ISSN: 2766-9823 Volume 2, 2020

Fog Assisted and IoT Based Real-Time Health Monitoring System Implementation

Umme Afifa Jinan
Department of Information and Communication Engineering
Noakhali Science and Technology University
Noakhali, Bangladesh
afifajinanict@gmail.com *anization*

Ashikur Rahman Khan Given
Department of Information and Communication Engineering
Noakhali Science and Technology University
Noakhali, Bangladesh
ashik@nstu.edu.bd

Zayed-Us-Salehin
Department of Information and Communication Engineering
Noakhali Science and Technology University
Noakhali, Bangladesh
salehin.fahad@yahoo.com

Abstract— **With the proliferation of IoT devices in medical healthcare systems, IoT based health-monitoring systems and applications have brought about a ground-breaking breakthrough in modern healthcare facilities, medical data processing, and analysis. Meeting the challenge of co-operative and distributed IoT based healthcare systems; the problems like latency, network congestion, and data traffic in the systems can be overcome by fog computing, a decentralized cloud computing platform. In this study, we enhanced such an IoT-enabled real-time patient health monitoring system by exploiting the fog computing concept for extracting sensor data, visualizing them at reduced cost and power, storing at local storage, monitoring, and interacting remotely. Using three different sensor devices that extract health data, we developed a new type of fog computing interface using a combination of Raspberry Pi and Arduino. A dedicated local server called fog server was implemented as well for the storage and maintenance of the sensor extracted data and real-time notification. The implemented system portraits the efficacy of our proposed concept for monitoring patients' vital parameters like temperature, pulse rate, and ECG simultaneously at low-cost, low power, simpler scheme, and real-time remote monitoring. The comparison between medical data, biosignals and sensor generated data, signals showed the feasibility of the system. The performance and compatibility of the system was evaluated in terms of cost, power consumption, and latency as well. The development and testing of the health monitoring system proves that it serves perfect in giving relief to medical caregivers while taking care of patients remotely irrespective of time and place.**

Keywords— Arduino, Fog Computing, Health Monitoring, IoT, Raspberry Pi, Real-Time Monitoring

I. Introduction

Healthcare is always considered a very complex system and with the increase in population and patient surge; the medical care demand is increasing in an unprecedented way. Hence, the importance of transforming the healthcare system into smart healthcare is evolving to cope with its increasing technological demand and aspects [1]. The 21st-century Healthcare Information Technology (HIT) has enabled us to electronically store, maintain, and move data electronically across the world in a matter of seconds. It has the potential to provide healthcare with tremendous increasing productivity and quality of services as well [2].

Several smart healthcare systems using health sensors and cloud server have been proposed to provide health monitoring and support service to the person in concern anytime, anywhere [3]. Besides, the emergence of the Internet of Things (IoT) as new technology increases the need for cloud computing services and enhances the role of cloud computing in business and technology [4]. The incorporation of IoT in healthcare has thriven to tremendous improvement in efficiency, scalability, affordability, security, quality of service, and better outcome; with or without human intervention. There are estimates that healthcare data produced by IoT-based systems will be 25000 petabytes in 2020 [5]. However, cloud-IoT driven infrastructure cannot solve the problems of latency, data traffic, network congestion, system complexity, cost, and real-world deployment. Therefore, an additional layer that brings computing and intelligence close to the IoT data sources is introduced by CISCO, known as 'Fog'. Fog computing is suitable in applications that demand low latency such as health care since it supports mobility, computational resources, integration with the cloud, heterogeneous interface, distributed data analytics, etc [6]. Fog contributes to reducing latency, transmission power, network usage, data traffic, balancing the load between resources and increasing throughput, and enhancing the quality of service in a distributed computing and networking environment while reducing healthcare costs. A combination of fog computing and IoT systems using smart gateways and wearable devices can be a sustainable solution for existing challenges in remote continuous health-monitoring systems [7].

In fog-IoT based healthcare system architecture, the lower layer includes medical sensors and sensor devices like electrocardiogram (ECG) sensor, EMG sensor, EEG sensor, temperature sensor, blood pressure sensor, smartwatch, etc. The centrepiece of Fog computing is a low-power, intelligent, wireless, embedded computing node that carries out signal conditioning and data analytics on raw data collected from wearables or other medical sensors [8]. Fog uses network switches, routers, gateways, and mobile base stations to provide cloud service with minimum possible network latency and response time [9]. The upper cloud layer is for heavy computation, long term larger storage, and wide geographical distribution [10]. The architecture is shown in Fig. 1. The compatible wireless protocols in the system are Bluetooth, Zigbee, WiFi, IEEE 802.11, 2G/3G/4G, WiMax, 6Lowpan, TCP/ IP, etc. It is noted that extracted biosignals and data generated from sensors can be stored in fog local storage/ database. Push Notification for informing emergent is one of the most important services of the fog. It allows the doctors or medical caregivers to be notified about any

Identify applicable funding agency here. If none, delete this text box.

abnormality of patients in real-time through emails or and text messages in case of emergencies [7].

