DESIGN AND IMPLEMENTATION OF FOG ASSISTED IOT-BASED HEALTH MONITORING SYSTEM

UMME AFIFA JINAN

Thesis submitted in fulfillment of the requirements for the award of the degree of Master of Science in ICE

Department of Information and Communication Engineering

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

DECEMBER 2020

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this thesis and, in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Science in Information and Communication Engineering.

Professor Dr. Md. Ashikur Rahman Khan
Chairman,
Dept. of Information and Communication Engineering
Noakhali Science & Technology University
Sonapur – 3814, Noakhali, Bangladesh

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature

Name: Umme Afifa Jinan

ID Number: BKH1711MSc104F

Date:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Professor Dr. Ashikur Rahman Khan for his germinal ideas, invaluable guidance, continuous encouragement, and constant support in making this research possible. I also would like to express very special thanks to my co-supervisor Assistant Professor Zayed-us-Salehin for his suggestions and co-operations throughout the study.

My sincere gratitude goes to all the members of the staff of the Information and Communication Engineering Department, NSTU who helped me in many ways and made my stay at NSTU pleasant and unforgettable. Many special thanks go to the networking and programming research group for their excellent co-operation and support during this study.

I acknowledge my profound indebtedness and gratitude to my parents for their dream, love and sacrifice throughout my life. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions which were crucial for the successful completion of this study.

ABSTRACT

With the proliferation of IoT devices in healthcare systems and the health informatics arena, IoT based health-monitoring systems have brought about a groundbreaking breakthrough in this 21st-century health information technology, and medical data processing. Meeting the challenges of distributed IoT based healthcare systems implementation, cloud computing alone cannot contribute to the scenario with IoT effectively. Huge medical sensors generated data to be stored in a centralized cloud server may generate problems like latency, network congestion, and increased data traffic due to the overflow of data in the system. Hence, fog computing can overcome these problems by working as a middle layer in between IoT and cloud. This decentralized cloud computing platform facilitates intermediate data processing, analysis, storage, real-time delivery. 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 at a reduced cost and power, storing at local storage, monitoring, and interacting remotely in real-time. Using a couple of sensor devices that measure vital body parameters, we developed a new type of fog computing interface using Raspberry Pi and Arduino UNO. A dedicated local server was implemented as well for the storage and maintenance of the sensor extracted data and giving real-time notification. The implemented monitoring system portraits the efficacy of our proposed system architecture for monitoring patients simultaneously at low-cost, low power with simpler system establishment and real-time remote monitoring. The comparison between medical data and biosignals and sensor generated data and signals showed the feasibility of the system. The technical feasibility of the developed fog computing node in terms of cost, power consumption, and latency was evaluated as well. The development and testing of the health monitoring system proves that this system is intended to give relief to medical caregivers while taking care of patients remotely irrespective of time and place.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

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]. With the frequent use of mobile devices and other small gadgets like wearable sensors in daily life, today, tiny-sized sensors could be placed on the human body to record various physiological parameters [2]. Internet of Things (IoT) is in the modern era of Information Technology that comes with enabling usual regular objects to be intelligent. IoT projects are often developed using generic microcontroller-based development boards or single-board computers like the Raspberry Pi [3]. Centralized computing scenario of the cloud in IoT based system makes the IoT establishments slower, congests the network, and increases data traffic. It can be solved by a new decentralized computing paradigm referred to as Fog Computing that brings data processing close to the data source. To establish an efficient, real-time, low-cost, pervasive health-monitoring system that incorporates both IoT and fog schemes, a fog computing interface using Raspberry Pi and other embedded devices can be deployed. By this, the patient can be monitored uninterruptedly through the whole system establishment, thereby giving medical caregivers a relief for 24/7 health-monitoring of patients.

1.2 PROBLEM STATEMENT

Many works have been carried out to create a simple, low-cost, efficient patient health monitoring system by integrating IoT and fog computing concepts. Among several types of fog computing architecture discussed in [4], 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 [5] [6]. Conventional health-monitoring systems like ECG monitoring often have many drawbacks, such as non-ubiquitous access to data and discontinuous monitoring [7]. This 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 computing interface with a combination of Raspberry Pi and Arduino UNO 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 etc 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 just perfect in taking data from different types of sensor devices and performing computations. However, extracting the ECG signal correctly through an ECG sensor with only the Raspberry without any mistakenly taken data can be a little difficult to accomplish. In other words, along with using the Raspberry, there is another faster and efficient way to extract ECG signals through a microcontroller named Arduino. We establish a new type of fog interface by incorporating Raspberry Pi and Arduino for extracting data and signals from three selected sensor devices including temperature sensor, pulse rate sensor, and ECG sensor. While raspberry extracts temperature and heart rate data, Arduino connected to the Raspberry will extract ECG signals. And we implement only the local storage referred to as fog storage which is enough for storing the textual temperature and pulse rate data, yet the functionality is the same as a usual backend cloud system. Along with extracting ECG signals, Arduino incorporated with Raspberry Pi can meet the requirement of sending real-time data notification (push notification) to the medical caregivers in an efficient manner in real-time. In this manner, 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 bio signals of implemented system with actual medical data and signals respectively will prove 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.

1.3 OBJECTIVES

The objectives of this study are as follows:

i. To establish a real-time pervasive health monitoring application to demonstrate the efficacy of fog computing in the healthcare monitoring system

ii. To develop a new computer-aided Fog computing system emphasizing on low latency, low cost, low power consumption, and high efficiency

iii. To build a working prototype using Raspberry Pi embedded with Arduino UNO to demonstrate the efficiency of the new architecture

iv. To carry out telehealth monitoring experiments using three different biosensors, two integrated computing devices, and backend storage to show that the new system substantially increased the efficiency of IoT health monitoring system

v. To establish real-time continuous interaction during ECG signal monitoring (i.e., real-time push notification via Email)

1.4 SCOPE OF THE STUDY

This study covers a unique architecture and development of a real time remote health monitoring system that is based on the concept of IoT and fog computing. The three sensors to be used here are the basic parameters needed to observe a patient's health status. The proposed and implemented fog computer combining Raspberry Pi and Arduino UNO is a unique interface for better monitoring the vital parameters from patient body. The study also highlights the development of a dedicated local server needed to store, maintain the sensor extracted data through fog computer interface. Moreover, a unique concept of push notification sent in real time for fog computing is demonstrated here which justifies the feasibility of our developed system. The scope is limited to the redundant generated data of sensor devices that are to be used only for demonstration and experimentation purposes. The study does not cover the efficacy of the implemented system for any kind of actual diagnosis or any kind of direct medical treatment.

1.5 ORGANIZATION OF THE THESIS

The rest of the dissertation is comprised of four chapters: The Literature Review, Research Methodology, Results and Discussion, and Conclusions. Chapter 2 discusses the literature review and related works based on Fog Computing and IoT in Healthcare applications and systems. Chapter 3 presents detailed information about the architecture, methodology and working procedure of the proposed work. Chapter 4 illustrates the implementation work of the proposed system and discusses the outcome. Chapter 5 provides the demonstration of experimental results and discussions of the implemented system. Chapter 6 and chapter 7 provide the conclusion and future work related to the accomplished system.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter presents some research works related to the present work which have been performed in the past under various circumstances. The limitations of those previous works are mentioned along with them as well. At the end of the review, a brief description on how the study will be conducted to mitigate the limitations of previous works is presented.

Internet-Of-Things (IoT) in Healthcare

Cloud-IoT driven systems usually eliminates the cost of hardware, software, and controls as well as manages the on-site data centers [8]. Cloud-based healthcare system makes it quite easy to get healthcare services over the internet using a web browser on a range of devices [9]. Several smart healthcare systems using health sensors and cloud server has been proposed to provide health monitoring and support service to the person in concern anytime, anywhere [10][11]. Typically, a large volume of sensor data is collected and transmitted to the server in the cloud for further analysis [12]. Internet-of-Things (IoT) usually refers to the real world and little things with limited storage and processing ability, and the important problems about reliability, performance, security, and privacy [9]. The emergence of the IoT as new technology increases the need for cloud computing services and enhances the role of cloud in business and technology [13]. The versatility of IoT makes it more suitable for smart grids, smart homes, smart cities, and wearable health monitoring systems [14]. The evolution of IoT defined as an ecosystem of smart device, driven by sensed, collected, and exchanged information between devices, making it highly prominent in the healthcare sector [15]. The concept of IoT has been widely adopted on Healthcare applications that involve ubiquitous sensors and actuators communicating through Wireless Sensor Network as well as solutions for real-time data analysis [16].

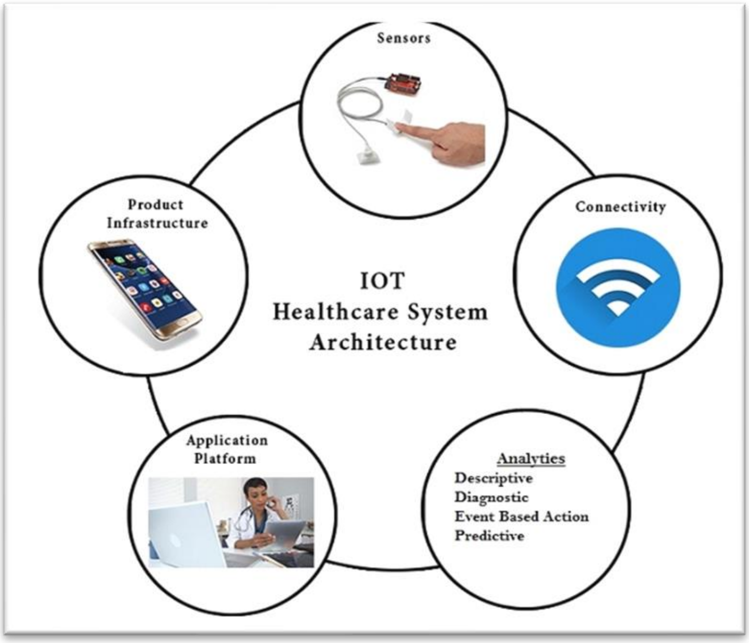


Figure 2.1: IoT in healthcare - A concept

In IoT-based healthcare systems, the possibility of having a real-time lecture on the vital signs of people can prevent a live threat situation [14]. Such health-monitoring systems generally involves wearable devices, wireless body sensor networks, and cloud computing layer and can provide high-quality services in terms of low costs and long-term history of data [7]. In IoT, devices assemble and communicate information directly with each other via the internet and the cloud manages to collect records and analyze data blocks [16].

