Received: 10 August 2022 | Revised: 5 September 2022 | Accepted: 15 September 2022

IET Networks

DOI: 10.1049/ntw2.12064

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Towards a cooperative hierarchical healthcare architecture using the HMAN offloading scenarios and SRT calculation algorithm

Ahmed M. Jasim[1,2] | **Hamed Al-Raweshidy**[1]

[1]College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge, London, UK

[2]Department of Computer Engineering, University of Diyala, Baqubah, Iraq

Correspondence

Hamed Al-Raweshidy, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge, London, UK.
Email: hamed.al-raweshidy@brunel.ac.uk

Funding information

Brunel University London

Abstract

The Internet of medical things (IoMT) is one of the most promising fields that is expected to rapidly expand in the near future. Consequently, vast amounts of data will be generated, necessitating faster and more intensive processing capabilities. Several healthcare architectures based on Edge/Fog technologies have been created to lower healthcare expenses and provide better and more reliable services. Scalability, availability, capacity, latency, and privacy are some of the most pressing issues to consider when designing such architectures due to the critical and sensitive nature of healthcare data. To contribute to the reliability and robustness of electronic health services, this work proposes Healthcare Metropolitan Area Network (HMAN), a cooperative hierarchical Edge/Fog computing-based architecture for the urban healthcare systems. The presented architecture suggests HMAN offloading scenarios and system response time calculation (HOSSC) algorithm which is specially designed to provide an abundance of offloading and processing scenarios within the network. The architecture also connects patients to the healthcare system by utilising the existing infrastructure in cities (e.g. medical centres and hospitals). Simulation results revealed that the designed architecture produced a ubiquitous and scalable healthcare system with promising and competitive performance, such as the computing capacity and service availability, by adopting multiple cooperative hierarchical offloading scenarios across the framework units. Moreover, the HMAN system was evaluated for latency and found to be very robust, with a short response time ranging between 6.043 and 31.45 ms in responding to 1 to 300 patients simultaneously sending. In addition to these appealing features, the proposed architecture ensures patients' privacy because the data are locally stored and processed in the most anticipated scenarios. The proposed architecture is a viable solution to providing healthcare services to a large number of patients.

1 | INTRODUCTION

The Internet of Things (IoT) phrase was coined in 1999 by Kevin Ashton, a British technology pioneer [1]. Nowadays, the phrase 'Internet of Things' refers to any network of devices or things that can be characterised as a system connecting objects, involving humans, animals, and inanimate objects, over the Internet. The number of IoT applications have been exponentially growing [2, 3]. According to Gartner's research, the worldwide number of connected devices could increase from 8.4 billion in 2020 to 20.4 billion in 2022 [4]. The McKinsey Global Institute estimated that IoT applications will have an annual global economic impact ranging from $3.9 trillion to $11.1 trillion [4]. According to Facts and Factors, the IoT market was expected to reach $1842 billion by 2028, rising at a Compound Annual Growth Rate of 24.5% from 2021 to 2028 [5].

The Internet of medical things (IoMT) is one of the most attractive areas in the recent IoT developments, with a promising business growth forecast of $158.07 billion by 2022 [6]. In certain ways, the IoMT might be considered one of the future's economic foundations. Furthermore, a wide range of services can be provided through a variety of IoMT

applications. Hence, massive data collected from endless IoMT sensors or devices must be processed at powerful data centres to provide additional insight to users and service providers [7].

In the case of near-real-time applications, the widespread adoption of cloud-based infrastructure may exacerbate the real-time constraints and burden the network infrastructure from the on-premises gateway to the cloud. Meanwhile, simple networking approaches are rarely feasible when attempting to provide healthcare services to a wide large of patients due to the complex nature of healthcare. For instance, patient privacy prevents data from being stored in a public cloud. Another issue to address is the patient's safety as data must be instantly available with a predetermined delay and intolerant cloud failures.

The emergence of cloud computing technology is creating new business opportunities, with the global market for this technology expected to exceed $1 trillion by 2024 [8]. However, relying on cloud computing to transfer data and wait for a response after the data are processed has several drawbacks. With regard to the data transmission cost, this process places additional burden on the network in terms of the required bandwidth and resources, resulting in degraded performance as the data volume increases.

This situation is considerably worse for time-sensitive applications, such as smart transportation, electricity grid, smart city, and some healthcare applications, wherein a short response time is not negotiable. Cloud computing cannot meet these standards because most data centres are located far away from users, causing delays. In addition to limiting available resources and bandwidth, cloud computing implementation might result in significant unacceptable network latency for time-sensitive applications [9].

To address these challenges, Edge/Fog computing paradigms have been introduced as complementary technologies for cloud computing. The primary goal of Edge/Fog computing is to shorten the distance between users and data processing centres (servers). Accordingly, various advantages are realised, such as faster response time. In addition, placing the servers at the same user network can improve the bandwidth and privacy level because the data can be stored and processed locally. This feature highlights how useful this technology would be for most applications, especially time-sensitive ones, such as healthcare [10].

According to projections, the world's population will reach 9.7 billion people in 2050 [11]. Overcrowding in cities poses serious threats to the quality of life to those who live there. According to WHO, poor healthcare management contributes to 2.6 million fatalities per year, and the rate is rising [12, 13]. Consequently, cities will face several challenges in the near future if appropriate solutions are not implemented. One of the major problems is the potential stress on the health system in cities. To address such issues, the emerging technology that allows humans and devices to work and communicate using IoT, Edge, Fog, and Cloud technologies is the electronic healthcare (e-healthcare) system. This system can be a reasonable alternative to the traditional healthcare techniques based on the rapid advancements in communication and information technologies.

In this work, we propose a unique cooperative hierarchical architecture for the healthcare system in cities based on Edge/Fog computing technology supported by the HMAN offloading scenarios and SRT calculation (HOSSC) algorithm that provides adaptable offloading techniques. The presented architecture, named as Healthcare Metropolitan Area Network (HMAN), connects the patients in a city to the healthcare system by utilising infrastructures that already exist in that city, such as medical centres (MCs) and hospitals (Hs). Consequently, this will provide an easy way to decide where to place servers since these facilities are spread all over the cities. This work mainly contributes to a ubiquitous and scalable e-healthcare system capable of delivering health services to patients without restrictions of space and time. The proposed system manages the data within the network as much as possible through the HOSSC algorithm and various offloading scenarios, resulting in less delay and high privacy levels. The suggested system networks patients, physicians, and family members to cooperatively monitor the patients and remotely obtain regular updates on their health conditions. In addition, relying on the hierarchical architecture is promising to gain an accumulative computational capacity because the data are cooperatively processed at multiple system units before reaching the cloud. Furthermore, the proposed system also contributes to increased availability because the designed scenarios ensure multiple methods to process the data, hence avoiding singular unit failures.

1.1 | Paper organisation

The remainder of this paper is organised as follows: Section 2 reviews the related work; Section 3 introduces the proposed HMAN architecture; Section 4 presents the data flow algorithm and the offloading scenarios in the suggested architecture; Sections 5 and 6 describe the proposed HMAN system and the offloading model, respectively; Section 7 presents and discusses the simulation results; Section 8 identifies system limitations and possible imporvements in the future; finally, Section 9 draws the conclusions along with the future work.

2 | RELATED WORK

In order to review the publications from the most reliable sources, a systematic literature review (SLR) technique is used in this paper. Finding, interpreting, and evaluating research findings that address the research questions is the major goal of the SLR. Both automated and manual searches were performed to gather the research findings from primary studies. The primary studies underwent a quality assessment in order to analyse the data and find the most appropriate results. The first step in the SLR is to define and document the search strategy. Inclusion and exclusion criteria of the research papers are the next steps. Then, quality criterion assessment is defined, while quantitative meta-analysis is the last step in the process.

Based on the existing literature, the idea of using wearable sensors to monitor patients has been investigated for many

years. It started with the use of PC-based stations, right up to the use of modern technological solutions such as the IoT technologies that have proven their ability to effectively develop healthcare at various levels. A large number of research works examine developments in the key enabling technologies for the IoMT, which include edge/fog computing, wearable medical devices, communication networks, and cloud computing. For example, Asif-Ur-Rahman et al. [14] proposed a heterogeneous IoHT framework consisting of five layers. Although mobile healthcare data can be offloaded to the cloud for processing, analysis and storage, offloading data to remote clouds still results in excessive latency. Furthermore, privacy is not addressed when offloading, placing sensitive health data at risk of external attacks. The number of possible scenarios for data transmission and processing paths is limited, which raises concerns about system scalability and service availability.

In the same way, M. Ahmad et al. [15] proposed a framework of Health Fog where the cloud and patients are separated by a layer of fog computing to reduce the E2E extra communication cost. The framework was provided by a cloud access security broker to enhance data privacy and security. In ref. [16], the researchers presented a privacy preserving healthcare system for data management in cloud. The blockchain technology was used to store all medical data to increase privacy. T. Muhammed et al. [17] proposed a four-layer ubiquitous healthcare framework based on edge computing technology to optimise data rate, data caching and data decisions. A cloud-fog-based IoT healthcare framework was structured in ref. [18] to optimise the latency issues when cloud computing was used only to process the offloaded healthcare data. However, sending medical personal data outside the network increases privacy concerns and latency issues with the increase in the data size.

Another system for patient monitoring is suggested by A. M. Rahmani et al. [19], which introduced a fog-computing-based healthcare system architecture integrated with smart e-Health gateways. The strategic position of these gateways at the edge of the network was exploited to present a Smart e-Health Gateway through offering some important services, such as local storage, real-time local data processing, and data mining.