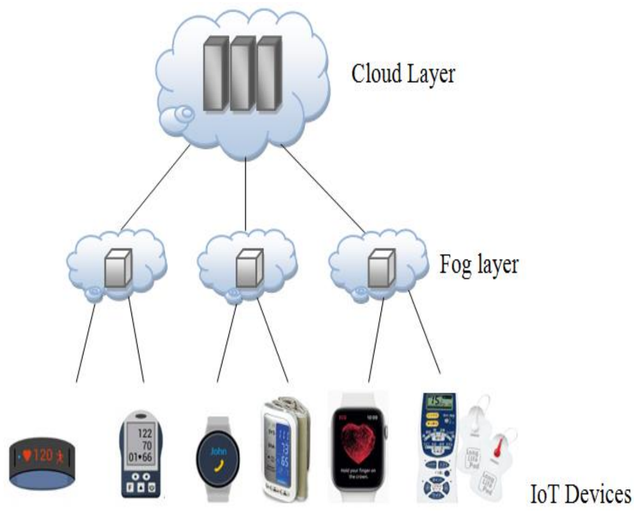

Fig. 1 Fog enabled IoT based system architecture

Among several types of fog computing architecture discussed in [11], the device type deployment scenario in healthcare is taken into consideration. Fog computing that uses embedded systems holds great promise to reduce the burden of the complex medical big data by acting as a local gateway (close to the sensor nodes) for communication, computation, storage, networking, and other functions [12, 13]. Conventional health-monitoring systems like ECG monitoring often have many drawbacks, such as non-ubiquitous access to data and discontinuous monitoring [7]**.** Our study emphasizes on simpler, easily buildable and maintainable, low-cost, energy-efficient, real-time and remote monitoring demonstration of fog-IoT based health monitoring system. A new type of fog interface with the combination of Raspberry Pi and Arduino is proposed and implemented here. Moreover, this study introduces an efficient method of extracting and monitoring ECG signals included in the system. In several related works, Raspberry Pi, Intel Edison, Pandaboard, Orange Pi were chosen as the main fog computer for taking bodily parameters, data processing, and analysis. However, focusing on low-cost, energy-efficiency, real-time monitoring capabilities with a simpler and user-friendly establishment, there seems a little gap in system implementation and applications. As discussed in other works, Raspberry Pi is the key element to work as the fog computer and serves perfect in taking data from different types of sensor devices and performing computations. However, while extracting the ECG signals, Raspberry produces redundant and incorrect signals. Instead, a simple microcontroller called Arduino can extract more accurate ECG signals faster and efficiently. We establish a new type of fog interface by incorporating Raspberry Pi and Arduino for extracting data and signals from the temperature sensor, pulse rate sensor, and ECG sensor. While the Raspberry extracts temperature and heart rate data, Arduino with the help of Raspberry will extract ECG signals. And we implement only the fog local storage which is enough for storing the textual temperature and pulse rate data. Along with extracting ECG signals, Arduino incorporated with Raspberry can meet the requirement of sending real-time notification (push notification) to the medical caregivers efficiently in real-time. Thus, it can meet the most anticipated criteria of fog based real-time health monitoring services. The data comparison measurements of the sensor generated data and biosignals of the implemented system with actual medical data and signals respectively proves the feasibility of the system. The performance of the fog computing node in terms of cost, power consumption, and latency will validate the efficacy of the system as well.

II. Problem formulation and solution

A. Problem Formulation

To present the creation of a simple, low-cost IoT-based healthcare system that uses fog computing, Raspberry Pi (RPi) was chosen as a key building element based on its small, cheap, powerful, and fully customizable properties by Maksimovic [14]. A complete system using temperature, heart rate, and blood pressure sensors, RPi with a gateway (fog server), a cloud was proposed to build custom IoT-based healthcare solutions. To demonstrate the benefits of fog computing in IoT systems, Gia and Jiang implemented a secure system that included wearable sensor nodes, smart gateways with fog services, and cloud servers with end-user terminals [7]. Smart gateways were built with a combination of Pandaboard, HC05 Bluetooth, nRF24L01, Smart-RF06 board, etc. But fog architecture using Pandaboard seems more complex compared to other establishments that use Raspberry. The same telehealth services can be provided with a relatively simpler fog scheme. Dubey et al. implemented and tested a fog computing system using Intel Edison and Raspberry Pi for various medical data including pathological speech data, Phonocardiogram (PCG) signal for heart rate estimation, and ECG-based Q, R, S detection [8]**.** Along with fog computers, smartwatch as BSN, gateways e.g. smartphones/tablets, and secure cloud backend were implemented. The case studies justified that computations and analysis on Intel Edison reduces the data by 99%. However, according to authors, surveying the useful tools for telehealth monitoring was time-consuming. Besides, establishing necessary hardware and programming on the Intel Edison fog was extremely difficult, therefore system complexity and compatibility issues arose. Since the raw data was not communicated to the cloud, there was no way to perform additional analysis in the cloud. Borthakur et al. presented a fog architecture that relied on unsupervised machine learning big data analysis for discovering patterns in physiological data [15]**.** They developed a prototype using Intel Edison and Raspberry Pi that was tested on real-world pathological speech data obtained from smartwatches for telemonitoring of patients with Parkinson's disease.

Gia et al. enhanced a health monitoring system by exploiting the concept of fog computing at smart gateways that provided advanced techniques and services such as embedded data mining, distributed storage, and notification service at the network edge [16]**.** They implemented fog based smart gateway using Pandaboard, Arduino, Zigduino, wireless and Ethernet modules, etc. They chose ECG feature extraction as a case study analyzed in gateways. Their experimental results showed more than 90% bandwidth efficiency and low-latency real-time response at the network edge. But considering the fog gateway architecture, the system would seem less complex by using Raspberry as it has built-in wireless protocols and Ethernet support; comparatively low cost as well. Constant et al. presented an end-to-end architecture and developed a prototype of smart fog gateway using Intel Edison and Raspberry Pi, wearable smart e-textile gloves using Arduino and flex sensors, and a cloud backend for long-term data storage [17]. Results demonstrated the potential of the proposed architecture for converting the real-world data into useful analytics while making use of knowledge-based models. Cao et al. developed new fall detection algorithms and employed a real-time fall detection system for stroke mitigation employing fog computing paradigm [18]. It split the detection task between the edge devices (e.g., smartphones attached to the user) and the server (cloud). Experimental results show that distributed analytics and edge intelligence supported by fog paradigm, are very promising solutions for pervasive health monitoring.