However, there problems unsolved in IoT based applications includes mobility support, geo-distribution, location-awareness, and low latency [17]. Today's IoT devices generate data at a remarkable speed which requires near real-time processing [9]. There are estimates that healthcare data produced by IoT-based systems will be 25000 petabytes in 2020 [19]. In many cases, each patient requires a high number of sensors, and hence, creating an infrastructure dedicated to an individual becomes inefficient [20]. For the smooth working of high-end IoT applications, significant processing, and interface capabilities are always needed [21]. To address these limitations, Fog Computing comes as a suitable solution to bring computing resources closer to the data sources and for dealing with the challenges in IoT systems while maintaining the high quality of healthcare services. It also allows us to exploit fog computing at smart gateways [22]. Fog computing, introduced by CISCO, is an extension of cloud computing. It is a paradigm to manage a distributed environment that can compute and provide network services between the sensing layer and the cloud layer [23]. Cloud can get assistance by transferring some of the processing and decision making to the fog either close to the client layer or cloud backend [24]. They contribute to reducing the challenges including latency constraints, network bandwidth constraints, resource-constrained Devices, uninterrupted services with intermittent connectivity to the cloud, and new IoT security challenges [25]. The IoT and connected smart devices have exponential growth in terms of involved technologies, paving the way to the evolution of fog computing [26].

Fog Assisted IOT Based Health-Monitoring System

There is a need for enhanced healthcare systems that proffer continuous real-time health monitoring and other advanced services for improving the quality of healthcare services in a manner which will help medical doctors to remotely access the collected data for real-time analysis [7]. Real-time ECG monitoring, neurological diseases, fall detection for stroke mitigation etc require low latency and high response time, therefore fog computing can be the best possible solution for such applications. Also, for research and development, there is a wide variety of hardware development platforms with wireless communication with small form factors and low energy consumption [8]. To sum up, 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].

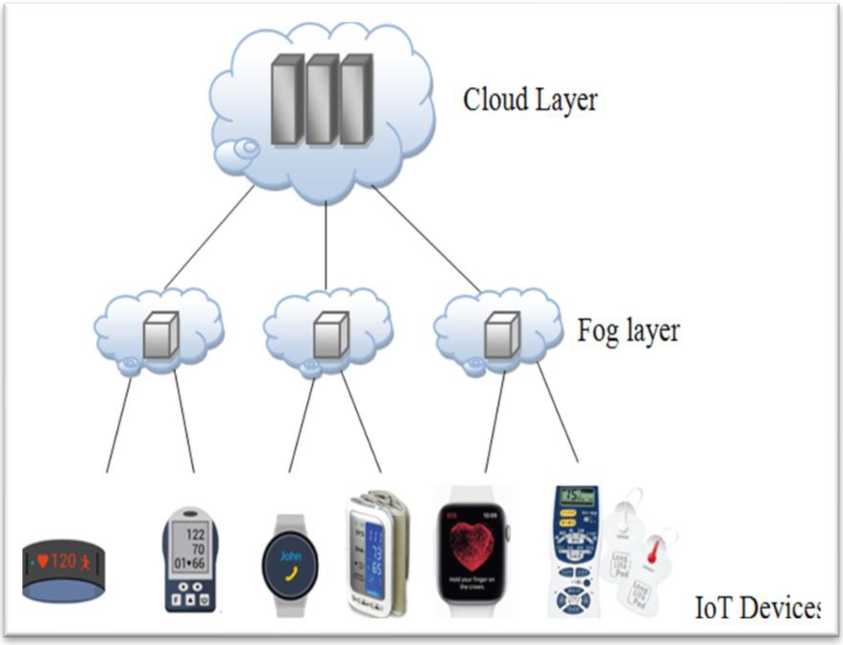


Figure 2.2: Fog computing enabled system architecture

The system architecture includes the three following layers:

1. Health Sensor Layer: Sensors can be different invasive/implantable and non-invasive/ wearable sensors including Peacemaker, wireless capsule endoscope (electronic pill), Retina implants, Electrocardiogram (ECG), Glucose sensor, EMG, EEG, Temperature, Pulse Oximeter, Blood Pressure, Respiration Rate and Blood Oxygen sensor [2]. The patient-related health information is captured by networked sensors, either body-worn or embedded in our daily living activities with which the patient is equipped for personal monitoring of multiple parameters [16]. Sensor may use heterogeneous communication links such as WiFi, WiMax, Zigbee, Bluetooth, IEEE 802.15.4., LAN, etc. to send data to upper layers [17].

2. Fog Computing Layer: The centerpiece 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. [18]. The fog node will condition, analyze, and package this data for the cloud to learn and store [19]. As part of fog computing services, it sends signals to transmit data or notifies to medical caregivers in case of emergencies as well. Fog uses network switches and routers, gateways, and mobile base stations to provide cloud service with minimum possible network latency and response time [20]. This multi-service edge even supports both wired and wireless connectivity [16].

3. Cloud Layer: As in traditional IoT infrastructure architecture, the tasks of the cloud layer include heavy computation, long term larger storage, and wide geographical distribution. Fog nodes interact directly with the cloud layer via networking protocols. Data is stored and processed on cloud servers after collection and aggregation of data from fog devices. The network supported by the Cloud is the single one, which is the TCP/IP protocol.

Finally, biosignals and data generated from sensors are stored in fog local storage/ database. End users like doctors, caregivers can access the fog data by using terminals and the local network like Wi-Fi to connect to local storage. In most cases, this database is not synchronized with the cloud servers' databases except for the back-up cases [7]. Lastly, web browsers, mobile applications are the primary terminals of the health-monitoring IoT systems. End-users can access real-time data in textual or graphical forms through devices including smartphones, laptops, or desktop via these easy-to-use and popular terminals regardless of time and place. Concisely, a local server hosts web pages that can show necessary data in both text and graphical forms in the easy-to-use interface [7].

Fog Computing Services

One of the most important features of fog computing is push notification that allows the doctors or caregivers to be notified about any abnormality of patients in real-time [7]. The unusual data history is sent to the authorized individuals through emails, text messages, or even a loud alarm to alert medical caregivers of abnormal cases. In conventional health-monitoring IoT systems, the push notification is implemented at a cloud for informing unusual cases or in emergencies.

2.2 RELATED WORKS

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 [15]. 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. However, only a concept was presented in the aforementioned study. 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. Two case studies related to human fall detection and heart rate variability were evaluated to illustrate fog gateway's ability to reduce heavy computational loads of sensor nodes. Data processing in the fog included simple filtering and advanced processing algorithms. Results illustrated that fog computing shows its capability to reduce heavy computational loads of sensor nodes by switching the loads to fog-assisted smart gateways. But fog architecture using Pandaboard seems complex compared to other establishments that uses Raspberry Pi. Moreover, the same telehealth services can be provided with a relatively simpler fog scheme. 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 [21]. They benchmarked the performance of developed prototypes on real-world data from smart e-textile gloves. Results from this work demonstrated the potential of the proposed architecture for converting the real-world data into useful analytics while making use of knowledge-based models. In another study, 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 by Gia, T. N. et al. [22]. The system consisted of energy-efficient sensor nodes and a fog layer altogether taking advantage of IoT. The designed sensor node collects and sends wirelessly transmitted ECG, respiration rate, temperature, and humidity data to the Orange Pi (fog gateway). In addition, the system can represent the collected data in useful ways, perform automatic decision making and provide many advanced services such as real-time notifications for immediate attention. End-users (e.g., doctors) can monitor their patients remotely by accessing real-time and history data in Cloud via an Internet browser or a mobile app. However, the performance of Orange Pi as a fog gateway is not properly justified here.

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 Electrocardiogram (ECG)-based Q, R, S detection [23]. Along with fog computers, smartwatch as BSN, gateways e.g., smartphones/tablets, and secure cloud backend were implemented. The case studies on various types of physiological data show that the proposed fog architecture could be used for signal enhancement, processing and analysis of various types of bio-signals. Computations on Intel Edison reduce the data by 99%. However, according to authors, surveying the useful tools for telehealth monitoring was time-consuming. In addition, establishing necessary hardware and programming on the Intel Edison fog was extremely difficult, therefore system complexity and compatibility issues arose. Moreover, since the raw data was not communicated to the cloud, there was no way to perform additional analysis in the cloud. 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 edge of the network [24]. They implemented fog based smart gateway using Pandaboard, Arduino, Zigduino, wireless and Ethernet modules, etc. They chose Electrocardiogram (ECG) feature extraction as a case study analyzed in gateways with heart rate; P wave and T wave feature extraction via a flexible template based on lightweight wavelet transform mechanism. The real-time data is then transmitted to a remote cloud server for real-time processing, visualization, and diagnosis. Their experimental results showed more than 90% bandwidth efficiency and low-latency real-time response at the edge of the network. But considering the fog gateway architecture, the system would seem less complex if Raspberry was deployed as it has built-in wireless protocols and Ethernet support and is comparatively low cost as well.

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 [25]. UT-GATE was constructed from a combination of Pandaboard, SmartRF06 board integrated with a CC2538 module, and several wireless modules. They chose to implement ECG signal processing for movement artifact removal and feature extraction (i.e., heart rate calculation from R-to-R interval, P wave, and T wave). Results show that by outsourcing the processing burden from the sensor node to the smart gateway, about 55.7% of energy is saved. For every 1000 raw samples, it provided 74.1% reduction in sample size. Molina, et al. developed heart rate monitoring system based on IP networking with only the Arduino controller as the main component [14]. So, we can get an insight of the efficiency of Arduino in building such monitoring systems from this study. To demonstrate the effectiveness and efficacy of fog computing paradigm in health monitoring, Cao et al. developed new fall detection algorithms and employed a real-time fall detection system for stroke mitigation employing fog computing paradigm (e.g., distributed analytics and edge intelligence) [15]. This new computing paradigm, if successfully applied for pervasive health monitoring, has great potential to accelerate the discovery of early predictors and novel biomarkers to support smart care decision making in a connected health scenario. It split the detection task between the edge devices (e.g., smartphones attached to the user) and the server (cloud). Experimental results in this work show that distributed analytics and edge intelligence supported by fog paradigm, are very promising solutions for pervasive health monitoring.