A different solution by C. Kai et al. [20] that investigated the collaborative computation offloading, computation and communication resource allocation schemes and developed a collaborative computing framework to improve the cloud-edge-end task processing efficiency whilst maintaining limited computation and communication capabilities. A pipeline-based offloading strategy was proposed to partially process the collected data at the terminals, edge nodes and cloud. Nguyen et al. [21] presented BEdgeHealth, a decentralised architecture that combined multi-access edge computing with blockchain for data offloading and sharing amongst distributed nodes. Rajasekaran et al. [22] suggested an autonomous monitoring system model to provide healthcare services. This model uses the analytical hierarchy process for equitable distribution of energy amongst the nodes. The results demonstrated that the proposed model could support a large number of nodes with less energy.

There are also other applications in patients monitoring: I. Azimi et al. [23] introduced a portable detection and prediction monitoring system of the patients' health deteriorations that can be used at Hs or at homes. G. Muhammad et al. [24] presented a pathology monitoring system by using a cloud-based IoT and a machine learning classifier. Scalability, secure transmission and availability are amongst the concerns that still need to be addressed. S. He et al. [25] proposed an IoT-enabled medical services framework called FogCapCare to detect patented heart health conditions by integrating a cloud layer with a sensor layer. Verma [26] presented a Fog-enabled Technique for Clinical Healthcare framework based on edge computing, deep learning and automated monitoring to deliver highly useful real-life healthcare systems, such as cardiology.

In summary, the research studies reviewed here provided system architectures to improve data collection, management, and processing. These systems have reflected good performance in providing health services. However, concerns regarding the lack of data flow and offloading scenarios to increase the possibility of local data processing continue to persist. This situation mainly leads to other concerns, such as latency, data privacy, service availability, and scalability issues. Therefore, the main objective of this research is to design a ubiquitous and scalable healthcare architecture with a high level of data management flexibility and system availability, while increasing privacy and capacity, and reducing latency.

An effective solution is to develop the system to be able to handle different workloads with various data processing scenarios by adopting a cooperative hierarchical structure. The architecture presented in this article is unparallelled in combining the concept of hierarchical Edge/Fog computing with unique cooperative offloading techniques (HOSSC algorithm) between the architecture units exploiting many data flow scenarios to reflect additional options of data processing paths within the network. This combination can create a scalable and ubiquitous health system applicable at cities by utilising (for the first time) the already existing infrastructure (e.g. Medical centres and Hs). The proposed architecture takes advantage of the MCs' and Hs' geographical locations to provide electronic health services whilst increasing the computational capacity and the quality of healthcare with greater privacy and reduced latency. Table 1 presents a comparison of the proposed architecture with the previous existing ones.

3 | PROPOSED HEALTHCARE METROPOLITAN AREA NETWORK ARCHITECTURE

One of the biggest challenges the healthcare providers can face is to meet the needs of people as their multiple health conditions worsen. These people require extensive support from healthcare providers because they may have a lower quality of life than others and a higher risk of premature death than usual [27]. However, a lack of patient status information hampers the ability of healthcare providers to meet these needs [27]. Moreover, any latency in providing the necessary medical

TABLE 1 Feature comparison table of Healthcare Metropolitan Area Network (HMAN) (proposed) with the previous existing architectures

	Structure		Features				
References	**Hierarchical**	**Cooperative**	**Scalability**	**Availability**	**Capacity**	**Privacy**	**Latency**
Ref. [14]	No	No	N/A	N/A	Yes	No	Yes
Ref. [15]	No	No	N/A	N/A	N/A	Yes	Yes
Ref. [16]	No	No	Yes	N/A	N/A	Yes	No
Ref. [17]	No	No	N/A	N/A	Yes	Yes	Yes
Ref. [18]	No	Yes	N/A	N/A	Yes	No	Yes
Ref. [19]	No	No	N/A	Yes	No	No	No
Ref. [20]	No	No	No	No	No	Yes	No
Ref. [21]	No	Yes	No	N/A	No	Yes	Yes
Ref. [22]	Yes	No	N/A	N/A	Yes	No	Yes
Ref. [23]	No	No	No	No	Yes	No	Yes
Ref. [24]	Yes	No	No	No	No	No	No
Ref. [25]	No	No	N/A	N/A	No	No	Yes
Ref. [26]	No	Yes	N/A	N/A	No	Yes	Yes
Proposed HMAN	Yes	Yes	Yes	Yes	Yes	Yes	Yes

response makes the physicians unable to deliver all the preventive services recommended by the US Preventive Services Task Force. According to Yarnall et al. [28], preventive services require roughly 37 min/year per child and 40 min/year per adult. Accordingly, providing high-quality healthcare services to these targeted patients at MCs or Hs requires more time than what these caregivers can afford [29]. On this basis, designing an advanced ubiquitous e-healthcare system has become crucial to fulfiling these aspirations through local monitoring and data processing. Based on the primary analysis of the locally collected data, the outcome of such an advanced system can help healthcare providers in managing their resources in providing services for the people in need.

In this section, we describe the proposed HMAN, a cooperative hierarchical architecture for the healthcare system based on Edge/Fog computing technology supported by a collaborative offloading algorithm (HOSSC). The HMAN system aims to provide a mobile 24 h monitoring service for patients in need. Furthermore, the suggested framework is promising to meet the healthcare important requirements, such as availability, scalability, communication delay, computational capacity, and privacy. The main idea of the proposed architecture design is based on the procedure of the existing standard healthcare systems. The integration of the existing infrastructure in the city (e.g. MCs and Hs) with the recent technology (i.e. Edge/Fog computing) can play a key role in achieving a robust system and providing its services to all patients in any city.

For example, most health systems, certainly in the USA and the UK, have two main levels of health services delivery, namely, primary and advanced. The primary level is provided by MCs distributed around cities to provide primary care for patients. Individuals that need to be examined or treated with a higher degree of expertise are filtered by these primary care facilities. Meanwhile, most Hs tend to provide advanced, more specialised, and skilled care to patients referred to them by the MCs. Accordingly, the proposed architecture can be built on these two layers and additional layers, as demonstrated in Figure 1. Based on the figure illustration, the HMAN architecture consists of four layers, an IoT layer, an Edge/Fog layer, a cloud layer and an application layer. A further detailed explanation of these four layers is provided in the following parts of this section.

3.1 | IoT layer

The IoT layer is the first layer of the proposed HMAN architecture. The incorporation of WBANs and innovative technologies in the various layers of IoT ecosystems has many benefits, such as enhancing storage and data availability while lowering data transmission latency [7]. Therefore, this layer consists of vital signs monitoring sensors, including on body or/and in body (implants) sensors, and a smartphone (or any wearable devices). The sensors and Wireless Body Area Network are connected to the patient's body to sense vital signs and send the sensed data to the smartphone through a wireless link (e.g. Bluetooth or Zigbee). The smartphone is responsible for receiving the data from the sensors and performing a basic preprocessing and initial data analysis (i.e. aggregation, fusion, filtering, and classification). Consequently, the received data are classified into normal and abnormal according to a predefined threshold assigned based on the patient's conditions. A physician may assign a threshold value for a diabetic patient, for instance, based on the patient's history record to take further actions. Meanwhile, the normally

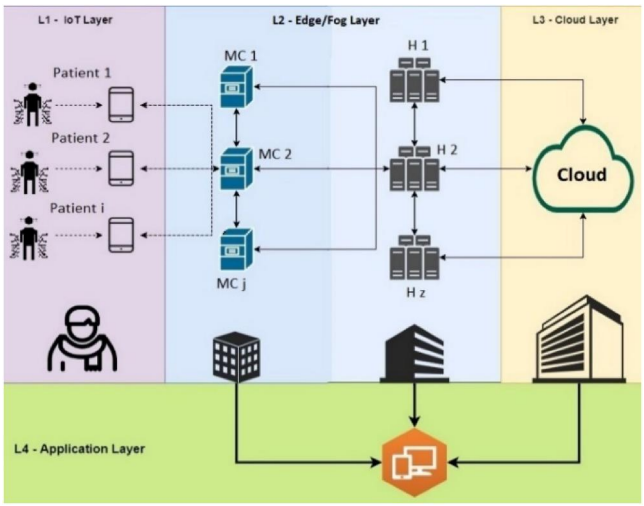

FIGURE 1 Proposed Healthcare Metropolitan Area Network (HMAN) Architecture. The arrow directions indicate the flow of data. Medical centre (MC) = MCs, H = Hospitals. The MCs and Hs are hierarchically connected

classified data are temporarily locally stored with no actions required, whilst the abnormal data can be offloaded to the next layer for further actions. Moreover, the offloaded data generate an alert to the patient and a family member to notify the condition.

3.2 | Edge/fog layer

The Edge/Fog layer is the next layer in the proposed architecture, where the local MCs and the Hs are hierarchically connected to cooperatively accommodate the patients' needs by the allocation of the nearest resources. This layer of the hierarchy's servers, which are often geo-distributed desktops or workstations, receives workloads directly from patients' smartphones through wireless connections. The proposed architecture suggests that each MC is connected to the nearest 2 MCs, and each group of MCs is linked to a local H through optical fibre links. Meanwhile, each H is connected to two neighbouring Hs through optical fibre links, whilst each H is linked to the cloud via the Internet backbone. This cooperative hierarchy architecture summarises the main contribution of this study on how the data offloading can be collaboratively managed between the neighbouring MCs and Hs and how it can help the system gain more computational capacity. If the workloads received by an MC exceed its computational capacity, then the overload amount of data is further offloaded either to a neighbouring MC or to the local H. The same method is applied to the local H when dealing with incoming data. Consequently, the forwarded data inherit more computational capacity gained at the servers of these facilities based on the proposed hierarchy architecture. More detailed scenarios of the data flow are provided in Section 4 to emphasise the cooperative hierarchy of data processing along with the accumulatively gained computational capacity. Assigning only two neighbouring facilities to back up the local MCs and Hs achieves the aimed inherited computational capacity while sharing the data within as fewer local parties as possible and maintain a simplistic, applicable and cost-effective system. Concurrently, this neighbouring units' collaboration reduces the distance and latency.