In another article, a low-cost health monitoring system has been carried out that comprises energy-efficient low-cost sensor nodes, low-power, and low-cost fog gateway, and a secure back-end system consisting of cloud and an end-user application [19]. The designed sensor node collects and sends wirelessly transmitted ECG, respiration rate, temperature, and humidity data to the Orange Pi (fog gateway). It has advanced services like real-time push notification for immediate attention. But the performance of the Orange Pi as a fog node was not specified properly. Dubey, H. et al. proposed, validated, and evaluated *Fog Data,* a service-oriented architecture for Fog computing using Intel Edison for monitoring Parkinson's disease data and ECG signals collected from wearable sensors [20]. The obtained results from case studies for speech motor disorder and real-time ECG monitoring showed substantial improvement in system efficiency using the *Fog Data* architecture. A method of fog based networking is established using Arduino boards with temperature sensors and with Raspberry pi as the fog network device [21]. For 1Mbps bandwidth, the result showed a better response time of fog based architecture compared to the cloud architecture. Molina, et al. developed a system similar to the system mentioned in based on IP networking but for monitoring heart rate with only the Arduino controller [21, 22]. Rahmani, et al. exploited the concept of fog computing in healthcare IoT systems by implementing a prototype of Smart e-Health Gateway (UT-GATE) and IoT-based Early Warning Score (EWS) monitoring [23]. UT-GATE was constructed from a combination of Pandaboard, SmartRF06 board, and several wireless modules for processing ECG signal for movement artifact removal and feature extraction. Results showed energy efficiency for about 55.7% and sample size reduction to 74.1%. Some other authors justified the validation of fog computing through simulation using iFogSim simulator for Fog-based IoT-Healthcare solution or fog enabled cloud environment in IoT healthcare [9, 24]. Others proposed only a conceptual architecture of incorporating fog as the middle layer in IoT-cloud driven healthcare monitoring establishments [10, 25].

Overall, all the previous works aimed at establishing efficient fog-IoT based systems for smart homes and smart hospitals. Some studies were conducted with different sensors and computing devices while some others used software and simulation techniques to validate their proposed system architectures. Some works were carried out using predefined medical data and some studies worked on real world data taken synchronously at that time. However, the implementation of a relatively simpler system with real time sensory data is missing. Moreover, there was no synchronized server system developed that would store data continuously exactly at the time they were produced at sensor ends. The intermediary fog devices seemed sophisticated even though they were compatible in heavier processing and analysis. When focusing on simple sensor data like temperature and pulse rate, a lightweight simple yet effective fog node comprising of Raspberry Pi can be sufficient for processing sensor extracted data. The works that focused on only ECG signal processing and feature extraction considered much complex embedded devices that are not user-friendly or popular in ECG signal processing. The process of push notification in previous works needs more effective method while considering real time ECG signals.

Thus, this research work aims to establish a real-time remote health monitoring application to demonstrate the effectiveness and efficacy of fog computing in IoT driven health monitoring system. It is intended to develop a new type of Fog computing interface emphasizing on low cost, low power consumption, simpler arrangement, user-friendliness, easier maintenance, and high efficiency. The real world implementation based on our concept justifies this development for a custom IoT based healthcare solution.

B. Problem Solution

Fog layer works in collaboration with the IoT framework to provide real-time health monitoring services effectively. The fog computer needs to perform faster, be cost-effective, efficient, and user-friendly, and be compatible with IoT sensors devices and backend storage. Besides, it is crucial for the system to provide advanced services such as push notification for reporting in case of emergencies like during ECG monitoring to corresponding personnel in real-time. To demonstrate all these requirements, a complete system using simple, low-cost sensor devices, efficient fog computing interface with a combination of Raspberry Pi and Arduino, and gateway (fog server) with flexible storage and an end-user terminal is implemented in our work. We measure the performance of the system by comparing the sensor generated data with medical data and the biosignals. The performance of the fog computing interface in terms of cost, power consumption, and latency is also evaluated to prove the feasibility of the total system establishment.

III. System design and implementation

A. System Design

To validate the proposed architecture, we design and implement a system prototype based on Raspberry Pi and Arduino working together as fog computer connected end-to-end. With the developed system, we will carry out experiments using real-world data collected from the patient body through our chosen temperature, pulse rate, and ECG sensors. As the fog node is connected directly with the sensors, it will ensure local connectivity once placed. The embedded fog computer will extract data and biosignals through connected sensors and send them to the fog gateway for storage and display. However, cloud backend is not necessary for this implementation. The fog local storage will serve the purpose of data storage, maintenance, and access like the usual cloud server. Moreover, the system will provide one of the most crucial fog computing services i.e. real-time push notification by sending ECG signal plots continuously in real-time once it is generated at fog computer. The prototype of the proposed system is demonstrated in Fig. 2.