Borthakur et al. presented a Fog architecture that relied on unsupervised machine learning big data analysis for discovering patterns in physiological data [11]. 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 (PD). Results show that proposed architecture is promising for low-resource machine learning. It could be useful for other applications within wearable IoT for smart telehealth scenarios by translating machine learning approaches from the cloud backend to edge computing devices such as Fog. 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 [26]. A working prototype of the proposed architecture was built and used to carry out case studies on telehealth big data applications. Case studies in there used the data from the sensors worn by patients with either speech motor disorders or cardiovascular problems. The authors implemented and evaluated both generic and application specific data mining techniques to show orders of magnitude data reduction and hence transmission power savings. 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 by Krishnan, Y. N. et al. [27]. The tagged temperature data packets would be sent to the cloud after time series prediction while untagged data would be sent to the cloud directly. For 1Mbps bandwidth, the result showed a better response time of fog-based architecture compared to the cloud architecture. Monteiro, et al. developed a working prototype of FIT that enabled remote monitoring of clinical speech data to get speech features like loudness, short-time energy, zero-crossing rate, and spectral centroid [16]. They designed fog interface with Intel Edison and other modules using a smartwatch for speech data features extraction, Echowear app for end-user, and secure back-end cloud for long-term analysis. Their results showed the efficacy of FIT to translate the clinical speech processing chain (CLIP) at fog smart gateway.

Some other authors justified the validation of fog computing through simulation using iFogSim simulator for Fog-based IoT-Healthcare solution or fig enabled cloud environment in IoT healthcare [16, 28]. Gill, S. S. et al. proposed a fog-assisted information model which delivers healthcare as a cloud service using IoT devices [29]. The authors aimed to design a fog computing-based resource management model that would gather the information from heart patients using different fog nodes (IoT devices), analyze the patient information to diagnose the status of their health. iFogSim toolkit was used to analyze the performance of proposed model in Fog-enabled cloud environment. Barik, R. K. et al. discussed the concept of edge-assisted cloud computing and its relation to the emerging domain of "Fog-of-things (FoT)" [24]. Such systems employ low-power embedded computers to provide local computation close to clients or cloud. The discussed architectures cover applications in medical, healthcare, wellness and fitness monitoring, geo-information processing, mineral resource management, etc. The fog devices act as intermediate intelligent nodes in such systems where these could decide if further processing is required or not. They summarized case studies employing proposed architectures in various application areas. Others proposed only a conceptual architecture of incorporating fog as the middle layer in IoT-cloud driven healthcare monitoring establishments [15, 16]. Nandyala, C. S., & Kim, H. K. proposed a four-tier architecture for IoT based u-healthcare monitoring with the motivation and advantages of Cloud to Fog (C2F) computing which interacts more by serving closer to the edge (end points) at smart Homes and Hospitals [16].

2.3 SUMMARY

Overall, all the previous works aimed at establishing efficient health monitoring systems with the help of IoT and fog computing concept for smart homes and smart hospitals. Some studies were conducted with the help of different sensors and computing devices while some others used software and simulation techniques to validate their own proposed healthcare system architectures. Again, some works were carried out using predefined medical data and some studies worked on real world data taken synchronously at that time. However, we found a little gap in the implementation of a relatively simpler system that would retrieve real world sensor data at the same time while monitoring the patient. 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 good at heavier processing and analysis. When focusing on simple sensor data like temperature and pulse rate, a lightweight simple 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 is good but 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 of fog computing in the IoT driven healthcare monitoring system. It is also intended to develop a new type of Fog computing interface emphasizing low cost, low power consumption, simpler arrangement, user-friendliness, easier maintenance, and high efficiency. The real-world implementation and performance evaluation through data comparison based on our concept justifies this development for a custom IoT based healthcare solution.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

The fog layer works in collaboration with cloud support and IoT framework to provide real-time health monitoring services effectively. Besides, it often becomes crucial for the system to provide advanced services such as push notification for reporting in case of emergencies to correspondent personnel in real-time. To demonstrate all these requirements along with the benefits of fog computing in IoT systems, a complete system including wearable sensor nodes, gateways with fog computers, and end-user terminals are implemented in our work. To validate the proposed architecture, we have designed and implemented a system prototype based on Raspberry Pi and Arduino embedded processor connected to the wearable telehealth systems. Using the working prototype, we have carried out experiments using real-time data sets collected from patients through our chosen sensors. Raspberry Pi and Arduino UNO together are used here as Fog computing devices for the analysis discussed in this work. However, we have skipped implementing the cloud backend to make the system simpler and lightweight in functionality. The fog local storage which will be implemented here will serve the purpose of data storage and maintenance. We measure the performance of the system by comparing the sensor generated data with medical data along with the biosignals. The performance of the fog computing interface in terms of cost, power consumption, latency is also evaluated to prove the feasibility of the total system establishment.

3.2 FOG ASSISTED IOT BASED SYSTEM MODEL

The success of Fog computing is based on the implemented Fog devices that must be capable to perform successful processing at a very early stage. [15] Features are implemented including acting as a repository (i.e., local database) to temporarily store sensors' and users' data. As the fog node is connected directly with the wearable biosensors, it will ensure local connectivity once placed perfectly. Moreover, it will provide lightweight, limited yet the most anticipated varied computation depending on the requirements needed in the health monitoring system. Furthermore, it will have an onsite local database referred to as a fog server containing features and clinical parameters that can be queried internally and externally.

The prototype of the proposed system is demonstrated below:

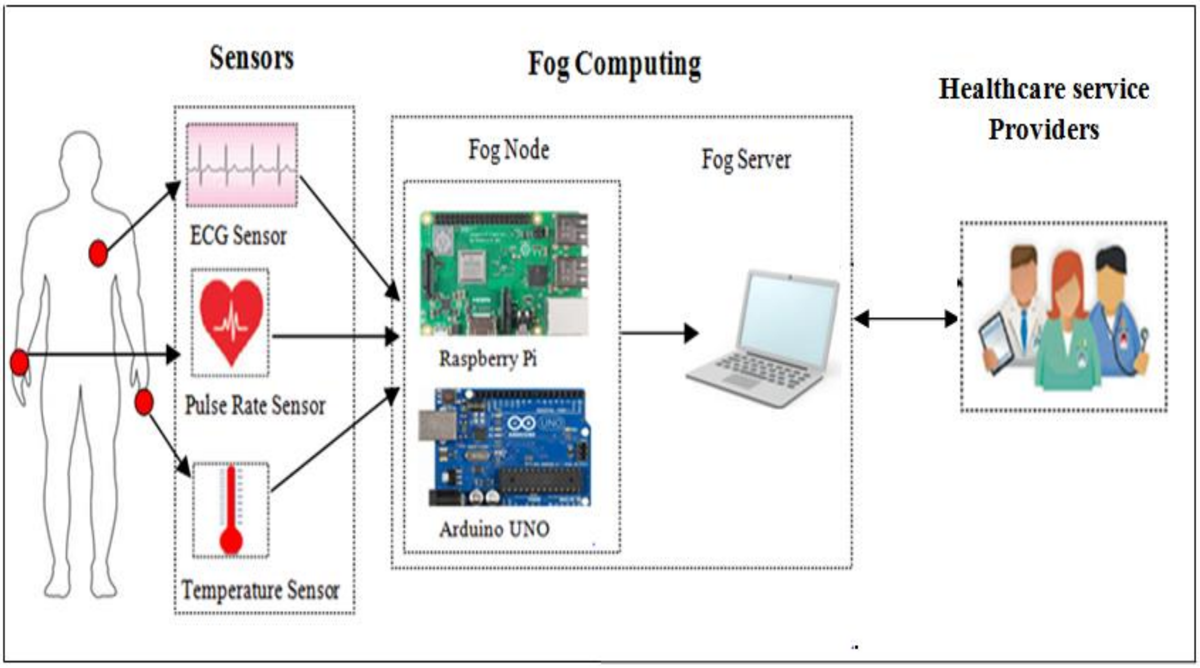


Figure 3.1: Demonstration of fog enabled IoT based health-monitoring system

As shown in the figure above, our proposed system has three layers that is required for a conventional fog-IoT based health monitoring system. The sensors chosen for this work will be implanted on the patient's body for taking the vital parameters. These are temperature, pulse rate, and ECG sensors. After plugging in the system, sensors will start extracting data from the body and send those sensor-generated data to the fog computer of the second layer. The second layer or the fog computer built with Raspberry Pi and Arduino will be responsible for taking those data and perform necessary processing. Arduino has been incorporated with the raspberry pi exclusively only for measuring ECG signals. Arduino is generally the best-suited system for extracting and plotting ECG signals with maximum accuracy. As the system runs, it continuously takes in sensor-generated data and stores in the dedicated fog local server. The fog server implemented here is well synchronized with the fog computer. Finally, healthcare service providers can access the storage for observing the patient's healthcare data as well as taking any decisions in case of emergencies if required. Besides, we have exclusively introduced fog computing service which is referred to as push notification. When ECG signals are generated at the Arduino board and displayed at the proper platform, the signal plot is also sent to the designated medical caregiver as a video file which 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 along with the most crucial benefit of a fog computing concept.