3.3 | Cloud layer

The cloud layer is the third layer in the proposed architecture, which represents the furthest layer the data can reach according to the data flow scenarios detailed in Section 4. Specifically, it is the worst case of the proposed offloading techniques happening only when the workload overloads all the previous layers' capacity. Hence, the overload data must be sent to the cloud.

3.4 | Application layer

The application layer is the topmost layer of the proposed HMAN architecture. The user interface between the relevant members and the system itself is provided by this layer. Moreover, the system administrator can access the system resources through this layer.

4 | HOSSC—DATA FLOW AND OFFLOADING ALGORITHM

Computational offloading is a distributed paradigm for transferring all or a portion of the data from local servers to remote ones to speed up data processing and conserve energy. However, this process has several conditions. First, local execution cannot be done due to the limited resources of the local servers. Second, the offloading time, including the communication time and the remote execution time, is less than the local execution time, as expressed in Equation (1) [30].

$$T_{\text{offloading}} < T_{\text{local}} \tag{1}$$

Algorithm 1 presents an explanation of the proposed HOSSC algorithm at different layers of the suggested system. The algorithm determines whether or not the data offloading is necessary between the HMAN units after considering the time latency in Equation (1) based on the size of the workload received by an MC or an H compared with the maximum data processing capacity of these facilities. To better understand the HOSSC algorithm, as illustrated in Figure 2, the following step-by-step data flow and offloading algorithm from the data collection to the final outcome is described in the next part of this section.

The monitored patient's data are collected with a smartphone and classified into normal and abnormal based on a preset threshold assigned by healthcare specialists. The normal data are temporarily stored for further future analysis of the

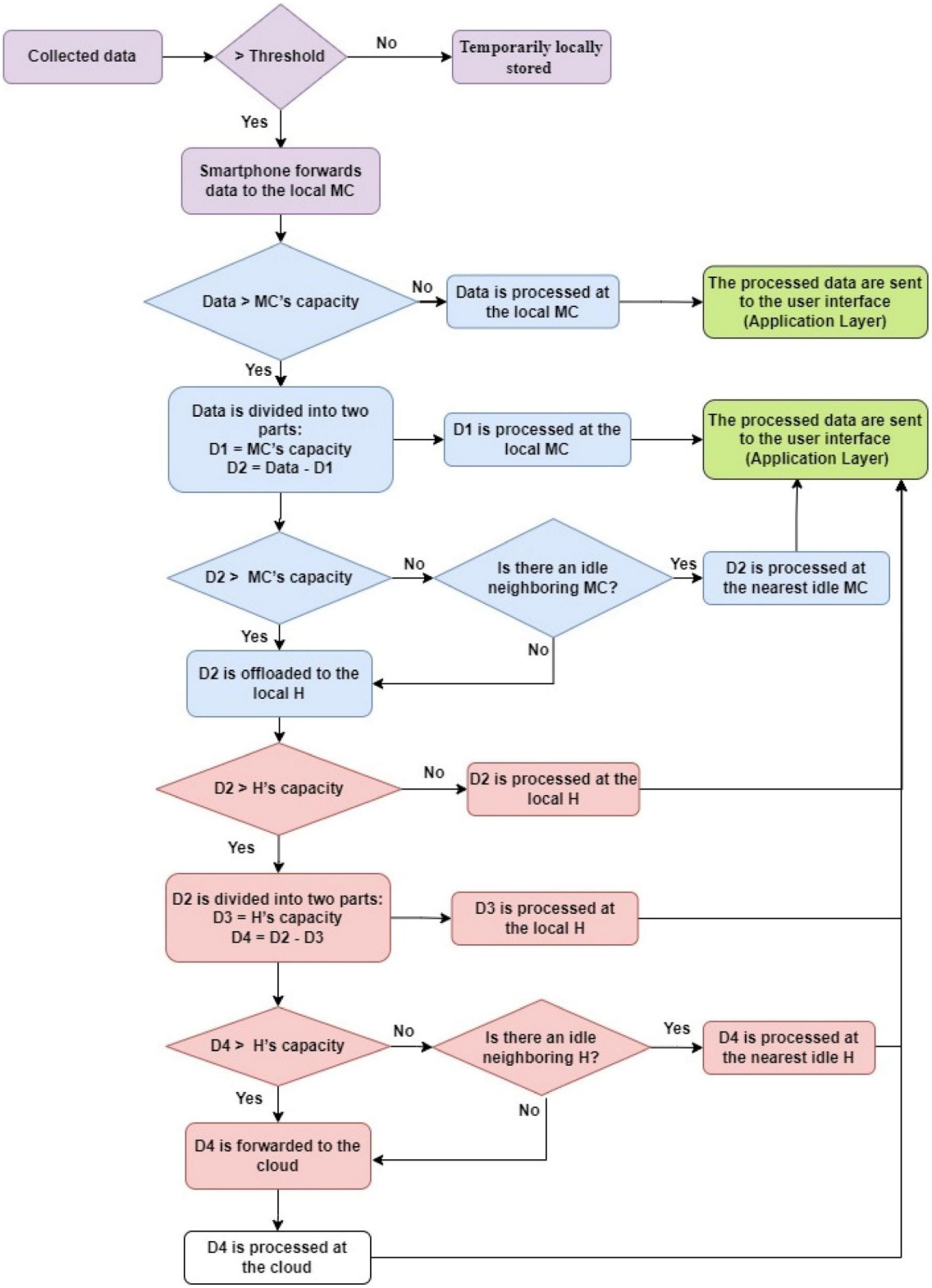

FIGURE 2 Proposed HMAN offloading scenarios and SRT calculation (HOSSC) data flow and offloading algorithm of the Healthcare Metropolitan Area Network (HMAN) architecture

patient's records. Meanwhile, the abnormal data are immediately sent to the local MC, and an alert to a family member is initiated. Once the abnormal data reaches the nearest MC in the patients' area, that MC processes the data directly if the available resources are capable to do so. However, if the received workload requires greater computational resources

than the MC's capacity, then the MC processes the data within its capacity and offloads the overload data to the connected neighbouring MCs. The only possibility that data must be forwarded to the local H is when both neighbouring MCs are busy.

The same cooperative processing in between MCs is applied to the Hs to manage the received data by the local H. The received data is processed within the H server's capacity unless it requires more computational capacity. If so, the overload data can be then forwarded to the neighbouring cooperative Hs. Therefore, the data are mostly managed within the MCs and the Hs without reaching the cloud. The data arrives the cloud only if all local and neighbouring MCs and Hs cannot accommodate the overload data. On this basis, the possible abnormal data flow and offloading scenarios are follows:

1) Scenario 1: The abnormal data collected by IoT devices require less processing capabilities than what the local MC can afford. Hence, the local MC manages to process the data.
2) Scenario 2: The data require more processing capabilities than that available at the MC. Accordingly, the data are partly processed at the MC, whilst the rest are offloaded to the nearest idle neighbouring MC. This scenario assumes that the offloaded overload data can be accommodated by the processing capacity of at least one neighbouring MC. Otherwise, scenario 3 is applied.
3) Scenario 3: The data require more processing capabilities than that available at the MC and at the neighbouring centres. Accordingly, the data are partly processed at the local MC, whilst the rest are offloaded to the local H. This scenario assumes that the offloaded overload data can be accommodated by the processing capacity of the local MC and the local H. Otherwise, scenario 4 is applied.
4) Scenario 4: The data require more processing capabilities than that available at the local MC and the local H. Accordingly, the data are partly processed at the local MC and H, whilst the rest are offloaded to the nearest idle neighbouring Hs. This scenario assumes that the offloaded overload data can be accommodated by the processing capacity of at least one H of the neighbouring ones. Otherwise, scenario 5 is applied.
5) Scenario 5: The data require more processing capabilities than that available at the cooperative MCs and Hs. According, the data are partly processed at the local MC and H, whilst the rest are offloaded to the cloud. This scenario expresses the only possibility that the data can reach the cloud. Figure 3 visualises the five scenarios.

Having addressed the possible data flow and offloading scenarios, the HMAN framework can contribute to enhancing the provided services by:

- Offering a high level of patients' privacy by processing the data locally without reaching the cloud as much as possible. This step is met through cooperative data processing and offloading in between facilities.

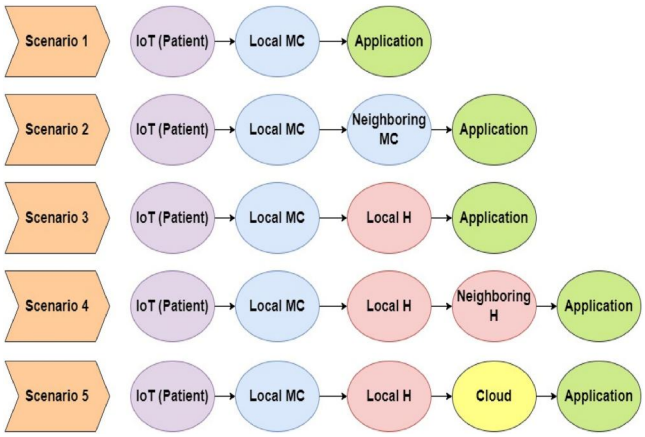

FIGURE 3 All potential scenarios in the proposed system

- Increasing the computational capacity through the hierarchical connection of the facilities in the Edge/Fog layer.
- Decreasing the system response time (SRT) (i.e. the latency) as a result of the greater possibilities of local data processing shortening the distance between the patients and the healthcare providers.
- Maintaining high availability of service because the presented system does not rely on one scenario to forward and process the data. Meanwhile, the HMAN sustains a reliable service by providing multiple scenarios that can cooperatively process the data within the system's units to ensure that the service is constantly maintained for patients

5 | HEALTHCARE METROPOLITAN AREA NETWORK SYSTEM MODEL

The HMAN architecture is cased studied on the healthcare system in the UK for a more convenient derivation of the proposed model. The healthcare system in the UK consists of two main levels, namely, the general practices (GPs) level and the general Hs (GHs) level. The GPs are the primary MCs distributed throughout the city and are responsible for providing medical services for local patients. Meanwhile, the GHs are the GHs, and each one is responsible for hospitalising patients referred by a group of GPs.