To measure the patient's vital parameters, three sensors namely temperature sensor, pulse rate sensor, and ECG sensor are chosen. While temperature and pulse rate sensors provide textual data, the ECG sensor will provide signal data or biosignals. After attaching the sensors properly and coming in touch with the body, sensors will start extracting data and continuously send to Raspberry Pi and Arduino. Raspberry is responsible for taking temperature and pulse rate data. Arduino incorporated with Raspberry takes ECG signals as the Arduino is the best-suited system for extracting and plotting ECG signals accurately. As the system runs, it continuously takes in sensor-generated data and then sends it to the local server or fog server, also known as fog gateway. The local database stores all the textual data simultaneously as they are generated at the sensors through fog interface. The fog server to be implemented here is well synchronized with the fog computer. Healthcare service providers access the database for observing the patient's healthcare data through a secure user terminal with a username and password and can take any decision accordingly. Finally, when ECG signals are generated at the Arduino board and displayed at the Arduino platform, the plot is also sent to the designated medical caregiver as a video file. This push notification service through video files enables the service providers to observe the patient heart status in near real-time. This aspect of the system validates and justifies our methodology for meeting the real-time requirement of the fog computing concept.

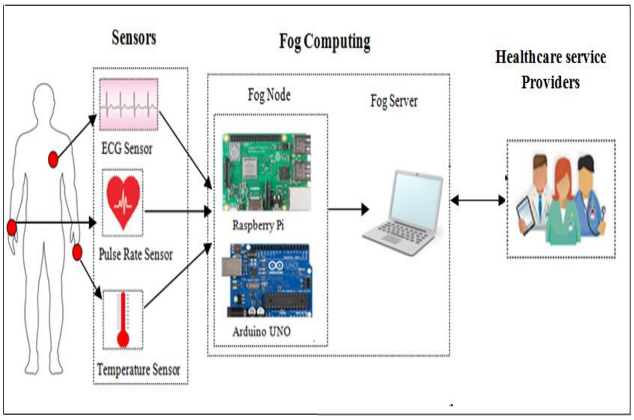

Fig. 2 Demonstration of fog enabled IoT based health-monitoring system

B. System Implementation

We implemented a patient health monitoring system based on the fog-IoT concept following the methodology discussed above. The three chosen sensor devices are LM 35 Temperature Sensor module, Pulse Rate Sensor, and AD8232 Sensor. LM 35 is a low-cost sensing module, low self-heating, and can measure surrounding temperature ranging from -55°C to 150°C. The Pulse Rate also called Heartbeat/ Heart rate Sensor measures the heart rate in beats per minute (BPM). This sensor has inbuilt amplification and noise cancellation circuit in it. The LED point of the sensor is placed over a vein in the human body to take measurements. The AD8232 sensor is used to measure the ECG signal. It extracts, amplifies, and filters small ECG signals. ECGs can be extremely noisy, the AD8232 single lead circuit acts as an op-amp to help obtain a clear signal from the PR and QT Intervals easily. The sensor is attached to the body through three ECG electrodes. The electrodes convert heart beat into an electric signal. LM35 sensor collects body temperature in as textual data every 0.1 second interval after the fingertip is put on the sensor. After putting a finger on the LED of the pulse sensor, it begins to extract pulse rates from the body and generate data in textual form. For ECG signal extraction, we placed one pad in the right of the heart, another pad in the left of the heart, and the last one around the kidney section or the liver to take signals accurately. We have used connecting wires to connect the sensors to the fog interface since fog supports both wired and wireless connectivity [10]. The practical demonstration of the implemented system is illustrated in Fig. 3.

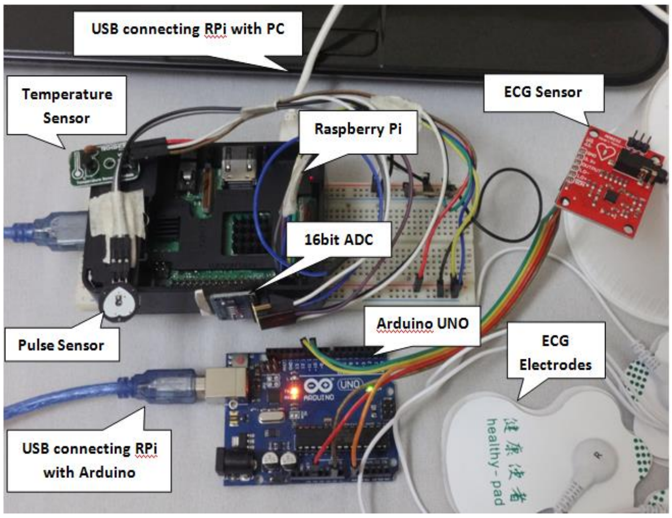

Fig. 3 Demonstration of fog-Iot based real-time health monitoring system

For implementing the fog interface, we have used the *Raspberry Pi model 3B+*. It is a 64-bit quad-core processor running at 1.4 GHz, dual-band 2.4 GHz and 5 GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet. The dual-band wireless LAN improved both cost and time. The temperature and pulse rate sensors are assembled with the Raspberry through connecting wires. However, since the GPIO of Raspberry Pi only allows digital signal inputs and has no built-in A/D converter, a higher-precision ADS 1115 16bit I2C ADC+PGA analog to digital (A/D) converter is integrated into the fog circuitry. 16 GB SD card is used to store the operating system and run the device. We have used Debian, a Linux Distribution, for the OS of Raspberry. NOOBS, an operating system installation manager, found on raspberrypi.org site is stored on the SD card as well. Among several operating systems from NOOBS, we have used Raspbian, a free operating system based on Debian, and optimized for the Raspberry Pi system. Python 3.2 programming IDE and Matlab is used for writing the codes. ***Arduino Uno*** is a microcontroller board based on the ATmega328P. This is connected to the Raspberry via a USB cable. The Arduino IDE and Processing IDE software for running the Arduino are installed in the Raspberry Pi so that the Arduino can be operated from the Raspberry Pi. The ECG sensor board is assembled with the Arduino board through connecting wires. As the system is powered on, the Arduino starts fetching ECG signals from ECG sensors through electrodes placed on the patient body and signals can be displayed on Arduino serial plotter. We have activated the real-time notification service (fog service) for ECG signals sent through email as an .mp4 video file format. The real-time health data extraction and plotting for ECG signals with a demo patient is shown in Fig 4.