The workflow diagram of the proposed system according to the method presented above is shown as follows:

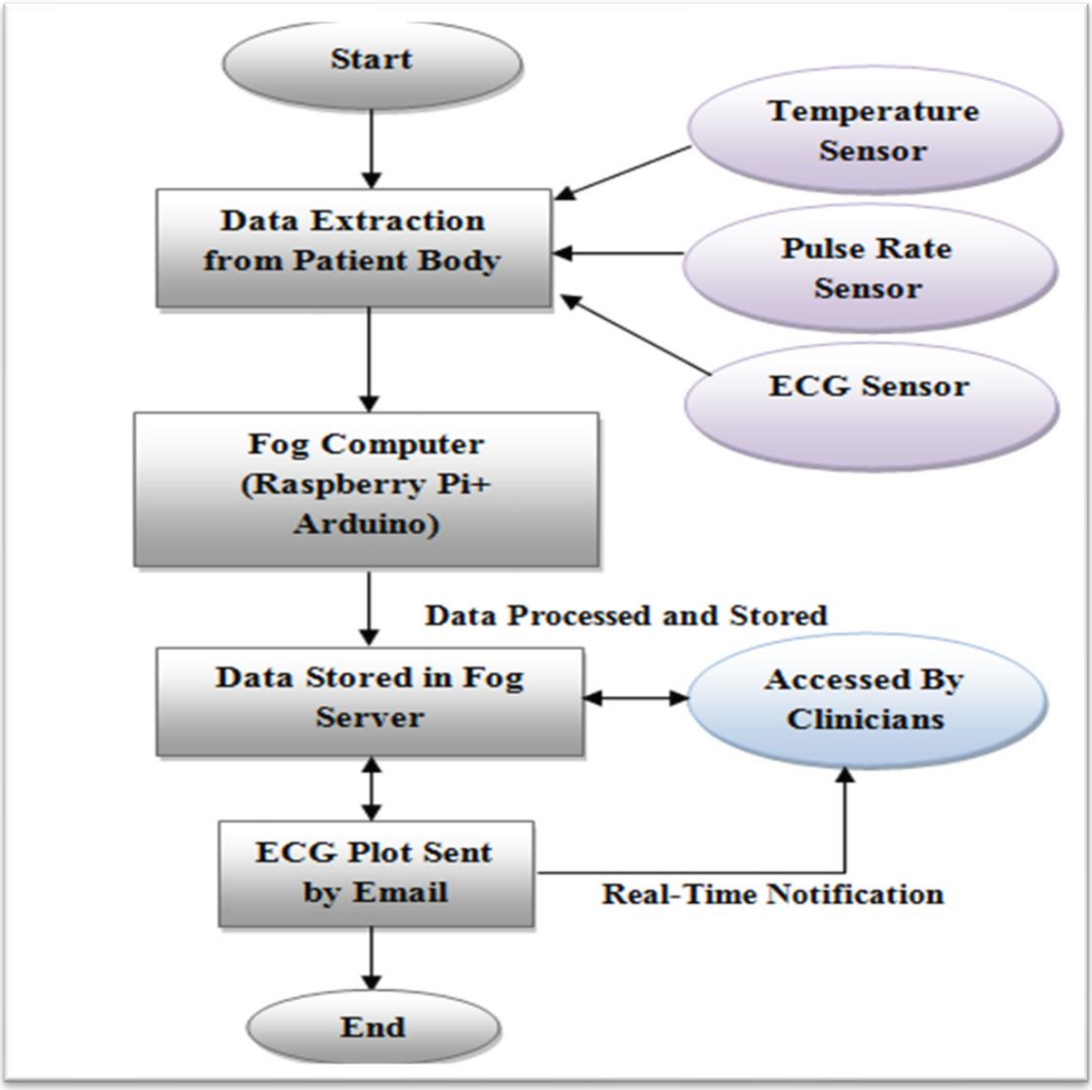


Figure 3.2: Workflow diagram of the proposed architecture

3.3 SYSTEM DESIGN

The design of the system and the necessary tools required for designing the system at each layer is described below:

3.3.1 Sensor Layer

To measure the patient's vital parameters, three sensors are chosen in this work to design the sensor layer of the proposed system. Their functions in extracting health data as well as in this work are described briefly below:

Temperature Sensor: The temperature sensors are precision integrated-circuit sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. It is a temperature measuring device having an analog output voltage proportional to the temperature. It is a 3-terminal sensor used to measure the surrounding temperature ranging from -55°C to 150°C. It is best suited for remote applications and is advantageous due to low-cost due to wafer-level trimming, low self-heating, etc. For body temperature measurement, with a human touch, it gives a variety of temperatures.

Pulse Sensor: The heartbeat is measured in beats per minute or BPM, which indicates the number of times the heart is contracting or expanding in a minute. The normal heart rate is between 60 and 100 bpm [14]. Pulse/ Heartbeat Sensor, also called Biometric Pulse Rate or Heart Rate detecting sensor is an electronic device that is used to measure the heart rate i.e., speed of the heartbeat. This Plug and Play type sensor has inbuilt amplification and noise cancellation circuit in it. The LED on the front side of the sensor is placed over a vein in the human body. This can either be fingertip or ear tips, but it should be placed directly on top of a vein.

ECG Sensor: ECG/ the AD8232 sensor is a neat little chip used to measure the electrical activity of the heart. This electrical activity can be charted as an ECG or Electrocardiogram and output as an analog reading. The AD8232 is a cost-effective integrated signal conditioning block for ECG and other biopotential measurement applications. It is designed to extract, amplify, and filter small biopotential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement. ECGs can be extremely noisy, the AD8232 Single Lead Heart Rate Monitor acts as an op-amp to help obtain a clear signal from the PR and QT Intervals easily. Provided on this board are RA (Right Arm), LA (Left Arm), and RL (Right Leg) pins to attach and use their custom sensors. Additionally, there is an LED indicator light that will pulsate to the rhythm of a heartbeat. It is recommended to snap the sensor pads on the leads before application to the body. The closer to the heart the pads are, the better the measurement. An ECG Sensor with disposable electrodes attaches directly to the chest to detect every heartbeat. The electrodes of the ECG sensor will convert the heartbeat into an electric signal. ECG Sensors are very lightweight, slim, and accurately to measures continuous heartbeat and give rate data of heartbeat.

These three bio-sensors will take patient body parameters in textual and graphical forms and send them to the fog computer. Temperature and pulse rate data will be sent in textual form to the Raspberry Pi, and the ECG data will be sent as a signal to the Arduino.

3.3.2 Fog Node

Raspberry Pi and Arduino together are being used here as a fog node and the collected data (temperature, pulse rate, and ECG signal) are sent from this fog node to the fog server for processing and storage. Raspberry Pi is responsible for extracting data from temperature and pulse sensors in a continuous manner. Arduino is used for extracting and displaying ECG signal data. Some specifications of the devices are also depicted as follows:

Raspberry Pi

With the advent of the Raspberry Pi, a credit-card-sized single-board computer, there is a great opportunity to construct a low-cost, low-power portable cloud to support real-time data processing next to IoT deployments. [31] It has gained much popularity owing to its small size and multipurpose utility. Some of its 26-pins GPIO (General Purpose Input and Output) port can be used as digital input/output signals and can be programmed directly on RPi through high-level programming languages (e.g., C++, Python, and Java.), while others can be used as interfaces for embedded protocols for controlling a set of an electronic circuit (sensors, analog to digital converters (ADC), relay, status button, etc.). Hence, the GPIO port represents the main way of connecting RPi with other electronic boards as well as the communication link with other computing devices using a variety of different protocols, including Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I2C). It is important to note that, based on its performances; RPi can also serve as a gateway [15].

Raspberry Pi Architecture

The Raspberry Pi (RPi) as a fog computer includes several embedded devices along with an embedded operating system to function properly. It has integrated USB Ports, display port, HDMI port, Ethernet, wireless LAN, GPIO pins, and SD card slot where the specific operating system is stored to make the system run. Moreover, RPi has a built-in WiFi module so we do not need additional wireless communication protocol. The specific operating system for running the RPi ensure that real-time data is transmitted into a remote server and all fog computing service function properly. The embedded operating system is lightweight, low-energy consuming, does not require powerful hardware, and serves quite perfectly in the Raspberry Pi system. From the fog service perspective, we can state that this computing node provides necessary services required for real-time monitoring including fog storage for temporary keeping sensor extracted data, push notification like sending emails to the anticipated medical caregivers and graphical interface with access management so that doctors, patients, and medical personnel can visualize data like ECG signal and other bio-data. The notification service is for real-time notifications about informing abnormal scenarios and cases to the medical personnel. The Arduino system performs this task in the fog system.

The bio-sensors are assembled to the RPi via an analog-to-digital converter so that digital textual data are retrieved at the fog end. As the system is securely plugged in, bio-sensors will take body parameters and start sending sensed data to the RPi through an analog-to-digital converter. Inside the RPi, these sensor-generated data will be processed and stored in the fog storage server. In this way, we can get patient body temperature and heart rate data in real-time.

Arduino UNO

Arduino Uno is a microcontroller board based on an 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists of other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header, and a reset button. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. It requires additional hardware modules for wireless communication [4]. It is exclusively used in healthcare robotics and serves excellent for extracting ECG signal data from the patient body.

Arduino UNO Architecture

The Arduino UNO microcontroller is said to be the appropriate system for ECG signal monitoring. The dedicated Arduino software includes a serial monitor for graphical representation of ECG signals continuously. After connecting the ECG sensor with the Arduino board and placing the electrodes on the patient's body, the ECG data will start sending signals from the sensed data to the circuitry. We can visualize the data via the serial plotter included in the Arduino software service. The Arduino sends ECG signal graph plots to the caregivers through emails in a video format. This is an exclusive method adopted in this work by us as a way to monitor the patient heart condition 24/7.

3.3.3 Fog Server

Intelligent processing can be done at the fog server by algorithms or fuzzy logic techniques [15]. The sensors extracted data will be sent to the Raspberry Pi and Arduino from where it will be sent to the fog server for the local storage of data. The fog local database in the fog server is configured to have the same structure as the global database in the cloud for achieving an easy synchronization between these databases. A remote server using JavaScript is created which will provide a secure environment for safe-keeping of patient records. In our application, we have implemented only the fog local storage for keeping the sensor data in the fog computer. In the case of ECG data, two types of data will be stored in the fog server: one is real-time data and the other is data stored in MySQL. All the data will be sent to the server as a String.

Overall, the usage of the Internet for running the system will follow the usual TCP/IP protocol. The fog computer Raspberry Pi used by us has a built-in WiFi facility so we do not have to assemble any separate Wifi module to the establishment.

3.3.4 Fog Computing Service

Among several fog computing services offered by fog computing scheme, we will try to proffer push notification service for our proposed system.

Push Notification

We will introduce an efficient mail notification system for ECG graphical representation that will be sent via SMTP protocol. This is a unique type of fog computing service adopted by us. The ECG plot will be sent to the designated email address of the medical personnel as a .mp4 video file format which will provide an accurate representation of the ECG signal taken and measured right away 15 seconds apart. This will justify the real-time notification service of our implemented fog computing scheme.

Using all the required tools for establishing the system and necessary methods, we will establish a low-cost, energy-efficient, real-time functioning healthcare monitoring system which will be explained in the following chapter. Emphasizing on simpler systems, low-cost, and energy efficiency criteria, we will gather necessary tools, assemble them all, and build the total monitoring system which greatly focuses on the benefits of applying fog computing in IoT based establishment scenarios. As our proposed work focuses on the fog computing concept yet building a simpler system, we will skip the cloud server scheme and at the same time build a local fog server that will serve just right for the storage and maintenance of patient medical data. As per our proposed method, we will also introduce an exclusive method of providing real-time notification service from our fog computer while monitoring ECG signals of the patient.