In terms of applying the proposed architecture on the healthcare infrastructure in the UK, we can represent the HMAN system as an undirected graph $G = \{p \cup GP \cup GH, E\}$, where p is a set of monitored patients, $P = \{p1, p2, p3, \ldots, pi\}$, GP is a set of MCs in a local area of the city, $GP = \{gp1, gp2, gp3, \ldots, gpj\}$, GH is a set of the GHs in the city, $GH = \{gh1, gh2, gh3, \ldots, ghz\}$, and E represents the connection links that can be a wire (optical fibre) or wireless (Wi-Fi connection). Figure 4 illustrates the components of the HMAN system.

In the HMAN system, the patient can access any local GP service either through a direct link if he/she is in that GP coverage area or through the access point (AP) he/she is connected to. Each GP is connected to two GPs (the two

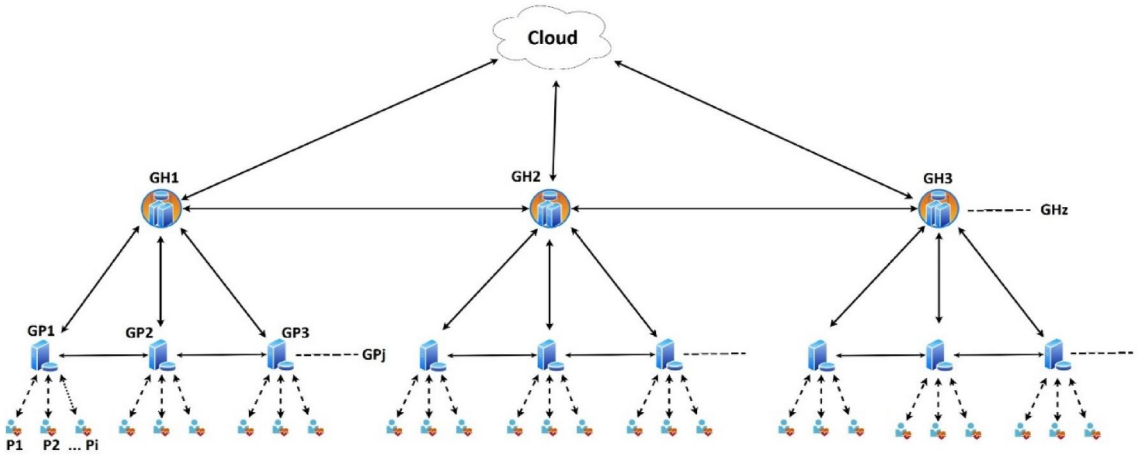

FIGURE 4 All Components of the healthcare metropolitan area network (HMAN) system. The arrow directions indicate the flow of data. *P* = patients, GP = medical centres (general practice), GH = general hospitals. The GPs and GHs are hierarchically connected

nearest neighbours) and is responsible for providing services to patients within its area of responsibility. Meanwhile, every GH is connected to two GHs (the two nearest neighbours) and is responsible for all local GPs located within its area of responsibility. The fibre optics are utilised to link the aforementioned healthcare facilities to provide high-speed connections. In addition, the remote cloud is accessible to all GHs through high-speed Internet connections.

In the first layer of the proposed architecture, the patient's smartphone classifies the collected data into normal or abnormal according to a threshold value of each health issue. The normal data are invisible to the further layers in the system, and the only offloaded data to the local GP for processing are the abnormal data. Accordingly, each patient, p_i, has a task that randomly arrives in the system with an arrival rate λi, according to the Poisson process.

Algorithm 1: HMAN offloading scenarios and SRT calculation (HOSSC) algorithm

Input: Ps, r_{Fib}, λ_I, T_{WI}, T_{AP}, B, μ_{MC}, μ_H, μ_{Cloud}, λ_{MC}, λ_H
Output: Present five offloading scenarios and facilitate the SRT calculation.
Assumptions: Each MC is connected to two neighbouring MCs, and each H is connected to two neighbouring Hs.

% 1- Data (workload) at the local MC.
a- Calculate: $\boldsymbol{\lambda_{Sum} \leftarrow \sum_{i}^{Ps} \lambda i}$ % The local MC's workload
b- Calculate: the data fraction scalar K1
if λ_{Sum} <= λ_{MC} then:
K1 ← 1 means all λ_{Sum} will be processed at the local MC and nothing will be offloaded **(Scenarios 1).**
Else then:
K1 ← λ_{MC} / λ_{MCSum} means a part of data (K1. λ_{Sum}) will be offloaded to the higher units.
end if statement.
% In both cases, we need to calculate: the MC's queueing time ***fQ* (k1. λ_{MCSum}) and** the processing time. Then calculate the response time: $t_{Mc} \leftarrow$ ***fQ* + processing time.**

% 2- When $\lambda_{Sum} > \lambda_{MC}$
a- Calculate: **$\lambda j \leftarrow$ (1-K1) * λ_{Sum}** % λj = The local MC workload which should be offloaded to: 1- one or both Idle neighbours, or 2- the local H
b- Calculate: ***V* and *S* ← (λj *8)/r_{Fib} + 4.9 microseconds × Distance (kilometre)** % Fibre Optic Latency
% Now, we have **two cases:** 1- λj <= λ_{MC} means that is possible to send data to one of the neighbouring MCs **(Scenario 2).** Therefore, the local MC will first check for an Idle neighbouring MC. 2- λj > λ_{MCSum} means the data must be offloaded to the local H.

% **Case 1: $\lambda j \leq \lambda_{MCSum}$**
% Calculate the response time in MC t_{Mc} s or Hs t_H based on the state of neighbouring MCs:
1- [0,0] → Both neighbours are busy, then the MC offloads the data to the H **(Scenario 3)**.
2- [1,0] → neighbour 1 is idle, then the MC offloads the data to the MC (neighbour 1) **(Scenario 2)**.
3- [0,1] → neighbour 2 is idle, then the MC offloads the data to the MC (neighbour 2) **(Scenario 2)**.
4- [1,1] → Both neighbours are idle, then the MC offloads the data to the nearest one **(Scenario 2)**.
% Case 2: $\lambda j > \lambda_{MC}$ means the local MC must offload the **λj data** to the local H. Now, we also have two cases:
% **Case A:** if $\lambda j <= \lambda H$ % All λj data is processed at the H **(Scenario 3)**.
$K2 \leftarrow 1$
% **Case B:** $\lambda j > \lambda H$ % A part of the λj data should be offloaded to the higher units.
else
$K2 \leftarrow \lambda_H / \lambda j$
End if statement.
% In both cases, we need to calculate: the queueing time ***fQ* (k2. λj)** and processing time in the local H. Then, calculate the response time **$t_{Hs} \leftarrow$ *fQ* + processing time**

% 3- **When $\lambda j > \lambda_H$**
a- Calculate: **$\lambda z \leftarrow (1-K2) * \lambda j$** % λj = The local H's workload which should be offloaded out of the local H to: 1- one or both Idle neighbours, or 2- the cloud.
b- Calculate **$L \leftarrow (\lambda z \times 8) / r_{Fib}$ + 4.9 microseconds × Distance (kilometre)** % Fibre Optic Latency
% Now, λz must be offloaded according to two cases: 1- $\lambda z <= \lambda_H$ means that is possible to send data to one of the neighbouring H (**scenario 4**). So, the local H will check for an Idle neighbouring H. 2- $\lambda z > \lambda_H$--> the data must be offloaded to the Cloud **(scenario 5)**.
% Case 1: $\lambda z <= \lambda_H$
% Calculate the response time in H t_H s or cloud t_{cloud} based on the state of neighbouring Hs:
1- [0,0] → Both neighbours are busy, then the H offloads the data to the Cloud **(Scenario 5)**.
2- [1,0] → neighbour 1 is idle, then the H offloads the data to the H (neighbour 1) **(Scenario 4)**.
3- [0,1] → neighbour 2 is idle, then the H offloads the data to the H (neighbour 2) **(Scenario 4)**.
4- [1,1] → Both neighbours are idle, then the H offloads the data to the nearest H (neighbour). **(Scenario 4)**.
% Case 2: $\lambda z > \lambda_H$ means the local H must offload the **λz data** to the cloud
% Calculate **$\lambda_{Cloud} \leftarrow \lambda z$** % λ_{Cloud} is H's workload which needs to be offloaded to the cloud
% Calculate the response time in the Cloud: **$t_{cloud} = (\lambda_{Cloud} / \mu_{Cloud})$;**

% 4- Now, calculate the delay for each patient:
$tpi \leftarrow T_{Wi} + T_{AP} + t_{Mc} \times (C1 - 1) + V \times C1 + t_H \times C2 + S \times C2 + C3 \times B + t_{cloud}$.
% C1, C2, and C3 are counters for counting the number of times data was offloaded and returned from system units.
% 5- Finally, calculate the system SRT:

$$SRT \leftarrow \frac{\sum_{1}^{Ps} tpi}{Ps}$$

Figure 5 depicts the proposed system model where the data can be offloaded from a patient to a local GP either through a direct connection with a wireless delay (Twi) or through an AP with a total delay of (Twi + T_{AP}). In addition, V, S, and B denote the delay in between GPs, GPs to GHs or in between GHs and the Internet delay respectively.