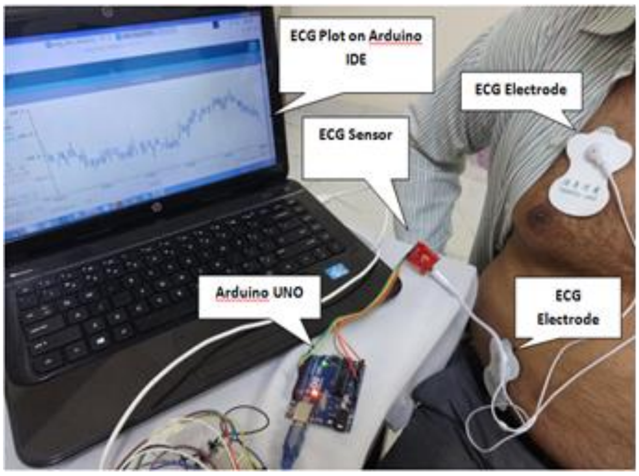

Fig. 4 Real-time health monitoring and ECG plotting with the developed system

To support the storage of clinical data, we implemented a backend database using PHP and MySQL. First, we set up a Linux, Apache, MySQL, PHP (LAMP) server for web servicing, MySQL as a database system for management and storage, and PHP as the language for server interaction with applications. We used a PC for accessing the server but there are other ways of operating and displaying data through a display monitor as well. In that case, the monitor needs to be connected via the HDMI port of the Raspberry Pi. Fog server hosts a webpage having an easy-to-use user terminal. The webpage is built using Python, HTML5, CSS, XML, JavaScript, and JSON. Upon entering username and password, end-user like doctors, medical caregivers having valid credentials can connect to the webpage and access the data in real-time. Our implemented fog server performs similar to the conventional cloud server and usually in such cases; the database is not synchronized with the cloud servers' databases except for the back-up cases [7]**.** Therefore, an additional cloud backend is not needed for data storage.

Moreover, the real-time notification service is activated as soon as the Arduino extracts and sends the ECG signal plot to the fog server via Email service. The signal is transmitted as an .mp4 video file format to the designated email address following SMTP protocol.

IV. Results, Analysis and Discussion

A. *Results and Analysis*

Through the implementation of this system, we can observe the real-world sensor extracted data continuously generated in real-time. The body temperature data are generated and displayed after the patient touches the temperature sensor for a while. Accordingly, the generated pulse rate data are produced after the patient place finger on the LED of the pulse sensor. The temperature and pulse data altogether are shown in Fig. 5. The first two temperature data show 17 and 25 degrees Celsius which are environment temperatures and the third output shows the body temperature 31 degrees Celsius. Similarly, the pulse is generated continuously as shown 40, 38, 35 BPM in the figure. The data keeps generated continuously as long as the patient holds the sensor.

Fig. 5 Temperature and pulse rate data generated by sensors

After placing the three electrodes attached with the ECG sensor are placed on the body and the ECG signals are generated continuously, the Arduino serial plotter lets us visualize the ECG signals as shown in fig 6. For activating the real-time push notification service, we created the email alert system which enables the service provider to observe the ECG signals remotely in real-time. After the ECG signals are generated, the signal plot is converted into a video file of 6-7 seconds in .mp4 video file format and sent to the designated email address of the medical caregivers following SMTP protocol. The video file for real-time email alert service is generated along with the detailed file structures like frame number in seconds (fps), data upload speed, time, and bit rate in seconds (kbps). This helps us to understand the video file better from a technical perspective.

The Email Notification for ECG Report which proves the delivered status of the video file as illustrated in fig. 7. The email along with patient details is sent from the server to the designated recipient.

As per our implemented system, the data generated by temperature and heart rate sensors are stored in the implemented fog server simultaneously. To represent the data with necessary details, we have created tables with an id, value, or the generated data, data types to differentiate between temperature and pulse rate data, and detailed date and time. This is flexible storage which can store numerous data exactly at the time they are created at the sensor end synchronously. The medical caregivers access the database with a username and password which in turn, ensure the security of the health data as well as the fog server scheme. A small section of the stored data in the database is featured in Fig. 8.

