions computing and storage near network edges to solve current limitations. The analysis divides latency optimization strategies into four main sections which encompass resource distribution along with task delegation and information compression and edge storage functions. Researchers analyze the approaches through two sets of metrics including effectiveness and implementation complexity as well as suitability levels across healthcare conditions. The study explores how machine learning alongside artificial intelligence enhances real-time latency prediction and management. Predictive analytics models allow organizations to distribute resources in advance when they anticipate upcoming demands. The authors recognize that hybrid fog-cloud model selection plays an essential role when designing network architecture since it allows operators to maintain system efficiency while reducing latency times. The research identifies security and privacy requirements for healthcare IoT before discussing ways to protect patient confidentiality while optimizing latency. This review suggests several directions for future research such as standardizing protocols and examining 5G networks and building better facilities for validating these methods in genuine healthcare settings. This study presents extensive details about recent approaches for healthcare IoT latency reduction together with an emphasis on how FC supports quick and effective medical care systems.

Funding information: Authors state no funding involved.

Author contributions: Zahraa A. Jaaz and Mohd Ezanee Bin Rusli: writing – review and editing, validation, methodology, and conceptualization. Nur Azzammudin Rahmat and Maythem Kamal Abbas Al-Adilee: writing – review and editing, supervision, project administration, methodology, and conceptualization.

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

Data availability statement: No data were used for the research described in the article.

References

[1] Rani M, Guleria K, Panda SN. Enhancing performance and latency optimization in fog computing with a smart job scheduling approach. Stat Optim Inf Comput. 2025;13(1):309–30. doi: 10.19139/soic-2310-5070-2141.

[2] Manthiramoorthy C, Khan KMS, Ameen NA. Comparing several encrypted cloud storage platforms. Int J Math Stat Computer Sci. 2023;2:44–62. doi: 10.59543/ijmscs.v2i.7971.

[3] Alatoun K, Matrouk K, Mohammed MA, Nedoma J, Martinek R, Zmij P. A novel low-latency and energy-efficient task scheduling framework for internet of medical things in an edge fog cloud system. Sensors. 2022;22(14):5327. doi: 10.3390/s22145327.

[4] Choppara P, Lokesh B. Efficient task scheduling and load balancing in fog computing for crucial healthcare through deep reinforcement learning. IEEE Access. 2025. doi: 10.1109/ACCESS.2025.3539336.

[5] Chuan WC, Manickam S, Ashraf E, Karuppayah S. Challenges and opportunities in fog computing scheduling: A literature review. IEEE Access. 2025;13:14702–26. doi: 10.1109/ACCESS.2024.3525261.

[6] Bhasker B, Kaliraj S, Gobinath C, Sivakumar V. Optimizing energy task offloading technique using IoMT cloud in healthcare applications. J Cloud Comput. 2025;14(1):9. doi: 10.1186/s13677-025-00733-0.

[7] Kang Y. Presenting a model in smart electronic health networks based on IoT-Fog for health care to optimize resources. Electr Eng. 2025;107(1):1125–40. doi: 10.1007/s00202-024-02575-6.

[8] Namratha P, Chaganti KR, Elicherla SLR, Guddati S, Swarna A, Reddy PT. Optimizing latency and communication in federated edge computing with LAFEO and gradient compression for real-time edge analytics. In 2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI). IEEE; Jan 2025. p. 608–13. doi: 10.1109/ICMCSI64620.2025.10883220.

[9] Choppara P, Mangalampalli SS. Resource adaptive automated task scheduling using deep deterministic policy gradient in fog computing. IEEE Access. 2025;13:25969–94. doi: 10.1109/ACCESS.2025.3539606.

[10] Hosseinioun P, Kheirabadi M, Kamel Tabbakh SR, Ghaemi R. aTask scheduling approaches in fog computing: A survey. Trans Emerg Telecommun Technol. 2022;33(3):e3792. doi: 10.1002/ett.3792.