6 | OFFLOADING SYSTEM MODEL

The HMAN system is designed as a queueing model. The offloaded data can be processed at the GPs, GHs, or remote cloud according to the servers' status in each site. We assume that all GPs' servers are identical, and the same assumption

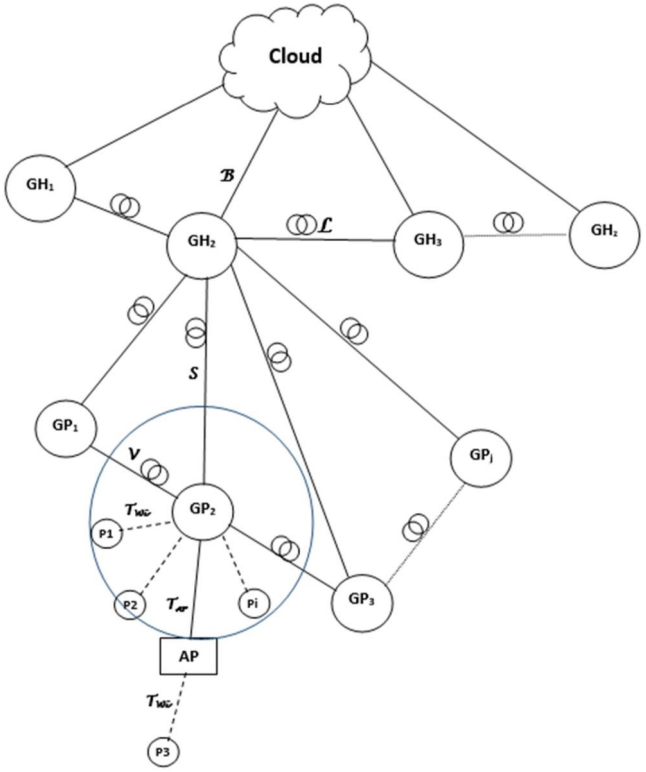

FIGURE 5 Healthcare metropolitan area network (HMAN) system model

applies on GHs' servers. However, the GHs host more servers with higher specifications and capabilities than the ones hosted at the GPs. Let us denote the number of servers at each GP site as n and the number of servers at every GH as m. In addition, GPs and GHs sites are modelled as M/M/n and M/M/m queue models, with the fixed rate service μGP and μGH respectively. The following part presents the offloading system model after applying the data flow scenarios described in Section 4 on the case studied healthcare system in the UK.

The collected data from each monitored patient create a task that can be executed at the patient's smartphone when the data are determined to be normal; at the local GP or its neighbouring GPs when the data are determined to be abnormal; at the local GH or its neighbouring GHs when the abnormal data require more processing capacity than the cooperative GPs' capacity; or at the cloud when the previous units cannot afford enough processing capacity to execute the abnormal data.

We can find the average queueing time, f_Q, using the equation below [31]:

$$f_Q(\lambda) = \frac{C\left(n, \frac{\lambda}{\mu}\right)}{n\mu - \lambda} \tag{2}$$

where λ is the arrival rate, and C(n, A) is Erlang's formula [31] obtained by:

$$C(n, A) = \frac{\left(\frac{(nA)^n}{n!}\right)\left(\frac{1}{1-p}\right)}{\sum_{k=0}^{n-1}\left(\frac{(nA)^k}{k!}\right) + \left(\frac{(nA)^n}{n!}\right)\left(\frac{1}{1-A}\right)} \tag{3}$$

6.1 | Response time at general practices

Assuming that $P = \{p_1, p_2, p_3, \ldots, p_i\}$is a set of patients assigned to the local GP, the patients' smartphones offload the collected abnormal data to the local GP for processing. If the local GP is overloaded, then the GP processes the data within its available capacity, whilst the remaining data are offloaded either to the nearest idle GP or to the local GH based on the criterion, $\lambda_{\text{GPmax}} < (\lambda_{\text{GPsum}} = \sum \lambda_i)$, where λGPmax represents the maximum GP workload, and λGPsum is the arrival workloads from the patients.

To determine the amount of data to be processed at the local GPs and those to be offloaded to the cooperative units, we must calculate the data fraction scalar K_1 as follows:

$$K_1 = \begin{cases} 1 & if \quad \lambda_{\text{GPmax}} \geq \lambda_{\text{GPsum}} \\ \dfrac{\lambda_{\text{GPmax}}}{\lambda_{\text{GPsum}}} & \text{otherwise} \end{cases} \tag{4}$$

Then, we can find the response time at the GPs, t_{GP}, for the received data as follows:

$$t_{GP} = f_Q\left(K_1 . \lambda_{\text{GPsum}}\right) + \frac{K_1 . \lambda_{\text{GPsum}}}{\mu_{GP}} \tag{5}$$

where $f_Q\left(K_1 . \lambda_{\text{GPsum}}\right)$ represents the queueing time needed for the fraction of the arrived data to be processed at the GP, and $\frac{K_1 . \lambda_{\text{GPsum}}}{\mu_{GP}}$ represents the service time at the GP for the processed workload.

6.2 | Response time at general Hs

Assuming that GP = $\{gp_1, gp_2, gp_3, \ldots, gp_i\}$ is a set of GPs assigned to the local GH, the GP offloads its data to the local GH. In case the local GH is overloaded, the data will be offloaded either to the nearest idle GH or to the cloud according to: $\lambda_{\text{GHmax}} < (\lambda_{\text{GHsum}} = \lambda_{\text{GHsum}} - \lambda_{\text{GHmax}})$, where λGHmax represents the maximum GH workload, and λ_{GHsum} is the arrival workloads from the GPs.

To determine the amount of data to be processed at the local GH and those to be offloaded to the neighbouring GHs or further to the cloud, we must calculate the data fraction scalar K2 according to:

$$K_2 = \begin{cases} 1 & if \quad \lambda_{\text{GHmax}} \geq \lambda_{\text{GHsum}} \\ \dfrac{\lambda_{\text{GHmax}}}{\lambda_{\text{GHsum}}} & \text{otherwise} \end{cases} \tag{6}$$

Accordingly, we can calculate the response time at the GHs, t_{GH}, for the arrived data using:

$$t_{GH} = f_Q\left(K_2.\lambda_{GHsum}\right) + \frac{K_2.\lambda_{GHsum}}{\mu_{GH}} \quad (7)$$

where $f_Q\left(K_2.\lambda_{GHsum}\right)$ represents the queueing time needed for the fraction of arrived data to be processed at the GH, and $\frac{K_2.\lambda_{GHsum}}{\mu_{GH}}$ denotes the service time at the GH for the processed workload.

6.3 | Response time at the cloud

The cloud can be modelled as M/M/∞ queue referring to its unlimited resources. Thus, we can consider the queueing time at the cloud as zero and estimate the response time at the cloud as follows:

$$T_{cloud} = \frac{\lambda_{Cloud}}{\mu_{cloud}} \quad (8)$$

According to Equations (5), (7) and (8), the average response time of the offloaded tasks by a patient in the HMAN system is calculated as follows:

$$\begin{aligned} t_{pi} = {} & T_{wi} + E_{AP} * T_{AP} + (c_1 + 1) * t_{GP} + c_1 * V \\ & + c_2 * S + c_2 * t_{GH} + c_3 * L + c_3 * t_{GH} \\ & + c_4 * B + c_4 * t_{Cloud} \end{aligned} \quad (9)$$

where c_1, c_2, c_3 and c_4 represent the counters to count the number of times to reach a certain unit in the system; c_1 = GP neighbours' counter; c_2 = the local GH counter; c_3 = GH neighbours' counter; and c_4 = the cloud counter. E_{Ap} represents the link between the patient and the local GP; and $E_{Ap} = 0$ (direct connection) or 1 (through an AP).

Finally, the SRT of the proposed framework can be determined by:

$$SRT = \frac{\sum_{i=1}^{n} t_{pi}}{n} \quad (10)$$

7 | RESULTS

In this section, we present the simulation results of all the conducted potential scenarios according to the proposed HMAN system model. A few parameters are defined and listed in Table 2 for a more convenient evaluation of the system's feasibility and advantages. The SRT when simultaneously processing varying numbers of patients sending data is considered a metric to measure the performance of the system. Meanwhile, the other important system aspect to look for is the system scalability and how far the HMAN system can sustain a reliable service responding to the simultaneous data flow.

The cooperative units (i.e. the local and neighbouring GPs and GHs) are gradually deployed to respond to different workloads to examine how efficient and scalable the hierarchy HMAN architecture is. The system units are gradually engaged in five stages starting from only the local GP and ending at the

TABLE 2 System parameters

Symbol	Parameter	Value
Ps	Number of patients	100, 200, 300
n	Number of servers in each GP	5, 6, 7
m	Number of servers in each GH	10, 12, 14
r_{Wi}	Link rate between IoT device and AP	54 Mbps
r_{AP}	Link rate between AP and the local GP	100 Mbps
r_{Fib}	Link rate between the GPs and GHs	Up to 10 Gbps
λi	Packet size	30 KB
T_{wi}	Wireless delay	4 ms
T_{AP}	Delay between AP and the local GP	2 ms
V	Delay between two neighbouring GPs	$(1-K_1).\lambda_{GPsum}/10$ Gbps
S	Delay between the GPs and GHs	$K_2.\lambda_{GHsum}/10$ Gbps
L	Delay between two neighbouring GHs	$(1-K_2).\lambda_{GHsum}/10$ Gbps
B	Internet delay	20 ms
μGP/μGH	Each GP/GH server service rate	100/200 KB per ms
μ_{Cloud}	Cloud service rate	1000 KB per ms
λ_{GPmax}	Maximum GP workload	200×n KB
λ_{GHmax}	Maximum GH workload	200×m KB

whole proposed system, including all the cooperative units. We initially evaluated the SRT to the data from a certain workload that can be only processed at the local GP and Cloud for greater clarity in visualising this progressive deployment.