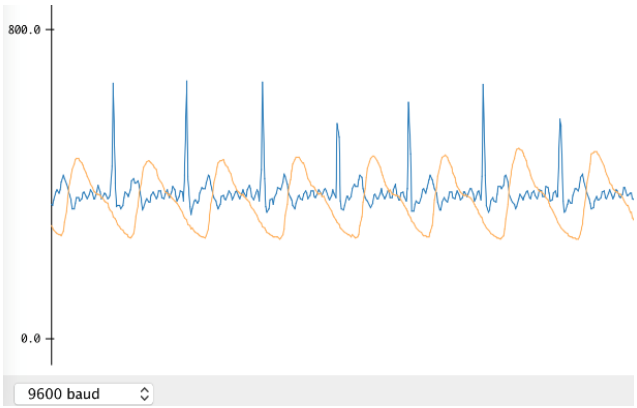

Fig. 6 ECG signals displayed in Arduino serial plotter

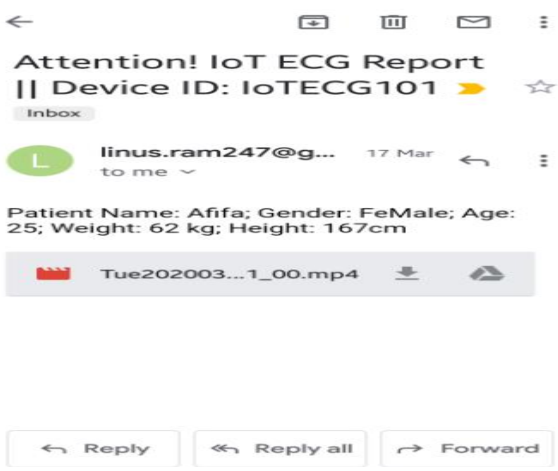

Fig. 7 Email Notification for ECG Report

The temperature and heart rate data stored in the implemented fog server is shown in Fig 8. The type name 'b' denotes body temperature and 'h' denotes heart rate. The measured data stored in our server at different time spans are justified in the date column. Data stored at different times during the runtime of the system in order to show the efficacy and flexibility of our implemented fog server.

Fog unit requires handling a large amount of sensory data continuously in a short time and response appropriately with respect to various conditions [23]. As we implemented a real time system, the performance of the overall system is emphasized on real time data delivery and notification at the receiver end with minimal latency (delay). So, the analysis is focused on the comparison of the extracted temperature and pulse data with medical data as well as the performance of the fog node or the overall system in terms of latency. The performance of the inter-communication among the layers is measured here as well. First, among continuously retrieved and stored thousands of data in the fog server, a total of 20 standard sample data are considered for comparing with the data taken from the patient body. The data comparison shows us the feasibility of the implemented system in monitoring patient health. The four different sets of temperature and pulse data are shown in fig. 9. We compare the actual temperature data with sensor generated temperature data and find the difference which is illustrated in a line graph as shown in fig. 10. We see the output lines are slightly varying with peaks and fluctuations almost similar for both sets of data. The pulse data are plotted in a similar method as a line graph to show the differences between the actual and sensor generated pulse rate data. This is illustrated in fig. 11. These difference in the illustrations validate the sensor data generated by our system as a feasible one.

id	value	type_name	created_date
13	45.0000	h	2020-03-03 00:00:00
14	40.0000	h	2020-03-03 00:00:00
15	80.0000	h	2020-03-03 00:00:00
16	80.0000	h	2020-03-03 00:00:00
17	86.0000	h	2020-03-07 01:46:33
18	100.0000	h	2020-03-07 03:05:34
19	71.0900	h	2020-03-07 03:07:17
20	66.1900	h	2020-03-07 03:10:21
21	96.1900	h	2020-03-07 03:10:39
22	96.0000	b	2020-03-07 04:35:34
23	56.0000	b	2020-03-07 04:35:39
24	50.0000	b	2020-03-07 04:35:43
25	70.0000	b	2020-03-07 04:35:46
26	70.0000	b	2020-03-07 04:46:32
27	40.0000	b	2020-03-07 04:46:48
28	70.0000	b	2020-03-07 04:46:52
63	78.0000	h	2020-03-16 08:37:35
64	88.0000	h	2020-03-16 08:44:09
65	88.0000	h	2020-03-16 14:00:03
66	78.0000	h	2020-03-16 14:00:20
67	0.0000	b	2020-03-16 14:41:17
68	0.0000	b	2020-03-16 14:42:43
69	56.9200	b	2020-03-16 14:44:07
70	53.6920	b	2020-03-17 01:40:54
1977	28.8600	h	2020-06-18 11:18:33
1978	28.8600	h	2020-06-18 11:18:33
1979	47.7740	b	2020-06-18 11:21:58

Fig. 8 Temperature and heart rate datasets stored in implemented fog server

As ECG signals are one of the essential parameters for monitoring patient health conditions, the fog-IoT based systems focused on generating ECG signals should produce almost accurate signals at the fog end. To check the performance of ECG signals generated by our system, we measured the ECG signal of a 45 years old male with a medical ECG machine. The resultant ECG waveforms have been considered for comparing with our ECG sensor generated waveforms. Now the ECG sensor generated signal at arduino and original ECG signal shows a bit dissimilar waveform. This is due to the noises and baseline wandering in the sensor generated signals. Through applying proper filtering like FIR filtering and removing all the noises like baseline wandering from the signals, we can generate the signal very close to the actual ECG signals. Our system focuses on the real time generation and delivery of the sensor generated data and signals at the receiver end. Hence the feasibility of the ECG sensor generated signals depends on the continuous generation and real time push notification of the plots uninterruptedly.