CHAPTER 4

IMPLEMENTATION

The implementation of the patient health monitoring system is illustrated following the methodology discussed above. The tools for connecting and assembling, the necessary operating systems and software required to run the system are also explained in detail below.

4.1 Sensor Layer

The three types of sensors that we have used in the implementation are as follows:

- LM 35 Temperature Sensor module
- Pulse Sensor
- AD8232 Sensor (For ECG Monitoring)
- ECG Electrodes- 3 Pieces
- Connecting Wires

LM35 sensor collects body temperature every 0.1-second interval. Finger needs to be put on the sensor and after few seconds apart, LM 35 sensor starts taking temperature data in textual form.

Pulse sensor that is used for extracting heartbeat in beats per minute is relatively easier to take heart rate data compared to the other two sensors. After holding the sensor with fingers, it begins to extract pulse rates from the patient's body and show it in textual form as well.
AD8232 sensor is exclusively used for retrieving the ECG signal continuously along with three electrodes worn on the patient body. After proper positioning of the electrodes with the sensor circuit, the three pads are placed appropriately on the body. There are two ways of placing the pads on the body. One way is placing one in the right arm, one in the left arm, and the last one in the right leg. Another method is placing one in the right of the heart, another one in the left of the heart, and the last one in the kidney section or around the liver. We are using the second method for extracting signals. As the Arduino serves excellent for retrieving and displaying the

ECG signal, we have assembled and attached the circuitry with the Arduino UNO with proper wires.

As the sensors need to be connected to the fog computer through wires, we have used some pairs of connecting wires to connect the sensors to the fog computing system.

4.2 Fog Node

For implementing the fog computing node, the necessary hardware, tools, and software to be used are listed below:

Raspberry Pi

The necessary hardware, tools as software for setting up the Raspberry Pi device for establishing fog node is depicted below:

Hardware and Tools

- Raspberry Pi Model 3B+
- ADS 1115 16bit I2C ADC
- SD Card (16 GB)
- 5V 4.8 A charger
- Connecting Wires

For implementing the fog computer, we have used the Raspberry Pi model 3B+ from the Raspberry family. The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting 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, and PoE capability via a separate PoE HAT. The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

The bio-sensors (Temperature and Pulse Rate sensors) are accurately assembled with the Raspberry through connecting wires. However, since the GPIO of Raspberry Pi only allows digital signal inputs, an analog to digital (A/D) converter was used to convert the analog vibration signal into a digital signal for transmission to Raspberry Pi. For microcontrollers without an analog-to-digital converter or when we want a higher-precision ADC, ADS 1115 16bit I2C ADC+PGA analog to digital converter is extensively used. As the temperature data and pulse data are initially analog and RPi has no built-in analog-to-digital converter, this converter is integrated into the fog computer circuitry. The SD card is inserted through the proper slot inside the RPi with an operating system loaded to run the RPi. We have used 16 GB SD cards which is enough to store the OS and necessary programs.

Software and Operating System Installation

- Debian (OS)
- Raspberry Pi terminal (UI)
- Python 3.2 (programming IDE)
- Matlab

Debian or, Debian GNU/Linux, is a Linux distribution composed of free and open-source software, an operating system based on the Linux kernel. We are using Debian for the operating system requirement of the Raspberry Pi. NOOBS, an operating system installation manager is found on the raspberrypi.org website and needs to be stored on the SD card that is inserted into RPi. Among several operating systems included in NOOBS, we are using Raspbian. Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware; an operating system that makes the Raspberry Pi run.

The Raspberry Pi terminal (also known as the shell or command-line interface) is a text-based interface that accepts and interprets our given commands. We use terminal commands in Raspbian to run our anticipated programs. The default terminal on the Raspberry Pi devices running Raspbian is called LXTerminal, a text-based interface in which we type in commands and get a response.

Python 3.2 programming IDE is used for writing the programs. Along with the other OS and terminals, it is inserted into the SD card as well. Matlab is used in the fog computer for necessary programming and running the system.

Arduino UNO

Hardware and Tools

- Arduino UNO REV 3
- USB 2.0 Cable Type A/B

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino board is connected to the Raspberry via the USB cable. These two devices together act as a fog computing system in our work. The necessary software needed to run the Arduino is integrated into the Raspberry Pi so that the central control lies on the Raspberry Pi. The ECG sensor is integrated properly through connecting wires to the Arduino board analog and digital pins and power pins.

Software

- Arduino IDE
- Processing IDE

The open-source *Arduino* Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The Arduino IDE is installed inside the Raspberry Pi terminal. So, the programs can be written and implemented within the Raspberry.

As the system is powered on, along with the RPi operation, the Arduino starts fetching ECG signals from ECG sensors through electrodes placed on the patient body. We have activated the real-time notification service (fog service) for ECG signals sent through email as an .mp4 video file format.

The final assembled monitoring system is illustrated below:

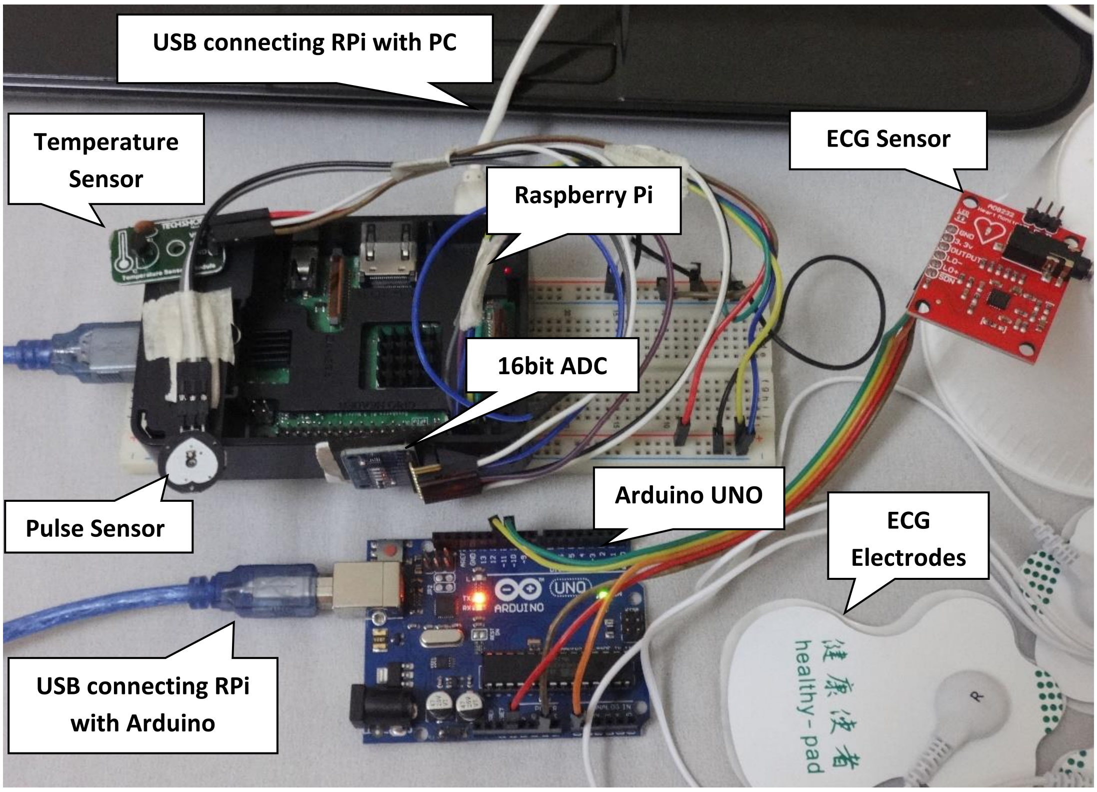


Figure 4.1: Demonstration of fog-IoT based real-time health monitoring system

The above figure demonstrates the developed health monitoring system with the components specified and labeled clearly. As per the methodology and requirements discussed earlier, we have assembled all the hardware, installed the software and run the system to monitor health status of patients in real time.

4.3 Fog Server

Software and Tools

The software and tools used for implementing the fog server in our system are described as follows:

- Scripting Language: HTML5, CSS, XML, JavaScript, JSON
- Database Language: PHP, MySQL
- Development Environment: Linux
- Server Configuration: Debian
- Web Server: Apache
- Server Security Management: Remote Server (Secured; Created only for this application)

To support the storage of clinical features and analytics, we implemented a backend cloud database using PHP and MySQL. Firstly, we set up a Linux, Apache, MySQL, PHP (LAMP) server, an open-source web platform for development on Linux systems using Apache 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 normal PC for using the server so that Fog computers can easily engage with the database of the fog server. There are ways of operating and displaying data and everything 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 interface. The webpage is built using Python, HTML5, CSS, XML, JavaScript, and JSON. The fog's webpage has a form having a username and password. By using an Internet browser, and end-user like medical doctors, caregivers having valid credentials can connect to the webpage and access the monitored data in real-time.