Then, we gradually deploy more units in between the local GP and the cloud for the same workload. We engage the neighbouring GPs, the local GH, the neighbouring GHs, and the whole cooperative system to respond to the same workload under these stages of more involved units. This situation can emphasise how the hierarchy proposed architecture can collaboratively manage the workload. In terms of workloads, the system is tested for 100, 200 and 300 patients, and the results of the SRT are presented in Figures 6, 7 and 8 respectively. Each figure includes five graphs representing the SRT of the HMAN system under each of the aforementioned five stages for the same workload. For a fair evaluation of the suggested architecture in responding to different workloads, each of the five graphs in Figure 6 is discussed with the corresponding graphs from Figures 7 and 8. Figure 6 shows that the SRT is reduced when the data are offloaded from the local units (GPs or GHs) to their neighbouring units rather than being transmitted to the upper layers.

Figure 6a shows that when two layers and 100 patients are sending data at the same time, a portion of the data must be offloaded to the cloud due to the limited resources at the GP layer. Accordingly, the response time is longer, which is undesirable in many pathological situations where a faster response is required. Additionally, these offloaded data are sufficient to raise privacy concerns because a portion of the data are sent outside the network in such a scenario. Furthermore, this situation can be even more challenging when serving more patients (i.e. a large number of data are likely to be processed outside the network), as shown in Figures 7a and 8a for 200 and 300 patients respectively.

In the second stage, the system supports the local GP by two GP neighbours. The local GP first checks if the data can be offloaded to the nearest available neighbour (i.e. when the overload data are less than λ_{GPmax}) rather than being sent to the cloud. Such neighbouring GP involvement creates an additional scenario to locally process the data and achieve more privacy, greater inherited computational capacity and less latency. Thus, we obtained less SRT and expanded the system to ensure that more patients can be served within the local network, as illustrated in Figures 6b, 7b and 8b.

The following stage is to engage the local GH to serve as a reference point for all the GPs in the same area. Accordingly, an additional scenario to process the data is provided compared with the previous stage. When the data exceed the capacity of the local GP neighbours, this new scenario becomes valid, and the overload data are offloaded to the local GH. Although the SRT increased in responding to a 100 patients' workload (Figure 6c), the SRT decreased when handling heavier loads of 200 and 300 patients (Figures 7c and 8c). The higher SRT in a 100-patient workload is attributed to the longer distance the offloaded data can go through as it travels from the local GP to the local GH, which is often located further than the neighbouring GPs. Nevertheless, the computational capacity of the GHs is higher than the GPs; hence, a larger number of patients can be accommodated. Furthermore, a higher level of the patients' privacy is achieved because this system expansion still assures the local processing of the data.

When the local GH is connected to the two GH neighbours in the fourth stage, the system's performance is improved. In the previous stage, when the data overload the local GP and GH, the overload data must be offloaded to the cloud, exposing the data to less privacy and higher latency. Meanwhile, this stage introduces an additional scenario to locally process the data at the neighbouring GHs. The local GH checks the status and the capacity of the neighbouring GHs and decides whether to send the overload data to the neighbouring GHs or to the cloud. Consequently, the system would have more probability of local data handling and less possibility of reaching the cloud. Thus, more privacy is met, more patients are served, and less latency can be achieved. The results depicted in Figures 6d and 7d cannot show this SRT improvement because the local GH along with the previous units are capable of processing the data; hence, no overload data reaches the local GH neighbours. However, we can clearly see the aimed SRT improvement in Figure 8d because the system affords further resources to avoid the data offloading to the cloud.

In the final stage, when the entire system is connected, more computational capacity is gained, resulting in a more powerful processing platform capable of responding to a higher number of monitored patients. All the five scenarios conducted in this study are involved in this stage. The results presented in Figures 6e, 7e and 8e show that the performance is significantly improved compared with those in the previous stages. Moreover, the more healthcare units are involved in the cooperative HMAN architecture, the more computational capacity can be inherited, thereby reflecting less SRT. Furthermore, the patients' privacy is also increased because such a collaborative network achieves a higher probability that data are locally processed.

The simulation considered the highest expected values of the time delay to calculate. In addition, all patients are assumed to be indirectly connected to the health centres; hence, the T_{AP} of 2 ms was added to all the considered scenarios. Accordingly, the obtained results are for the worst case when it comes to the consideration of the time delay assumptions. Nevertheless, the presented HMAN framework efficiently serves a workload of 300 patients simultaneously requesting service with a latency of 31.45 ms when compared the latency of 75 ms achieved in ref. [14]. This remarkable reduction in the service delay is attributed to the presented architecture that utilises the nearest units for cooperative data processing. This utilisation of the MCs' and Hs' geographical locations shortened the distance between patients and health facilities and eventually reflected reduced latency.

Furthermore, the system performance is improved in all resulting graphs when increasing the servers at the GPs (n) from 5 to 6–7 and the GHs (m) from 10 to 12–14. This result makes sense due to the increase in resources.

Consequently, more data are locally processed rather than being offloaded to the upper layers. Table 3 illustrates the system achievements to indicate the performance in

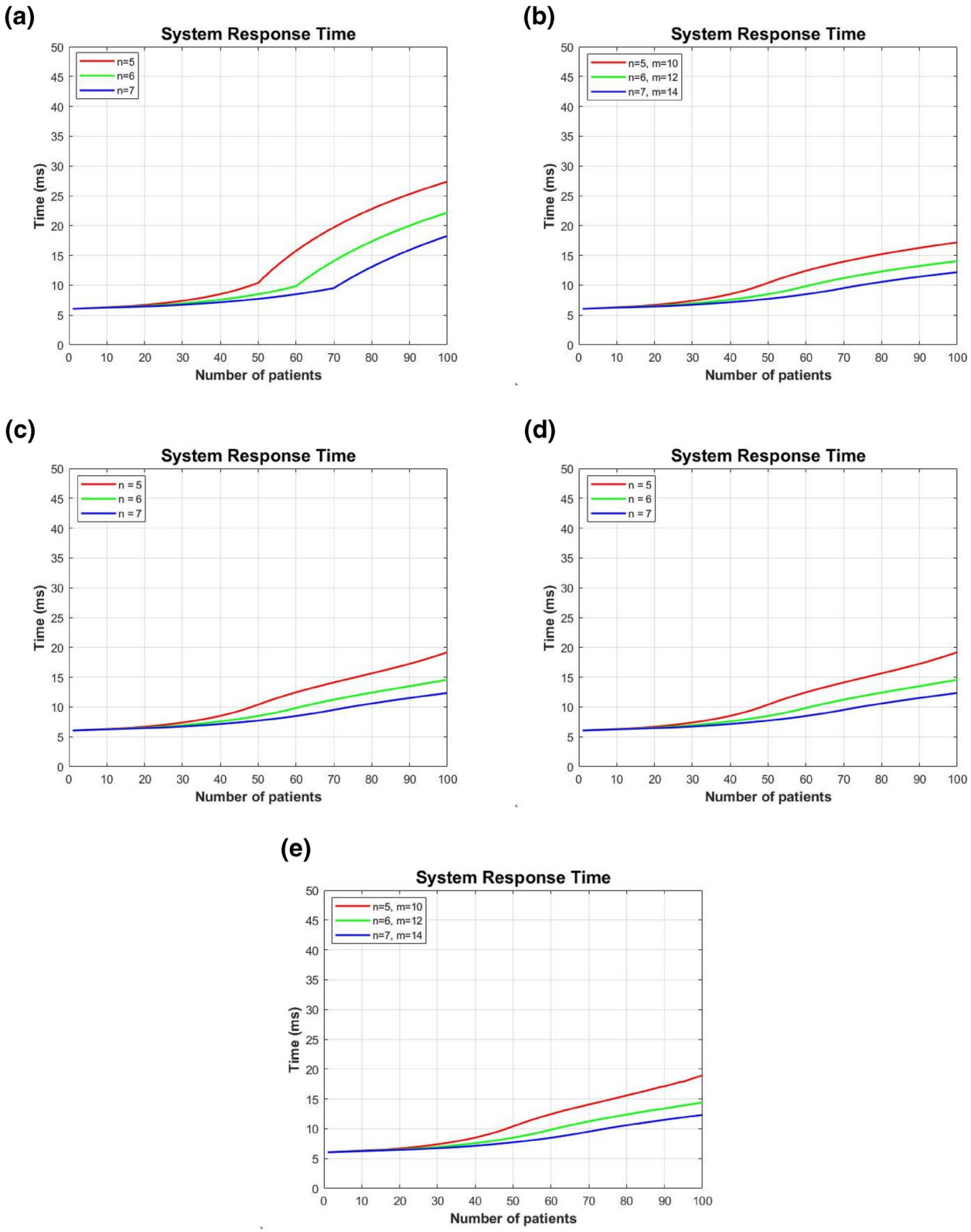

FIGURE 6 SRT of 100 patients in the expected different scenarios. (a) Patient → Local general practice (GP) → Cloud. (b) Patient → Local GP → GP neighbour → Cloud. (c) Patient → Local GP → Local general H (GH) → Cloud. (d) Patient → Local GP → Local GH → GH neighbour → Cloud. (e) Whole system

responding to different workloads as we gradually engage more units in the five testing stages. The aforementioned table emphasises how data become less exposed to the cloud despite the increased number of patients from 100 to 300 as more units are gradually involved from stage 1 utilising only the local GP and the cloud to the final stage, including all the