[11] Amzil A, Abid M, Hanini M, Zaaloul A, El Kafhali S. Stochastic analysis of fog computing and machine learning for scalable low-latency healthcare monitoring. Clust Comput. 2024;27(5):6097–117. doi: 10.1007/s10586-024-04285-x.

[12] Mahale A, Geetha G. Enhancing decision-making in healthcare with fog computing for low-latency data processing. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE; June 2024. p. 1–7. doi: 10.1109/ICCCNT61001.2024.10725102.

[13] Kaliyaperumal K. Adaptive heuristic edge assisted fog computing design for healthcare data optimization. J Cloud Comput. 2024;13(1):1–18. doi: 10.1504/IJCC.2024.136277.

[14] Goswami A, Modi K, Patel C. Latency aware adaptive ant colony algorithm for service placement for healthcare fog. SN Computer Sci. 2024;5(8):1–8. doi: 10.1007/s42979-024-03524-7.

[15] Ala'anzy MA, Zhanuzak R, Akhmedov R, Mohamed N, Al-Jaroodi J. Dynamic load balancing for enhanced network performance in IoT-enabled smart healthcare with fog computing. IEEE Access. 2024;12:188957–75. doi: 10.1109/ACCESS.2024.3516362.

[16] Baskar R, Mohanraj E, Sneka T, Yazhini S, Vasanth S. Teaching learning-based optimization for medical iot applications service placement in fog computing. In 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). IEEE; Feb 2024. p. 1–6. doi: 10.1109/SCEECS61402.2024.10482092.

[17] Wen B, Li S, Motevalli H. Exploitation of healthcare IoT–fog-based smart e-health gateways: A resource optimization approach. Clust Comput. 2024;27(8):10733–55. doi: 10.1007/s10586-024-04502-7.

[18] Jeyaraman N, Jeyaraman M, Yadav S, Ramasubramanian S, Balaji S, Muthu S, et al. Applications of fog computing in healthcare. Cureus. 2024;16(7):e64263. doi: 10.7759/cureus.64263.

[19] Hossam HS, Abdel-Galil H, Belal M. An energy-aware module placement strategy in fog-based healthcare monitoring systems. Clust Comput. 2024;27(6):7351–72. doi: 10.1007/s10586-024-04308-7.

[20] Duggal S, Kaur P. Optimizing healthcare through fog computing: Simulating and performance evaluation of real-time health monitoring systems. In 2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON). IEEE; Nov 2024. p. 1–7. doi: 10.1109/DELCON64804.2024.10866954.

[21] Sathyanarayana N, Raufi AM, Sharma M. Health-FoTs–A latency aware fog based IoT environment and efficient monitoring of body's vital parameters in smart health care environment. J Smart Internet Things (JSIoT). 2024;2024(2):26–41. doi: 10.2478/jsiot-2024-0010.

[22] Du H, Liu M, Liu N, Li D, Li W, Xu L. Scheduling of low-latency medical services in healthcare cloud with deep reinforcement learning. Tsinghua Sci Technol. 2024;30(1):100–11. doi: 10.26599/TST.2024.9010033.

[23] Tripathy SS, Bebortta S, Chowdhary CL, Mukherjee T, Kim S, Shafi J, et al. FedHealthFog: A federated learning-enabled approach towards healthcare analytics over fog computing platform. Heliyon. 2024;10(5):e26416. doi: 10.1016/j.heliyon.2024.e26416.

[24] Kaur N, Mittal A, Lilhore UK, Simaiya S, Dalal S, Saleem K, et al. Securing fog computing in healthcare with a zero-trust approach and blockchain. EURASIP J Wirel Commun Netw. 2025;2025(1):5. doi: 10.1186/s13638-025-02431-6.

[25] Moustafa N. A systemic IoT-fog-cloud architecture for big-data analytics and cyber security systems: A review of fog computing. arXiv preprint arXiv:190601055. 2019;10:1–17.

[26] Nandyala CS, Kim HK. From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals. Int J Smart Home. 2016;10(2):187–96. doi: 10.14257/ijsh.2016.10.2.18.