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

The practical demonstration of the developed monitoring system for collecting ECG signal and other readings with a demo patient is shown below:

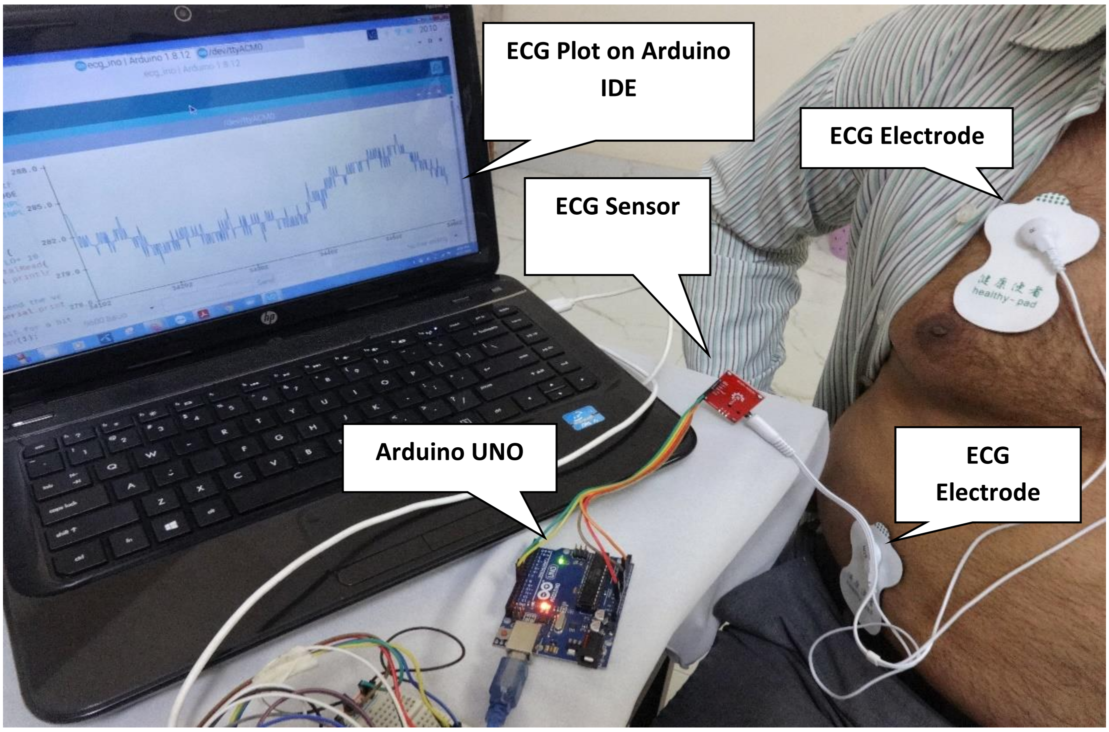


Figure 4.2: Real time Health monitoring and ECG Plotting with the developed system

As per shown in figure 4.2, the arduino code runs in Arduino IDE and starts taking ECG data continuously through ECG sensor and attached three electrodes in the patient body. The signal is shown as a graph in Arduino serial plotter as shown above. We can also get textual data from Ardino serial monitor.

The implementation and performance evaluation of the implemented system as a real time fog-IoT based health monitoring system is explained in the next chapter.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 INTRODUCTION

This chapter will show the demonstrations of the overall system after implementing and running the system regarding simpler, low-cost, energy-efficient real-time health monitoring aspect. This chapter will provide how the system establishment has enabled us to observe sensors extracted data through the implementation and operating of the system. The subsection 4.2 will show the temperature data generated by temperature sensor of the system including system temperature, body temperature, and a visualization of multidimensional temperature data. The subsection 4.3 will illustrate the pulse rate data generated by heartbeat sensor. Apart from showing the raw data in subsection 4.3, the subsection 4.4 will also demonstrate temperature and heart rate data as a line chart. The subsection 4.5 will provide the ECG signal demonstrations at Arduino UNO microcontroller. The subsection 4.6 will show us that ECG Signal Graph is sent as email notification. The subsection 4.7 will give us an insight of the implemented fog server and data demonstration in that fog server. Lastly, the subsection 4.8 will show the performance evaluation of the implemented system to check the feasibility of it as a real time fog-IoT based health monitoring system.

5.2 DATA GENREATED FROM TEMPERATURE SENSOR

After plugging in the system and running the temperature program, we get the output in degrees Celsius. The temperature data of the body is generated after the patient holds the temperature sensor for a while for temperature measurement. The data demonstration is as follows:

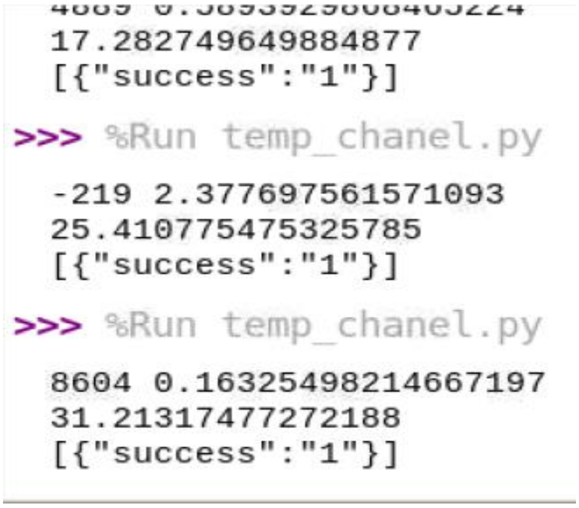


Figure 5.1: Generation of temperature data through temperature sensor

Figure 5.1 shows the temperatures generated by the temperature sensor as soon as the sensor starts working. The first two data show 17 degrees and 25 degrees Celsius which are environment temperatures and the third one shows body temperature 31 degrees Celsius when the sensor is touched with finger. And the data gets generated continuously as long as the patient tends to get the body temperature measurement.

5.3 DATA GENENRATED FROM PULSE SENSOR

The heart rate or pulse rate data is extracted from the pulse sensor by putting fingers and tapping multiple times. After a while, it starts showing the results continuously in beats per minute (bpm).

The pulse rate data plot is demonstrated below:

```
Shell
no beats found
no beats found
no beats found
BPM: 40.81632653061224
BPM: 38.897893030794165
BPM: 35.84443515144274
no beats found
BPM: 39.473684210526315
BPM: 37.686074995289246
no beats found
BPM: 41.899441340782126
BPM: 42.06393718452047
BPM: 42.17333239614817
```

Figure 5.2: Generation of Pulse rate data through pulse sensor (in BPM)

The generated pulse rate data are shown in figure 5.2 in BPM (Beats per Minute). After putting a patient finger on the LED of the pulse sensor, the pulse rates are extracted and shown in the monitor through running the program.

5.4 LINE CHART DEMONSTRATED FROM TEMPERATURE AND HEART RATE DATA

We can get the latest record of body temperature and pulse rate data as a line chart by accessing the following site: *api.appresource.com/ui/* in the browser. This is the demonstration of the user interface for viewing the temp and pulse data. This will help us to track the rate of the body generated data easily and act accordingly in case of emergencies if needed.

The line charts of body temperature and heart rate data generated by sensors and extracted by Raspberry Pi is shown as follows:

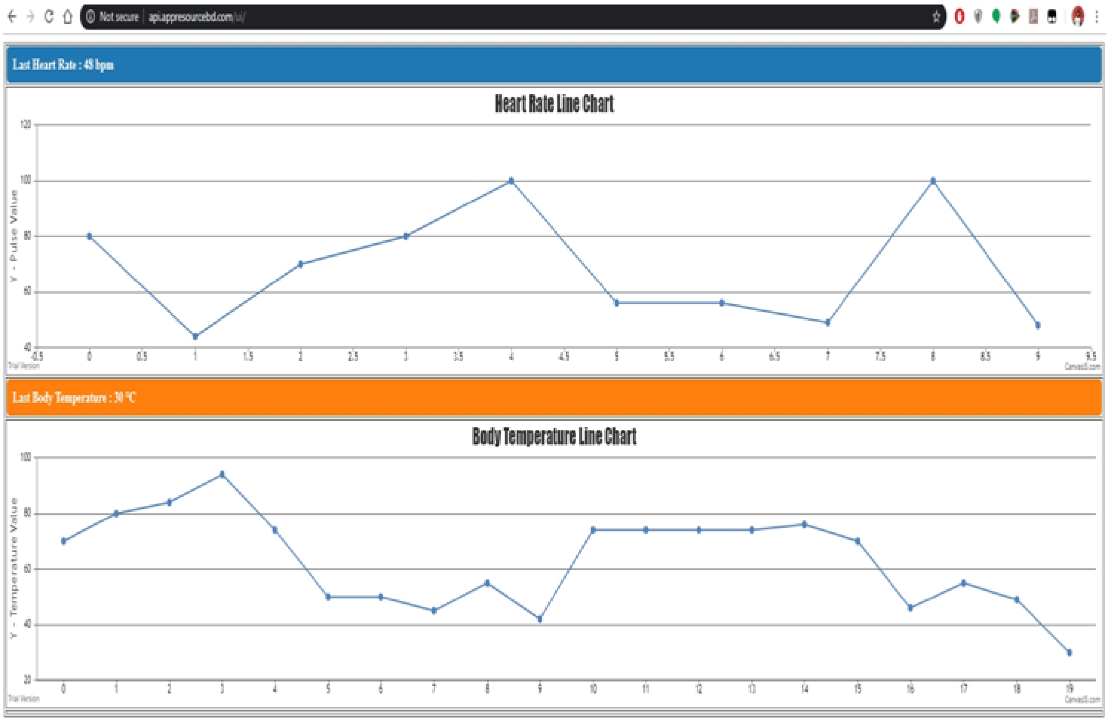


Figure 5.3: Illustartion of temperature and pulse rate data as line chart

Figure 5.3 shows temperature and pulse rate data as line graph illustration as well as the last recorded sensor data. This chart helps us to track some of the last generated data by temperature and pulse rate sensors and thus helps us to understand the situation of the patient better.

5.5 SIGNAL GENREATED FROM ECG SENSOR AT ARDUINO UNO

After placing the three electrodes into the body following the specific method and running the code in Arduino, we run the serial plotter of Arduino IDE to observe the ECG signal. The result is demonstrated below:

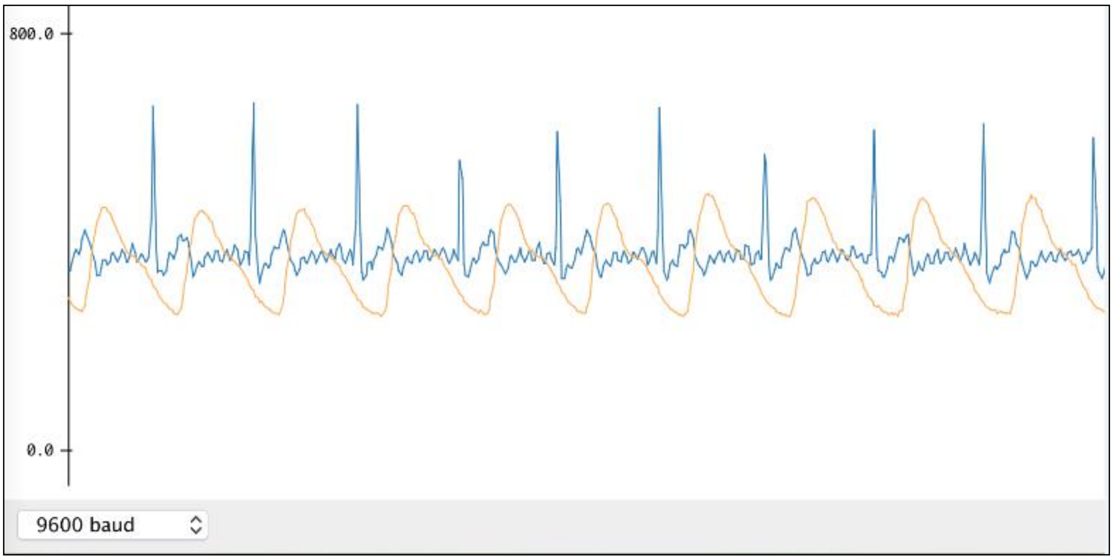


Figure 5.4: Generation of ECG signal plot in Arduino serial plotter

Figure 5.4 shows the ECG signal graph in Arduino Serial Plotter after running the program in Arduino IDE and opening serial plotter to observe the signal. After three pads attached with the AD8232 sensor are placed on the body and start generating an ECG signal continuously, the program in Arduino IDE is run and the data is observed in Arduino serial plotter.