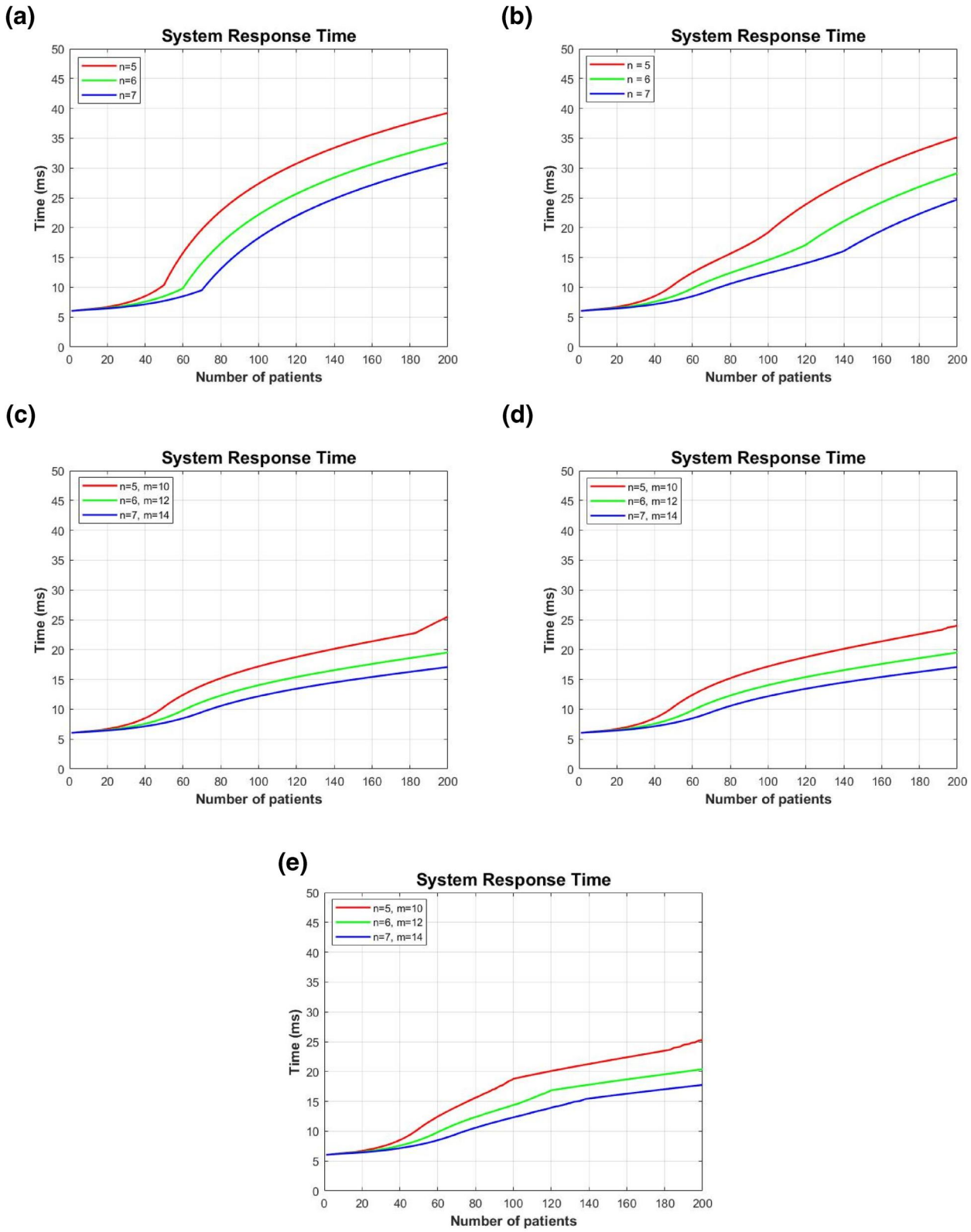

FIGURE 7 SRT of 200 patients in the expected different scenarios. (a) Patient → Local general practice (GP) → Cloud. (b) Patient → Local GP → GP neighbour → Cloud. (c) Patient → Local GP → Local general H (GH) → Cloud. (d) Patient → Local GP → Local GH → GH neighbour → Cloud. (e) Whole system

cooperative units. For example, in responding to a 300-patient workload with n = 6 and $m = 12$, moving from stage 1 to stage 5, the numbers of patients who reached the cloud are 240, 180, 60, 0, and 0. Specifically, we obtained a ubiquitous and scalable health system with high service availability, more computing capacity, less latency, and better privacy.

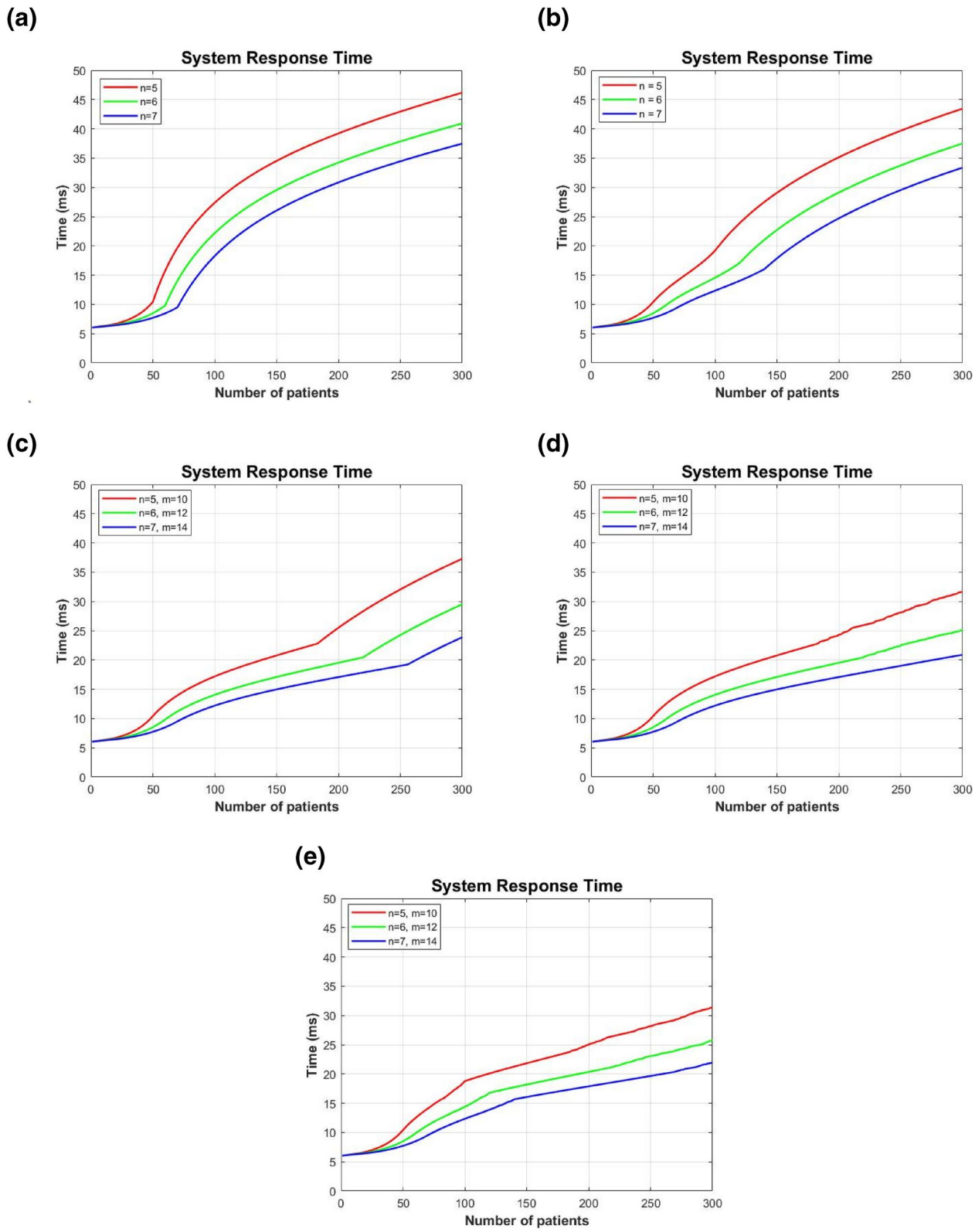

FIGURE 8 SRT of 300 patients in the expected different scenarios. (a) Patient → Local general practice (GP) → Cloud. (b) Patient → Local GP → GP neighbour → Cloud. (c) Patient → Local GP → Local general H (GH) → Cloud. (c) Patient → Local GP → Local GH → Cloud. (d) Patient → Local GP → Local GH → GH neighbour → Cloud. (e) Whole system

TABLE 3 Number of patients served in each architecture unit at every deployment stage (purple indicates unused units at a certain stage)

		No. patients served at														
		Local GP			GP neighbours			Local GH			GH neighbours			Cloud		
Stage	No. of patients	$n = 5$	$n = 6$	$n = 7$	$n = 5$	$n = 6$	$n = 7$	$m = 10$	$m = 12$	$m = 14$	$m = 10$	$m = 12$	$m = 14$	$n = 5$ $m = 10$	$n = 6$ $m = 12$	$n = 7$ $m = 14$
1	100	50	60	70										50	40	30
	200	50	60	70										150	140	130
	300	50	60	70										250	240	230
2	100	50	60	70	50	40	30							0	0	0
	200	50	60	70	50	60	70							100	80	60
	300	50	60	70	50	60	70							200	180	160
3	100	50	60	70				50	40	30				0	0	0
	200	50	60	70				133	140	130				17	0	0
	300	50	60	70				133	160	187				117	80	43
4	100	50	60	70				50	40	30	0	0	0	0	0	0
	200	50	60	70				133	140	130	17	0	0	0	0	0
	300	50	60	70				133	160	187	117	80	43	0	0	0
5	100	50	60	70	50	40	30	0	0	0	0	0	0	0	0	0
	200	50	60	70	0	0	0	133	140	130	17	0	0	0	0	0
	300	50	60	70	0	0	0	133	160	187	117	80	43	0	0	0

8 | LIMITATIONS AND FURTHER IMPROVEMENTS

Finally, this study has some limitations that the developed system anticipates data equally without taking urgency into account, which causes delays in the treatment of critically ill patients. By classifying the patient's data into multiple classes according to the patient's condition at edge servers, it would be easier to identify the most severe patients. Thus, giving them priority in receiving services, whether sending an ambulance, preparing necessary medical staff etc. Therefore, the main objective of our next study is to improve this work to build an intelligent healthcare monitoring system that can dynamically regulate patient data flow based on each patient's individual health status.