[27] Ijaz M, Li G, Lin L, Cheikhrouhou O, Hamam H, Noor A. Integration and applications of fog computing and cloud computing based on the internet of things for provision of healthcare services at home. Electronics. 2021;10(9):1077. doi: 10.3390/electronics10091077.

[28] Natesha BV, Guddeti RMR. Adopting elitism-based Genetic Algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment. J Netw Computer Appl. 2021;178:102972. doi: 10.1016/j.jnca.2020.102972.

[29] Tuli S, Basumatary N, Gill SS, Kahani M, Arya RC, Wander GS, et al. HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. Future Gener Computer Syst. 2020;104:187–200. doi: 10.1016/j.future.2019.10.043.

[30] Islam MSU, Kumar A, Hu YC. Context-aware scheduling in Fog computing: A survey, taxonomy, challenges and future directions. J Netw Computer Appl. 2021;180:103008. doi: 10.1016/j.jnca.2021.103008.

[31] Kaur M, Aron R. A systematic study of load balancing approaches in the fog computing environment. J Supercomputing. 2021;77(8):9202–47. doi: 10.1007/s11227-020-03600-8.

[32] Li K, Wang L. Elastic scheduling of virtual machines in cloudlet networks. In 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE; Oct 2021. p. 1–7. doi: 10.1109/IPCCC51483.2021.9679429.

[33] Darabkh KA, Alkhader BZ. Fog computing-and software defined network-based routing protocol for vehicular Ad-hoc network. In 2022 International Conference on Information Networking (ICOIN). IEEE; Jan 2022. p. 502–6. doi: 10.1109/ICOIN53446.2022.9687147.

[34] Aazam M, Huh EN. Fog computing and smart gateway based communication for cloud of things. In 2014 International Conference on Future Internet of Things and Cloud. IEEE; Aug 2014. p. 464–70. doi: 10.1109/FiCloud.2014.83.

[35] Rahmani AM, Gia TN, Negash B, Anzanpour A, Azimi I, Jiang M, et al. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Future Gener Computer Syst. 2018;78:641–58. doi: 10.1016/j.future.2017.02.014.

[36] Naas MI, Parvedy PR, Boukhobza J, Lemarchand L. iFogStor: An IoT data placement strategy for fog infrastructure. In 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC). IEEE; May 2017. p. 97–104. doi: 10.1109/ICFEC.2017.15.

[37] Shukla S, Hassan M, Tran DC, Akbar R, Paputungan IV, Khan MK. Improving latency in Internet-of-Things and cloud computing for real-time data transmission: A systematic literature review (SLR). Clust Comput. 2023;26:2657–80. doi: 10.1007/s10586-021-03279-3.

[38] You D, Doan TV, Torre R, Mehrabi M, Kropp A, Nguyen V, et al. Fog computing as an enabler for immersive media: Service scenarios and research opportunities. IEEE Access. 2019;7:65797–810. doi: 10.1109/ACCESS.2019.2917291.

[39] Osanaiye O, Chen S, Yan Z, Lu R, Choo KKR, Dlodlo M. From cloud to fog computing: A review and a conceptual live VM migration framework. IEEE Access. 2017;5:8284–300. doi: 10.1109/ACCESS.2017.2692960.

[40] Chiti F, Fantacci R, Picano B. A matching theory framework for tasks offloading in fog computing for IoT systems. IEEE Internet Things J. 2018;5(6):5089–96. doi: 10.1109/JIOT.2018.2871251.

[41] Wu J, Dong M, Ota K, Li J, Guan Z. FCSS: Fog-computing-based content-aware filtering for security services in information-centric social networks. IEEE Trans Emerg Top Comput. 2017;7(4):553–64. doi: 10.1109/TETC.2017.2747158.

[42] Yadav AM, Sharma SC, Tripathi KN. A two-step technique for effective scheduling in cloud–fog computing paradigm. In advances in computational intelligence and communication technology. Singapore: Springer; 2021. p. 367–79.