5.6 ECG SIGNAL GRAPH SENT AS EMAIL NOTIFICATION

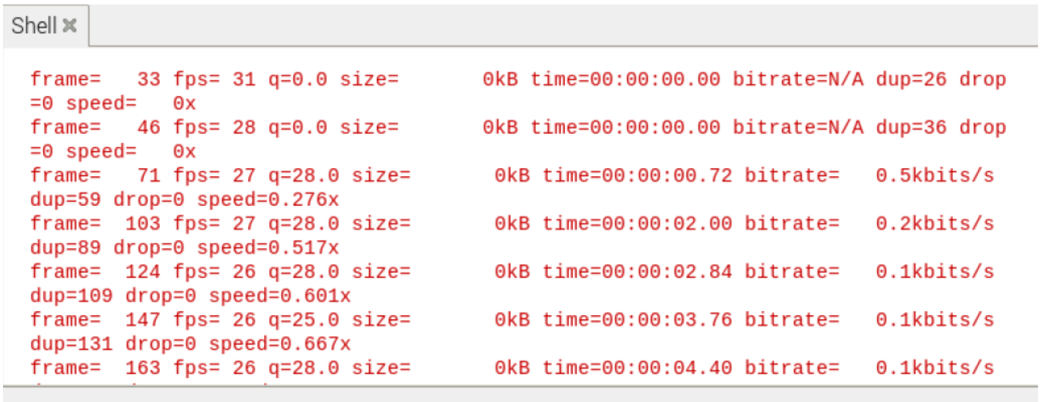
For activating the real-time notification service of fog computer, we created the video alert system which enables the medical personnel to observe the ECG signal remotely yet in real-time. After the ECG signal at the Arduino board is generated and illustrated, the signal plot is converted into a video file of 6-7 seconds in .mp4 format and sent to the designated email address of the medical caregivers following SMTP protocol. This push notification or fog computing service is demonstrated and works correctly after running the system and the programs in the fog computer.

First, the ECG signal data in the video file format is shown below:

Figure 5.5: Illustration of ECG signal plot in .mp4 file format to be sent as Email

Figure 5.5 shows the ECG signal plot in .mp4 video file format for easier demonstration of real-time fog computing service. The generated video file is illustrated along with the file structure to show us the details of the generated file. This helps us to understand the video file better from a technical perspective

The elaborated structure of the video file generated for ECG signal plot is also illustrated as follows:



```
Shell ✖

frame=   33 fps= 31 q=0.0 size=       0kB time=00:00:00.00 bitrate=N/A dup=26 drop
=0 speed=   0x
frame=   46 fps= 28 q=0.0 size=       0kB time=00:00:00.00 bitrate=N/A dup=36 drop
=0 speed=   0x
frame=   71 fps= 27 q=28.0 size=       0kB time=00:00:00.72 bitrate=   0.5kbits/s
dup=59 drop=0 speed=0.276x
frame=  103 fps= 27 q=28.0 size=       0kB time=00:00:02.00 bitrate=   0.2kbits/s
dup=89 drop=0 speed=0.517x
frame=  124 fps= 26 q=28.0 size=       0kB time=00:00:02.84 bitrate=   0.1kbits/s
dup=109 drop=0 speed=0.601x
frame=  147 fps= 26 q=25.0 size=       0kB time=00:00:03.76 bitrate=   0.1kbits/s
dup=131 drop=0 speed=0.667x
frame=  163 fps= 26 q=28.0 size=       0kB time=00:00:04.40 bitrate=   0.1kbits/s
```

Figure 5.6: Illustration of detailed video file structure of generated ECG signal plot

The figure 5.6 shows the detailed information of ECG signal plot as video file structure in terms of frame number in seconds (fps), the data upload speed, time and bit rate in seconds (kbps). This helps us to understand the video file clip of ECG signal better from a technical perspective.

The Email Alert notification sent to the anticipated address is demonstrated as follows:

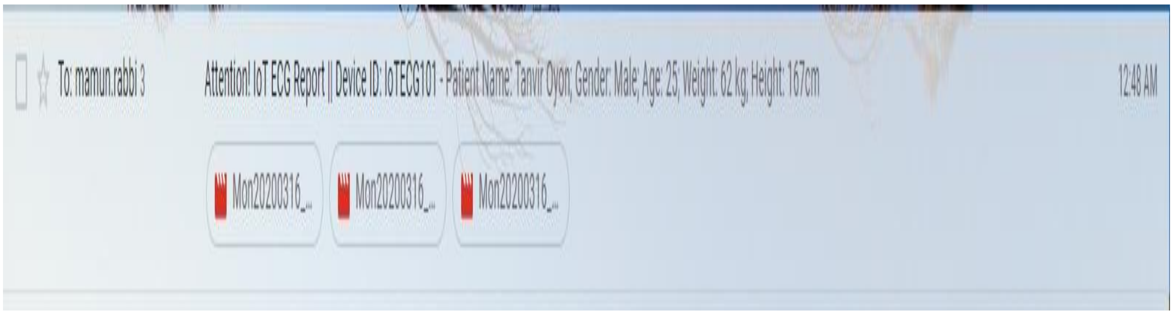


Figure 5.7: Illustration of Email Notification of ECG Report

The Email Notification for ECG Report is illustrated here which proves the accurately delivered status of the video file. The email is sent from the server to the designated medical personnel recipient.

The following figure also shows the delivered status of the email on the Android system:

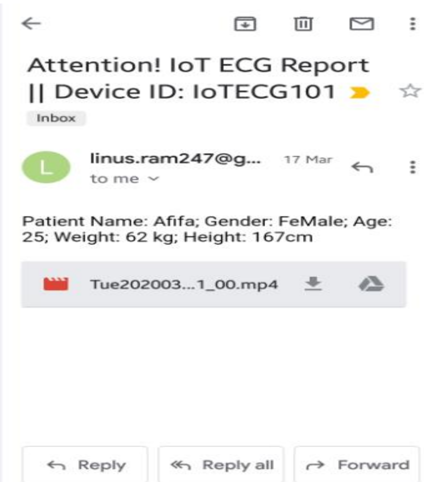


Figure 5.8: Illustration of Email notification for ECG report on android system

The above figure 5.8 shows us that the ECG plot as a video file can also be delivered to an android system to the designated email address. Simply put, people with stable internet connectivity can retrieve the .mp4 file anytime anywhere with either a PC or a mobile device.

5.7 DATA STORED IN FOG SERVER

As per our implemented system, the data generated by temperature and heart rate sensors are stored in our implemented fog server simultaneously. In the server, data are stored one by one whenever sensor data are created. To recognize the temperature and pulse rate data with necessary details, we have created self-generated tables with an id, value, or the data generated, data type to differentiate, and the detailed date and time for better understanding. This is flexible storage that can store numerous data exactly at the time they are created at the sensor nodes synchronously. The concerned medical caregivers will access the database with a username and a 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 as follows. We have shown sensor-generated data stored at a different time at the runtime of the system to show the efficacy and flexibility of our implemented fog server.

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
29	80.0000	b	2020-03-07 04:46:56

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
1980	78.0000	h	2020-06-18 11:22:55
1981	78.0000	h	2020-06-18 11:23:46
1982	19.8106	b	2020-06-18 11:24:53
1983	14.5371	b	2020-06-18 11:25:21

Figure 5.9: Illustration of the stored temperature and heart rate data in the implemented fog server

Figure 5.9 shows the implemented fog server for storing sensor generated data of the system. In the type name, ‘b’ denotes body temperature and ‘h’ denotes heart rate or pulse rate. The data measured in the Raspberry Pi and Arduino UNO and stored in our server at different periods are justified in the date column. This simple yet flexible and real-time interactive fog server implemented here proves the maximum efficacy of our patient health monitoring system regarding real-time and efficiency aspects.

We have successfully developed and implemented a simple, low-cost, energy-efficient, and flexible health-monitoring system where the fundamental components of IoT and fog computing are established and worked correctly according to the proposed method. We justified our model by incorporating the IoT devices as medical sensors, implemented a fog computer using Raspberry Pi and Arduino UNO, and developed a server system synchronized with our developed fog computers and connected sensors. The total system is responsible for extracting the bodily parameters (body temperature, pulse rate, ECG signals) through bio-sensors in close contact with the human body and is compatible with the established fog computing node and updating the processed data from the fog computer to the fog server continuously in real-time and synchronously. Moreover, this system accurately sends the ECG signal data to the Email recipient in 15 seconds apart which successfully validates the push-notification service of the fog computing concept. Moreover, the system is budget-friendly compared with other established and discussed fog computing systems dedicated to healthcare or health monitoring because of comparatively cheaper sensor nodes, easy to understand and maintain fog devices, and a dedicated secure fog server for health data storage. For presenting the real-life ECG signal illustration before the medical personnel, sending the signal plot as a video file via email is the best possible demonstrated concept adopted and implemented by us. Overall, the implemented patient health monitoring system serves excellent in all aspects of giving remote service to patients, interacting them in real-time and taking decisions upon any unusual cases.

5.8 PERFORMANCE EVALUATION

Fog or the local processing unit requires to continuously handle a large amount of sensory data in a short time and response appropriately with respect to various conditions [36]. As we implemented a real time health monitoring system, the performance of the overall system is emphasized on real time data delivery and notification at the receiver end with minimal latency (delay). The developed fog node should be able to process the sensory data right on time after they are extracted from patient body through sensors. Here, the analysis is focused on comparing the extracted temperature and pulse data with medical data as well as the performance of the fog node or the overall monitoring system in terms of latency. The performance of the inter-communication among the layers is measured here as well.