9 | CONCLUSION

This work proposed a cooperative hierarchical healthcare architecture, named HMAN, with Edge/Fog computing. The suggested architecture is supported by the HOSSC algorithm that provides five offloading and processing scenarios and facilitates SRT calculating. The architecture consists of four layers, namely, IoT, Edge/Fog, cloud, and application layers. Model analysis and thorough experimentation revealed the benefits of this architecture. The research confirmed the soundness of the suggested architecture and provided a method for measuring the system performance under various network workloads. This study investigated various scenarios to ensure that the data are locally processed to meet the objectives of more privacy and less latency. Five different stages have been established to aid in the evaluation of the presented architecture. The proposed architecture achieved a robust and scalable healthcare system exploiting the existing infrastructure in the city with low latency, ranging from 6.043 to 31.45 ms, considering the various workloads. According to the results, the proposed hierarchical architecture achieved inherited computational capacity, higher system scalability and greater availability. Moreover, the proposed architecture assured higher probability of local data processing in addition to the achieved less latency and higher privacy. The findings clearly showed that the suggested HMAN architecture is efficient enough for use in the healthcare domain. Nevertheless, some challenges in realising such architecture can be anticipated to include the prioritisation of the critical cases of some patients to provide an exceptional allocation of the healthcare resources for them. Finally, this study can be extended by utilising modern machine learning approaches (e.g. deep learning) in recognising diverse traffic and determining the optimal placement for GP servers in a certain area to improve the performance and reduce the deployment cost.

AUTHOR CONTRIBUTIONS

Ahmed M. Jasim: Conceptualisation, Data Curation, Formal Analysis, Methodology, Resources, Software, Validation, Visualisation, Writing – Original Draft Preparation. **Hamed Al-Raweshidy**: Funding Acquisition, Investigation, Project Administration, Resources, Supervision, Validation, Writing – Review & Editing

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

DATA AVAILABILITY STATEMENT

Data is available on request from the authors.

ORCID

Ahmed M. Jasim https://orcid.org/0000-0001-9276-577X

REFERENCES

1. Sikarwar, R., Yadav, P., Dubey, A.: A Survey on IoT enabled cloud platforms. In: 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), pp. 120–124 (2020). https://doi.org/10.1109/CSNT48778.2020.9115735
2. Al-Anbagi, H.N., Haidar, N., Vertat, I.: Cooperative reception of multiple satellite downlinks. Sensors. 22(8), 2856 (2022). https://doi.org/10.3390/s22082856
3. Makkar, A., Ghosh, U., Sharma, P.K.: Artificial intelligence and edge computing-enabled web spam detection for next generation IoT applications. IEEE Sensor. J. 21(22), 25352–25361 (2021). https://doi.org/10.1109/JSEN.2021.3066492
4. Qadri, Y.A., et al.: The future of healthcare internet of things: a survey of emerging technologies. IEEE Commun. Surv. Tutor. 22(2), 1121–1167 (2020). Secondquarter 2020. https://doi.org/10.1109/COMST.2020.2973314
5. Facts, Factors: Internet of things (IoT) market size, share—global forecast to 2026: Facts & Factors. Facts and Factors. Retrieved March 18, 2022, from https://www.fnfresearch.com/global-internet-of-things-iot-market-by-software-792
6. Pace, P., et al.: An edge-based architecture to support efficient applications for healthcare industry 4.0. IEEE Trans. Ind. Inf. 15(1), 481–489 (2019). https://doi.org/10.1109/TII.2018.2843169
7. Javaid, S., et al.: Medical sensors and their integration in wireless body area networks for pervasive healthcare delivery: a review. IEEE Sensor. J. 22(5), 3860–3877 (2022). https://doi.org/10.1109/JSEN.2022.3141064
8. Zhao, Y., et al.: Edge computing and networking: a survey on infrastructures and applications. IEEE Access. 7, 101213–101230 (2019). https://doi.org/10.1109/ACCESS.2019.2927538
9. Yu, W., et al.: A survey on the edge computing for the internet of things. IEEE Access. 6, 6900–6919 (2018). https://doi.org/10.1109/ACCESS.2017.2778504
10. Tong, L., Li, Y., Gao, W.: A hierarchical edge cloud architecture for mobile computing. In: IEEE INFOCOM 2016 - the 35th Annual IEEE International Conference on Computer Communications, pp. 1–9 (2016). https://doi.org/10.1109/INFOCOM.2016.7524340
11. Corchado, J.M., Trabelsi, S.: Advances in sustainable smart cities and territories. Electronics. 11(8), 1280 (2022). https://doi.org/10.3390/electronics11081280
12. Amirthalingam, K.: Medical dispute resolution, patient safety and the doctor-patient relationship. Singap. Med. J. 58(12), 681–684 (2017). https://doi.org/10.11622/smedj.2017073
13. Balansard, I., et al.: Revised recommendations for health monitoring of non-human primate colonies (2018): FELASA working group report. Lab. Anim. 53(5), 429–446 (2019). https://doi.org/10.1177/0023677219844541
14. Asif-Ur-Rahman, Md., et al.: Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things. IEEE Internet Things J. 6(3), 4049–4062 (2019). https://doi.org/10.1109/JIOT.2018.2876088

15. Ahmad, M., et al.: A novel framework for health and wellness applications. J. Supercomput. 72(10), 3677–3695 (2016). https://doi.org/10.1007/s11227-016-1634-x
16. Omar, A., et al.: Privacy-friendly platform for healthcare data in cloud based on block chain environment. Future Generat. Comput. Syst. 95, 511–521 (2019). https://doi.org/10.1016/j.future.2018.12.044
17. Muhammed, T., et al.: UbeHealth: a personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities. IEEE Access. 6, 32258–32285 (2018). https://doi.org/10.1109/ACCESS.2018.2846609
18. Abdelmoneem, R.M., et al.: A cloud-fog based architecture for IoT applications dedicated to healthcare. In: ICC 2019—2019 IEEE International Conference on Communications (ICC), pp. 1–6 (2019). https://doi.org/10.1109/ICC.2019.8761092
19. Rahmani, A.M., et al.: Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: a fog computing approach. Future Generat. Comput. Syst. 78, 641–658 (2018). https://doi.org/10.1016/j.future.2017.02.014
20. Kai, C., et al.: Collaborative cloud-edge-end task offloading in mobile-edge computing networks with limited communication capability. In: IEEE Transactions on Cognitive Communications and Networking. 7(2), 624–634 (2021). https://doi.org/10.1109/TCCN.2020.3018159
21. Nguyen, D.C., et al.: BEdgeHealth: a decentralized architecture for edge-based IoMT networks using block chain. IEEE Internet Things J. 8(14), 11743–11757 (2021). https://doi.org/10.1109/JIOT.2021.3058953
22. Rajasekaran, M., et al.: Autonomous monitoring in healthcare environment: reward-based energy charging mechanism for IoMT wireless sensing nodes. Future Generat. Comput. Syst. 98, 565–576 (2019). https://doi.org/10.1016/j.future.2019.01.021
23. Azimi, I., et al.: Self-aware early warning score system for IoT-based personalized healthcare. In: eHealth 360o, pp. 49–55. Springer, Cham, Switzerland (2017)
24. Muhammad, G., et al.: Smart health solution integrating IoT and cloud: a case study of voice pathology monitoring. IEEE Commun. Mag. 55(1), 69–73 (2017). https://doi.org/10.1109/MCOM.2017.1600425CM
25. He, S., et al.: Proactive personalized services through fog-cloud computing in large-scale IoT-based healthcare application. China Commun. 14(11), 1–16 (2017). https://doi.org/10.1109/CC.2017.8233646
26. Verma, P., et al.: FETCH: a deep learning-based fog computing and IoT integrated environment for healthcare monitoring and diagnosis. IEEE Access. 10, 12548–12563 (2022). https://doi.org/10.1109/ACCESS.2022.3143793
27. Stafford, M., et al.: Understanding the Health Care Needs of People with Multiple Health Conditions (2018). [online] The Health Foundation. Available at: https://www.health.org.uk/publications/understanding-the-health-care-needs-of-people-with-multiple-health-conditions> Accessed 9 March 2022
28. Yarnall, K.S., et al.: Primary care: is there enough time for prevention? Am. J. Publ. Health. 93(4), 635–41 (2003). https://doi.org/10.2105/ajph.93.4.635. PMID: 12660210; PMCID: PMC1447803
29. Østbye, T., et al.: Is there time for management of patients with chronic diseases in primary care? Ann. Fam. Med. 3(3), 209–14 (2005). https://doi.org/10.1370/afm.310. PMID: 15928223; PMCID: PMC1466884
30. Lin, L., et al.: Computation offloading toward edge computing. In: Proceedings of the IEEE, vol. 107(8), pp. 1584–1607 (2019). https://doi.org/10.1109/JPROC.2019.2922285
31. Kleinrock, L.: Queueing Systems, vol. 1, pp. 101–103, NJ, USA (1975)

How to cite this article: Jasim, A.M., Al-Raweshidy, H.: Towards a cooperative hierarchical healthcare architecture using the HMAN offloading scenarios and SRT calculation algorithm. IET Netw. 12(1), 9–26 (2023). https://doi.org/10.1049/ntw2.12064