[43] Li G, Wu J, Li J, Wang K, Ye T. Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things. IEEE Trans Ind Inform. 2018;14(10):4702–11. doi: 10.1109/TII.2018.2845844.

[44] Haghgoo M, Dognini A, Monti A. A cloud-based platform for service restoration in active distribution grids. IEEE Trans Ind Appl. 2022;58(2):1554–63. doi: 10.1109/TIA.2022.3142661.

[45] Almaiah MA, Ali A, Hajjej F, Pasha MF, Alohali MA. A lightweight hybrid deep learning privacy preserving model for FC-based industrial internet of medical things. Sensors. 2022;22(6):2112. doi: 10.3390/s22062112.

[46] Lakhan A, Mohammed MA, Obaid OI, Chakraborty C, Abdulkareem KH, Kadry S. Efficient deep-reinforcement learning aware resource allocation in SDN-enabled fog paradigm. Autom Softw Eng. 2022;29(1):1–25. doi: 10.1007/s10515-021-00318-6.

[47] Nishtala R, Carpenter P, Petrucci V, Martorell X. Hipster: Hybrid task manager for latency-critical cloud workloads. In 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE; Feb 2017. p. 409–20. doi: 10.1109/HPCA.2017.13.

[48] Kao YH, Krishnamachari B, Ra MR, Bai F. Hermes: Latency optimal task assignment for resource-constrained mobile computing. IEEE Trans Mobile Computing. 2017;16(11):3056–69. doi: 10.1109/TMC.2017.2679712.

[49] Alam MGR, Tun YK, Hong CS. Multi-agent and reinforcement learning based code offloading in mobile fog. In 2016 International Conference on Information Networking (ICOIN). IEEE; Jan 2016. p. 285–90. doi: 10.1109/ICOIN.2016.7427078.

[50] Waqar A, Raza A, Abbas H, Khan MK. A framework for preservation of cloud users' data privacy using dynamic reconstruction of metadata. J Netw Computer Appl. 2013;36(1):235–48. doi: 10.1016/j.jnca.2012.09.001.

[51] Soleymani SA, Abdullah AH, Zareei M, Anisi MH, Vargas-Rosales C, Khan MK, et al. A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing. IEEE Access. 2017;5:15619–29. doi: 10.1109/ACCESS.2017.2733225.

[52] Rafique H, Shah MA, Islam SU, Maqsood T, Khan S, Maple C. A novel bio-inspired hybrid algorithm (NBIHA) for efficient resource management in fog computing. IEEE Access. 2019;7:115760–73. doi: 10.1109/ACCESS.2019.2924958.

[53] Grinnemo KJ, Brunstrom A. A first study on using MPTCP to reduce latency for cloud based mobile applications. In 2015 IEEE symposium on computers and communication (ISCC). IEEE; July 2015. p. 64–9. doi: 10.1109/ISCC.2015.7405495.

[54] Sambyo K, Bhunia CT. Application of multi level ATM in reducing latency in clouds for performance improvement of integrated voice, video and data services. In 2014 11th International Conference on Information Technology: New Generations. IEEE; April 2014. p. 607–7. doi: 10.1109/ITNG.2014.27.

[55] Eccles MJ, Evans DJ, Beaumont AJ. True real-time change data capture with web service database encapsulation. In 2010 6th World Congress on Services. IEEE; July 2010. p. 128–31. doi: 10.1109/SERVICES.2010.59.

[56] Pan J, McElhannon J. Future edge cloud and edge computing for internet of things applications. IEEE Internet Things J. 2017;5(1):439–49. doi: 10.1109/JIOT.2017.2767608.

[57] Habak K, Ammar M, Harras KA, Zegura E. Femto clouds: Leveraging mobile devices to provide cloud service at the edge. In 2015 IEEE 8th International Conference on Cloud Computing. IEEE; June 2015. p. 9–16. doi: 10.1109/CLOUD.2015.12.