Serial No.	Actual Temperature©	Sensor Tempearature ©	Actual Pulse	Sensor Pulse
Patient 1	32	30	110	129
Patient 2	31	27	80	76
Patient 3	30	25	78	85
Patient 4	36	29	82	83
Patient 5	34	29	96	92
Patient 6	40	38	88	94
Patient 7	35	28	92	94
Patient 8	37	33	102	97
Patient 9	36	34	85	82
Patient 10	39	38	95	70
Patient 11	33	31	76	82
Patient 12	29	32	89	105
Patient 13	32	34	98	96
Patient 14	39	36	82	85
Patient 15	37	34	72	79
Patient 16	35	37	96	98
Patient 17	33	34	101	98
Patient 18	30	31	93	84
Patient 19	34	31	95	108
Patient 20	36	35	109	96

Fig. 9 Temperature and pulse data sets for comparison

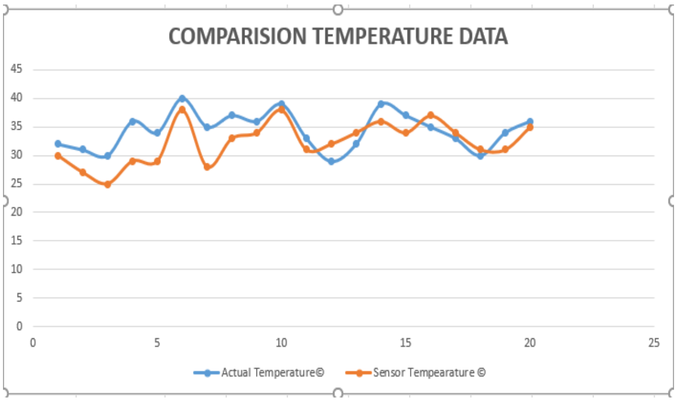

Fig. 10. Comparison of actual and sensor generated temperature data

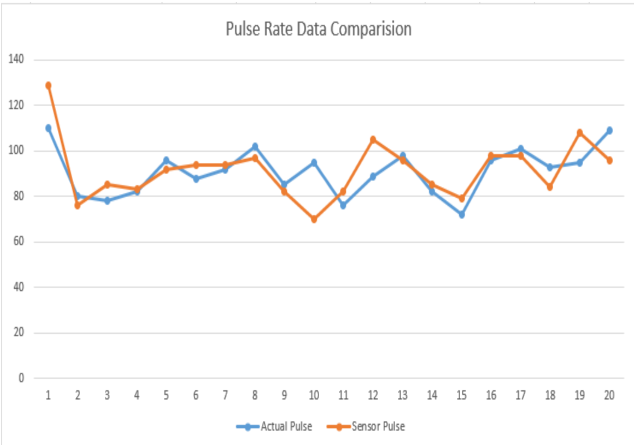

Fig. 11. Comparison of actual and sensor generated pulse rate data

The monitoring system is programmed in synchronization with each of the layers. The latency or delay between the retrieved sensor data and fog data sent to the server is monitored and the performance is evaluated accordingly as shown in table 1. The calculation of the delay justifies the efficacy of the fog computer in this system. We also measure the cost, power and energy required to run the system, along with latency.

TABLE I. PERFORMANCE EVALUATION IN TERMS OF COST, POWER, AND LATENCY

Cost ($)	Voltage (V)	Average Power (A)	Average Latency (s)	
100 dollars (approx.)	5	4.8	(Sensor to Fog) 0.1	(Fog to Server) 0.01

Table 1 shows us that this system is low-cost compared to other fog based health monitoring systems because of cheaper sensor nodes, fog devices, and other related components (around 100 dollars in total). The system draws a really small amount of power. If not connected directly to the PC, it requires 5V 4.8 A charger to run the system. Besides, the system can run by drawing small amount of power from PC when connected to the PC through a USB cord. The average time required to send a sensor data to the fog node and from fog node to the fog server is about 0.1 seconds and 0.01 seconds respectively. This means the delay is really low and the continuous delivery of data is real quick and near real time throughout the layers. This determines that the developed health monitoring system is feasible in checking patient vital parameters effectively.

B. *Discussion*

The health monitoring system is easier to establish through wired and easier connection with Raspberry Pi and Arduino. The dedicated fog local server is secure for health data storage and can be accessed from anywhere. Sending the real-world ECG signal as a video file format via email in real-time is a better push notification service. This method enables and validates remote monitoring as well. We have taken real-world data for demonstration that validates the efficacy of the system. However, taking real-life data also produces huge redundant data that must be acknowledged. Besides, the sensor modules used here are for development purposes and not for the actual diagnosis of patients. While incorporating such systems in hospitals, sophisticated medical sensors can be replaced instead of providing service. In the case of wireless devices, proper wireless and Ethernet modules can be inserted into the Raspberry. The server will run with any kind of Internet connectivity in compliance with fog node. In case of incorporating a cloud node into it, the system can be connected to the cloud server simply by uploading the local server data to the cloud database.

II. CONCLUSION

In this work, we established a simple, low-cost, low-power; smart real-time health monitoring system that effectively demonstrates all the requirements of an IoT-based healthcare system. The implementation and real-life demonstration prove the pervasiveness and ubiquity of the fog computing concept. We introduced a combination of Raspberry Pi and Arduino devices as a low-cost fog computing node that fetches, processes, and stores data generated from the sensor up to the fog server. It consumes low power and network usage, thereby enabling faster real-time health data delivery and remote monitoring. The established system is dedicated to physically challenged and elderly patients who barely move around hospitals for parameter measurements and find it harder to get treated accordingly in time. This system can highly assist medical caregivers, doctors, nurses by providing a way to monitor their patients remotely in real-time as well. In this way, the implemented patient health-monitoring system based on fog computing and IoT can provide as a way to alleviate many problems of existing healthcare systems and services in terms of patient status monitoring 24/7. It would also assist in healthcare data processing, storing, managing, and maintaining simultaneously in an efficient manner.