[58] Sajithabanu S, Balasundaram SR. Cloud based content delivery network using genetic optimization algorithm for storage cost. In 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE; Nov 2016. p. 1–6. doi: 10.1109/ANTS.2016.7947822.

[59] Lee M, Kim Y, Lee Y. A home cloud-based home network auto-configuration using SDN. In 2015 IEEE 12th International Conference on Networking, Sensing and Control. IEEE; April 2015. p. 444–9. doi: 10.1109/ICNSC.2015.7116078.

[60] Cao H, Cai J. Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach. IEEE Trans Veh Technol. 2017;67(1):752–64. doi: 10.1109/TVT.2017.2740724.

[61] Cavalcante E, Pereira J, Alves MP, Maia P, Moura R, Batista T, et al. On the interplay of Internet of Things and Cloud Computing: A systematic mapping study. Computer Commun. 2016;89:17–33. doi: 10.1016/j.comcom.2016.03.012.

[62] Liu Y, Fieldsend JE, Min G. A framework of fog computing: Architecture, challenges, and optimization. IEEE Access. 2017;5:25445–54. doi: 10.1109/ACCESS.2017.2766923.

[63] Name HAM, Oladipo FO, Ariwa E. User mobility and resource scheduling and management in fog computing to support IoT devices. In 2017 Seventh International Conference on Innovative Computing Technology (INTECH). IEEE; Aug 2017. p. 191–6. doi: 10.1109/INTECH.2017.8102447.

[64] Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutor. 2015;17(4):2347–76. doi: 10.1109/COMST.2015.2444095.

[65] Masri W, Al Ridhawi I, Mostafa N, Pourghomi P. Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication. In 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE; July 2017. p. 1005–10. doi: 10.1109/ICUFN.2017.7993950.

[66] Meng X, Wang W, Zhang Z. Delay-constrained hybrid computation offloading with cloud and fog computing. IEEE Access. 2017;5:21355–67. doi: 10.1109/ACCESS.2017.2748140.

[67] Naha RK, Garg S, Georgakopoulos D, Jayaraman PP, Gao L, Xiang Y, et al. Fog computing: Survey of trends, architectures, requirements, and research directions. IEEE Access. 2018;6:47980–8009. doi: 10.1109/ACCESS.2018.2866491.

[68] Yousefpour A, Ishigaki G, Jue JP. Fog computing: Towards minimizing delay in the internet of things. In 2017 IEEE International Conference on Edge Computing (EDGE). IEEE; June 2017. p. 17–24. doi: 10.1109/IEEE.EDGE.2017.12.

[69] Mukherjee M, Shu L, Wang D. Survey of fog computing: Fundamental, network applications, and research challenges. IEEE Commun Surv Tutor. 2018;20(3):1826–57. doi: 10.1109/COMST.2018.2814571.

[70] Mouradian C, Naboulsi D, Yangui S, Glitho RH, Morrow MJ, Polakos PA. A comprehensive survey on fog computing: State-of-the-art and research challenges. IEEE Commun Surv Tutor. 2017;20(1):416–64. doi: 10.1109/COMST.2017.2771153.

[71] Li J, Jin J, Yuan D, Zhang H. Virtual fog: A virtualization enabled fog computing framework for Internet of Things. IEEE Internet Things J. 2017;5(1):121–31. doi: 10.1109/JIOT.2017.2774286.

[72] Tuli S, Mahmud R, Tuli S, Buyya R. Fogbus: A blockchain-based lightweight framework for edge and fog computing. J Syst Softw. 2019;154:22–36. doi: 10.1016/j.jss.2019.04.050.

[73] Hassan MM, Lin K, Yue X, Wan J. A multimedia healthcare data sharing approach through cloud-based body area network. Future Gener Computer Syst. 2017;66:48–58. doi: 10.1016/j.future.2015.12.016.

[74] Hossain MS, Muhammad G. Cloud-assisted industrial internet of things (IIoT)–enabled framework for health monitoring. Computer Netw. 2016;101:192–202. doi: 10.1016/j.comnet.2016.01.009.

[75] Yousefpour A, Patil A, Ishigaki G, Kim I, Wang X, Cankaya HC, et al. FOGPLAN: A lightweight QoS-aware dynamic fog service provisioning framework. IEEE Internet Things J. 2019;6(3):5080–96. doi: 10.1109/JIOT.2019.2896311.

[76] Brogi A, Forti S. QoS-aware deployment of IoT applications through the fog. IEEE Internet Things J. 2017;4(5):1185–92. doi: 10.1109/JIOT.2017.2701408.

[77] Skorin-Kapov L, Matijasevic M. Analysis of QoS requirements for e-health services and mapping to evolved packet system QoS classes. Int J Telemed Appl. 2010;2010:628086. doi: 10.1155/2010/628086.

[78] Baker SB, Xiang W, Atkinson I. Internet of things for smart healthcare: Technologies, challenges, and opportunities. IEEE Access. 2017;5:26521–44. doi: 10.1109/ACCESS.2017.2775180.

[79] Margariti SV, Dimakopoulos VV, Tsoumanis G. Modeling and simulation tools for fog computing—a comprehensive survey from a cost perspective. Future Internet. 2020;12(5):89. doi: 10.3390/fi12050089.

[80] Kraemer FA, Braten AE, Tamkittikhun N, Palma D. Fog computing in healthcare–A review and discussion. IEEE Access. 2017;5:9206–22. doi: 10.1109/ACCESS.2017.2704100.

[81] Cheng B, Fuerst J, Solmaz G, Sanada T. Fog function: Serverless fog computing for data intensive IoT services. In 2019 IEEE International Conference on Services Computing (SCC). IEEE; July 2019. p. 28–35. arXiv:1907.08278v1.

[82] Dinh NT, Kim Y. An efficient availability guaranteed deployment scheme for IoT service chains over fog-core cloud networks. Sensors. 2018;18(11):3970. doi: 10.3390/s18113970.

[83] Mahmud R, Koch FL, Buyya R. Cloud-fog interoperability in IoT-enabled healthcare solutions. In Proceedings of the 19th International Conference on Distributed Computing and Networking; Jan 2018. p. 1–10. doi: 10.1145/3154273.3154347.

[84] Ahsan MM, Ali I, Imran M, Idris MYI, Khan S, Khan A. A fog-centric secure cloud storage scheme. IEEE Trans Sustain Comput. 2019;7(2):250–62. doi: 10.1109/TSUSC.2019.2914954.

[85] Pang AC, Chung WH, Chiu TC, Zhang J. Latency-driven cooperative task computing in multi-user fog-radio access networks. In 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE; June 2017. p. 615–24.

[86] Ali M, Riaz N, Ashraf MI, Qaisar S, Naeem M. Joint cloudlet selection and latency minimization in fog networks. IEEE Trans Ind Inform. 2018;14(9):4055–63. doi: 10.1109/TII.2018.2829751.

[87] Shukla S, Hassan MF, Tran DC, Akbar R, Paputungan IV, Khan MK. Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR). Clust Comput. 2023;26:1–24. doi: 10.1007/s10586-021-03279-3.

[88] Wu Q, Cui M, Zhang G, Wang F, Wu Q, Chu X. Latency minimization for UAV-enabled URLLC-based mobile edge computing systems. IEEE Trans Wirel Commun. 2023;23(4):3298–311. doi: 10.1109/TWC.2023.3307154.

[89] Ahmed EQ, Ameen ZH, Al-Mukhtar FS, Jaaz ZA. Maximizing mobile communication efficiency with smart antenna systems using beam forming and DOA algorithms. 2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI). IEEE; 2023. doi: 10.1109/EECSI59885.2023.10295948.

[90] Wang M, Mao J, Zhao W, Han X, Li M, Liao C, et al. Smart city transportation: A VANET edge computing model to minimize latency and delay utilizing 5G network. J Grid Comput. 2024;22(1):25. doi: 10.1007/s10723-024-09747-5.