ACKNOWLEDGMENT

The authors would like to acknowledge to the Lab staff of Department of Information and Communication Engineering for their regular assistance. In addition, authors express their sincere gratitude to the Research Cell, Noakhali Science and Technology University for providing the grand to carry out this research work.

REFERENCES

[1] Oueida, S., Kotb, Y., Aloqaily, M., Jararweh, Y., & Baker, T., “An edge computing based smart healthcare framework for resource management,” Sensors, 18(12), 4307, 2018.

[2] Jinan, U. A., & Soheli, S. J., “Sharing of Patient Health Record (PHR) in Jelastic Cloud in a Secured Manner,” Asian Journal of Research in Computer Science, pp. 1-15, 2018.

[3] Tyagi, S., Agarwal, A., Maheshwari, P., “A conceptual framework IoT - based healthcare system using cloud computing,” 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016.

[4] Khater, B. S., Wahab, A. W. B. A., Idris, M. Y. I. B., Hussain, M. A., & Ibrahim, A. A., “A lightweight perceptron-based intrusion detection system for fog computing,” Applied Sciences, 9(1), 178, 2019.

[5] Feldman, B., Martin, E. M., & Skotnes, T., “Big data in healthcare hype and hope,” Dr. Bonnie, 360, 122-125, September 27, 2012.

[6] Biswas, S. “A Novel Fog and Cloud based Holistic Secure Framework for Smart Healthcare,” International Journal of Applications, 975, 8887, 2018.

[7] Gia, T. N., & Jiang, M., “Exploiting fog computing in health monitoring,” Fog and Edge Computing: Principles and Paradigms, pp. 291-318, 2019.

[8] Dubey, H., Monteiro, A., Constant, N., Abtahi, M., Borthakur, D., Mahler, L., & Mankodiya, K., “Fog computing in medical internet-of-things: Architecture, implementation, and applications,” In Handbook of Large-Scale Distributed Computing in Smart Healthcare (pp. 281-321). Springer, Cham, 2017.

[9] Gill, S. S., Arya, R. C., Wander, G. S., & Buyya, R., “Fog-based smart healthcare as a big data and cloud service for heart patients using IoT,” In International Conference on Intelligent Data Communication Technologies and Internet of Things, pp. 1376-1383. Springer, Cham, August, 2018.

[10] Nandyala, C. S., & Kim, H. K., “From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals,” International Journal of Smart Home, 10(2), pp. 187-196, 2016.

[11] Kraemer, F. A., Braten, A. E., Tamkittikhun, N., & Palma, D., “Fog computing in healthcare–a review and discussion,” IEEE Access, 5, 9206-9222, 2017.

[12] Monteiro, A., Dubey, H., Mahler, L., Yang, Q., & Mankodiya, K., “Fit: A fog computing device for speech tele-treatments,” In 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1-3, IEEE, May, 2016.

[13] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S., “Fog computing and its role in the internet of things,” In Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16, August, 2012.

[14] Maksimovic, M., “Implementation of Fog computing in IoT-based healthcare system,” Jita-Journal of Information Technology and Aplications, 14(2), 2018.

[15] Borthakur, D., Dubey, H., Constant, N., Mahler, L., & Mankodiya, K., “Smart fog: Fog computing framework for unsupervised clustering analytics in wearable internet of things,” In 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 472-476, IEEE, November, 2017.

[16] Gia, T. N., Jiang, M., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H., “Fog computing in healthcare internet of things: A case study on ecg feature extraction,” In 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing, pp. 356-363, IEEE, October, 2015.

[17] Constant, N., Borthakur, D., Abtahi, M., Dubey, H., & Mankodiya, K., “Fog-assisted wiot: A smart fog gateway for end-to-end analytics in wearable internet of things,” arXiv preprint arXiv: 1701.08680, 2017.

[18] Cao, Y., Hou, P., Brown, D., Wang, J., & Chen, S., “Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing,” In Proceedings of the 2015 Workshop on Mobile Big Data, pp. 43-48, June, 2015.

[19] Gia, T. N., Jiang, M., Sarker, V. K., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H., “Low-cost fog-assisted health-care IoT system with energy-efficient sensor nodes,” In 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1765-1770, IEEE, June, 2017.

[20] Dubey, H., Yang, J., Constant, N., Amiri, A. M., Yang, Q., & Makodiya, K., “Fog data: Enhancing telehealth big data through fog computing,” In Proceedings of the ASE big data & social informatics 2015, pp. 1-6, 2015.

[21] Krishnan, Y. N., Bhagwat, C. N., & Utpat, A. P., “Fog computing—Network based cloud computing,” In 2015 2nd International Conference on Electronics and Communication Systems (ICECS), pp. 250-251, IEEE, February, 2015.

[22] Molina, M. G., Garzón, P. E., Molina, C. J., & Nicola, J. X., “Heart Rate monitor based on IP networking,” In MATEC Web of Conferences (Vol. 210, p. 01006), EDP Sciences, 2018.

[23] Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., & Liljeberg, P., “Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach,” Future Generation Computer Systems, 78, pp. 641-658, 2018.

[24] Mahmud, R., Koch, F. L., & Buyya, R., “Cloud-fog interoperability in IoT-enabled healthcare solutions,” In Proceedings of the 19th international conference on distributed computing and networking, pp. 1-10, January, 2018.

[25] Khattak, H. A., Arshad, H., ul Islam, S., Ahmed, G., Jabbar, S., Sharif, A. M., & Khalid, S., “Utilization and load balancing in fog servers for health applications,” EURASIP Journal on Wireless Communications and Networking, 2019(1), 91, 2019.