[91] Adhistian P, Wibowo P. QCI optimization to minimize latency and enhance user experience. J Nas Tek Elektro. 2024;13(2):83. doi: 10.25077/jnte.v13n2.1193.2024.

[92] Wang T, Shi T, Liu X, Wang J, Liu B, Li Y, et al. Minimizing latency for multi-DNN inference on resource-limited CPU-only edge devices. In IEEE INFOCOM 2024-IEEE Conference on Computer Communications. IEEE; May 2024. p. 2239–48. doi: 10.1109/INFOCOM52122.2024.10621120.

[93] Patel N, Choudhury L. Techniques for reducing latency in cloud-based networks: A comprehensive study. Balt Multidiscip Res Lett J. 2025;2(1):47–56. doi: 10.1109/TWC.2023.3307154.

[94] Lai X, Jiang H, Bhunia S, Tran H. Reducing latency in MEC networks with short-packet communications. IEEE Trans Veh Technol. 2023;73(2):3000–4. doi: 10.1109/TVT.2023.3320578.

[95] Su L, Zhou R, Wang N, Chen J, Li Z. Low-latency hierarchical federated learning in wireless edge networks. IEEE Internet Things J. 2023;11(4):6943–60. doi: 10.1109/JIOT.2023.3314743.

[96] Siefert CM, Pearson C, Olivier SL, Prokopenko A, Hu J, Fuller TJ. Latency and bandwidth microbenchmarks of US department of energy systems in the June 2023 top 500 list. In Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis; Nov 2023. p. 1298–305. doi: 10.1109/CLUSTERWorkshops61457.2023.00021.

[97] Tang W, Kiffer L, Fanti G, Juels A. Strategic latency reduction in blockchain peer-to-peer networks. Proc ACM Meas Anal Comput Syst. 2023;7(2):1–33. doi: 10.1145/3589976.

[98] Zheng X, Zhu F, Xia J, Gao C, Cui T, Lai S. Intelligent computing for WPT–MEC-aided multi-source data stream. EURASIP J Adv Signal Process. 2023;2023(1):52. doi: 10.1186/s13634-023-01006-1.

[99] Zhang S, Zhang S, Yuan W, Li Y, Hanzo L. Efficient rate-splitting multiple access for the Internet of Vehicles: Federated edge learning and latency minimization. IEEE J Sel Areas Commun. 2023;41(5):1468–83. doi: 10.1109/JSAC.2023.3240716.

[100] Agrawal A, Kedia N, Panwar A, Mohan J, Kwatra N, Gulavani B, et al. Taming {Throughput-Latency} tradeoff in {LLM} inference with {Sarathi-Serve}. In 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24); 2024. p. 117–34. arXiv:2403.02310.

[101] Desai B, Patil K. Reinforcement learning-based load balancing with large language models and edge intelligence for dynamic cloud environments. J Innov Technol. 2023;6(1):1–13. doi: 10.1016/j.comcom.2024.04.009.

[102] Hu S, Fang Z, Deng Y, Chen X, Fang Y. Collaborative perception for connected and autonomous driving: Challenges, possible solutions and opportunities. arXiv preprint arXiv:240101544. 2024;7:1–33.

[103] Warrier A, Aljaburi L, Whitworth H, Al-Rubaye S, Tsourdos A. Future 6G communications powering vertical handover in non-terrestrial networks. IEEE Access. 2024;12:33016–34. doi: 10.1109/ACCESS.2024.3371906.

[104] Oliveira LT, Bittencourt LF, Genez TA, de Lara E, Peixoto ML. Enhancing modular application placement in a hierarchical fog computing: A latency and communication cost-sensitive approach. Computer Commun. 2024;216:95–111. doi: 10.1016/j.comcom.2024.01.002.

[105] Agrawal H, Gupta T. Cloud computing: A latency and bandwidth cost optimization perspective. Conference 18th USENIX Symposium on Operating Systems Design and Implementation. vol. 18; 2024. p. 117–34.