5.8.1 Performance Testing of Sensor Data

First, we check on the data retrieved from temperature and pulse sensors. 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 through medical devices 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 table I below:

TABLE I

TEMPERATURE AND PULSE DATA COMPARISON

1	Serial No.	Gender	Age	Actual Temperature©	Sensor Tempearature ©	Actual Pulse(BPM)	Sensor Pulse(BPM)
2	Patient 1	Female	17	32	30	110	129
3	Patient 2	Female	19	31	27	80	76
4	Patient 3	Female	20	30	25	78	85
5	Patient 4	Male	25	36	29	82	83
6	Patient 5	Male	22	34	29	96	92
7	Patient 6	Male	32	40	38	88	94
8	Patient 7	Male	23	35	28	92	94
9	Patient 8	Male	22	37	33	102	97
10	Patient 9	Male	18	36	34	85	82
11	Patient 10	Male	19	39	38	95	70
12	Patient 11	Female	27	33	31	76	82
13	Patient 12	Female	23	29	32	89	105
14	Patient 13	Female	18	32	34	98	96
15	Patient 14	Male	16	39	36	82	85
16	Patient 15	Male	21	37	34	72	79
17	Patient 16	Female	29	35	37	96	98
18	Patient 17	Female	27	33	34	101	98
19	Patient 18	Female	28	30	31	93	84
20	Patient 19	Male	24	34	31	95	108
21	Patient 20	Male	22	36	35	109	96

According to table I, we take measurements from 20 different male and female patients through medical temperature and pulse measuring devices. We compare the temperature data measured through actual medical device with sensor generated temperature data and find the difference which is illustrated in a line graph as shown in figure 5.11 below:

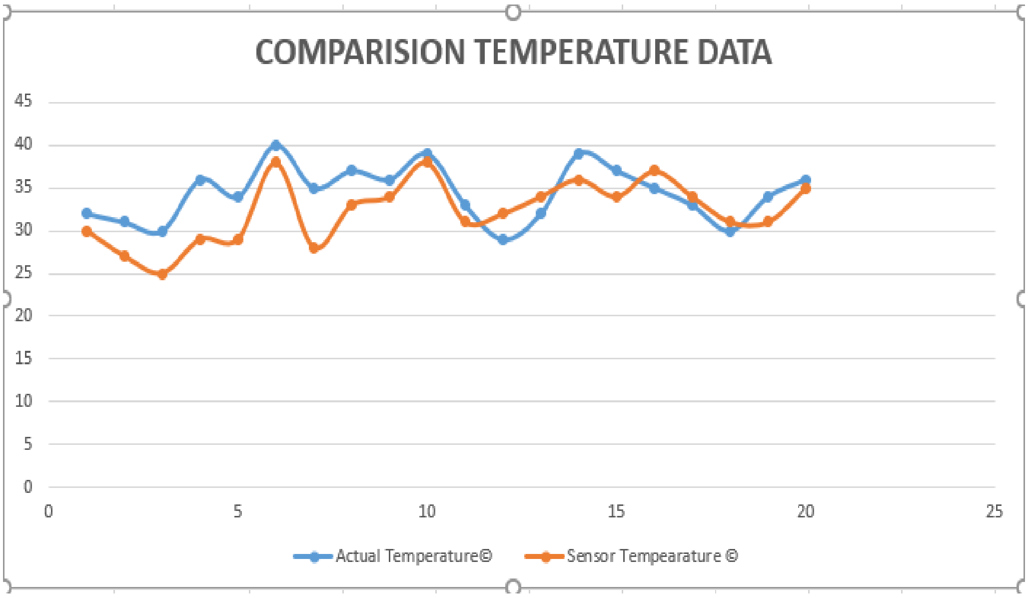


Figure 5.10: Comparison of actual and sensor generated temperature data

According to the figure 5.10 shown above, twenty different data are taken to demonstrate the difference between actual data and sensor data. The actual temperature data is shown as blue line whereas the sensor data is shown in the yellow line. We see the output lines are varying between values 30 and 40 against twenty measurement data with peaks and fluctuations almost similar for both sets of data. The difference illustration validates the sensor data generated by our system as a feasible one.

Then 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. It is illustrated in figure 5.11 below:

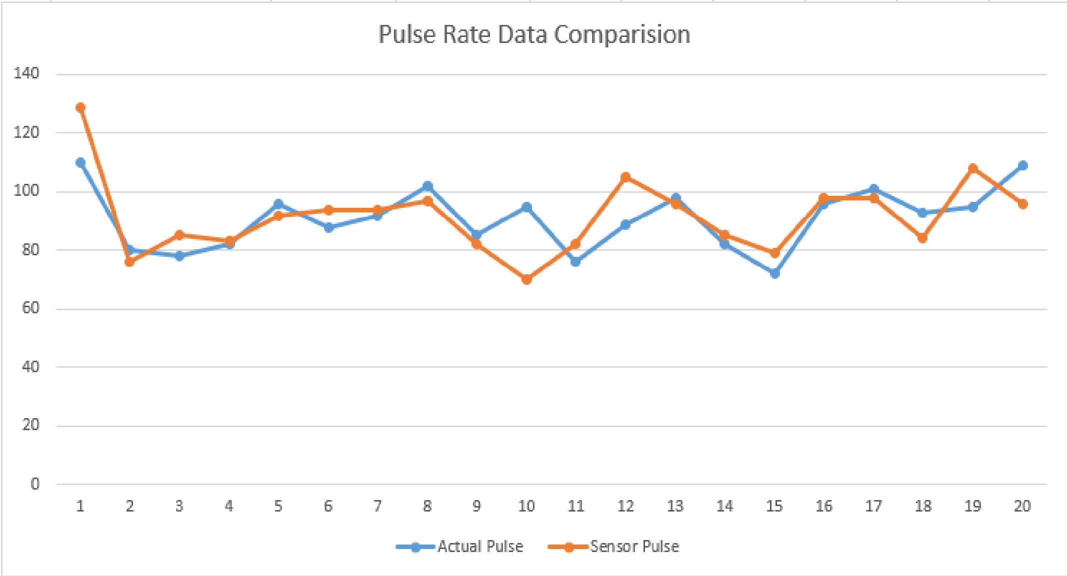


Figure 5.11: Comparison of actual and sensor generated pulse rate data

As shown in the figure 5.11 above, the blue line illustrates the actual pulse data output and the yellow line as the sensor pulse rate data output. For twenty different measurements, the sensor generated graph varies with actual data plot at peak values but also convene at certain points. This demonstartion also proves the feasibilty of the implemented system well.

The total comparisons for both temperature data and pulse rate data for either cases is demonstrated in the figure 5.12 below:

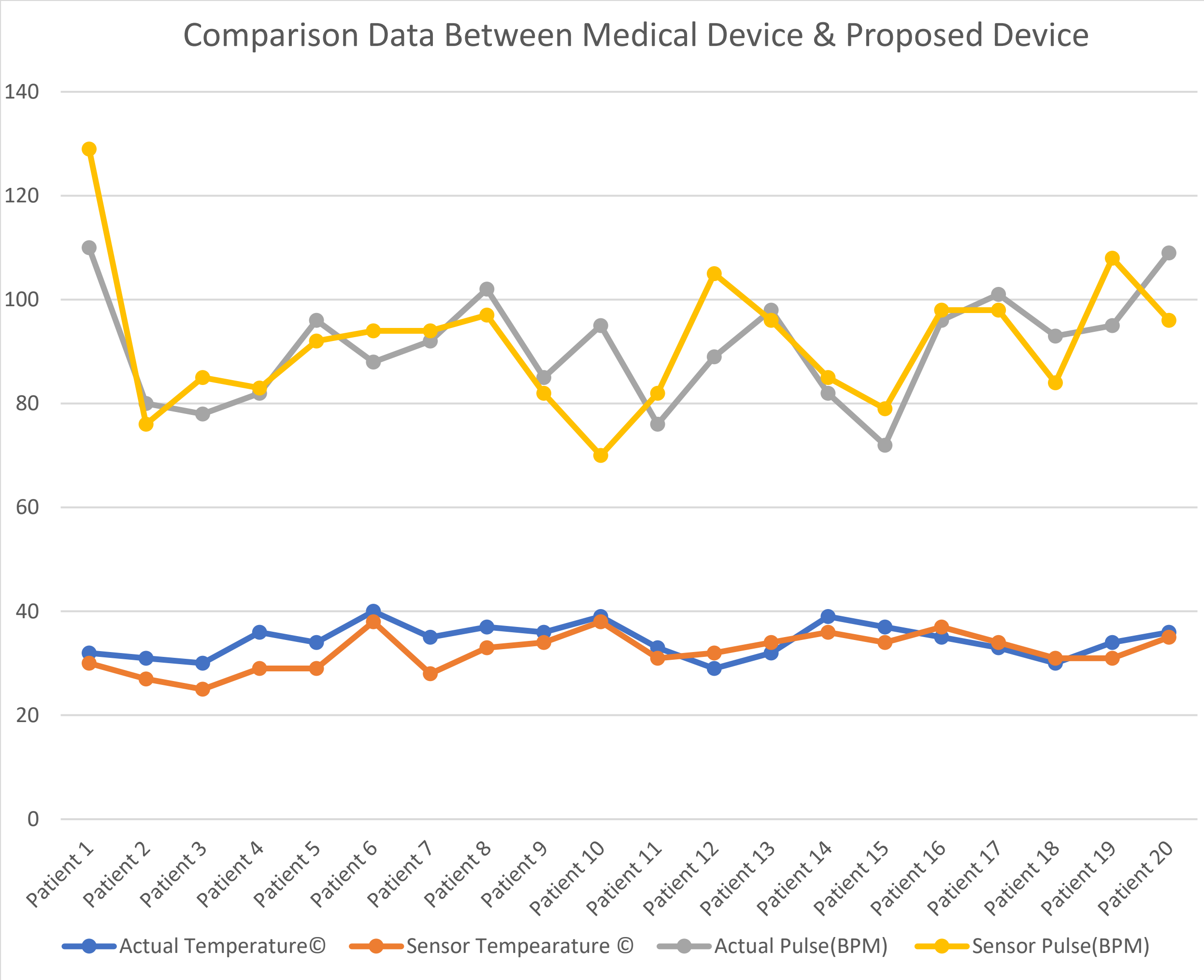


Figure 5.12: Comparison of actual and sensor generated temperature and pulse rate data

This figure shows us the differences between sensor generated data and actual medical device generated data for both cases. The dotted data chart for both cases is shown in the figure 5.13 below as well.

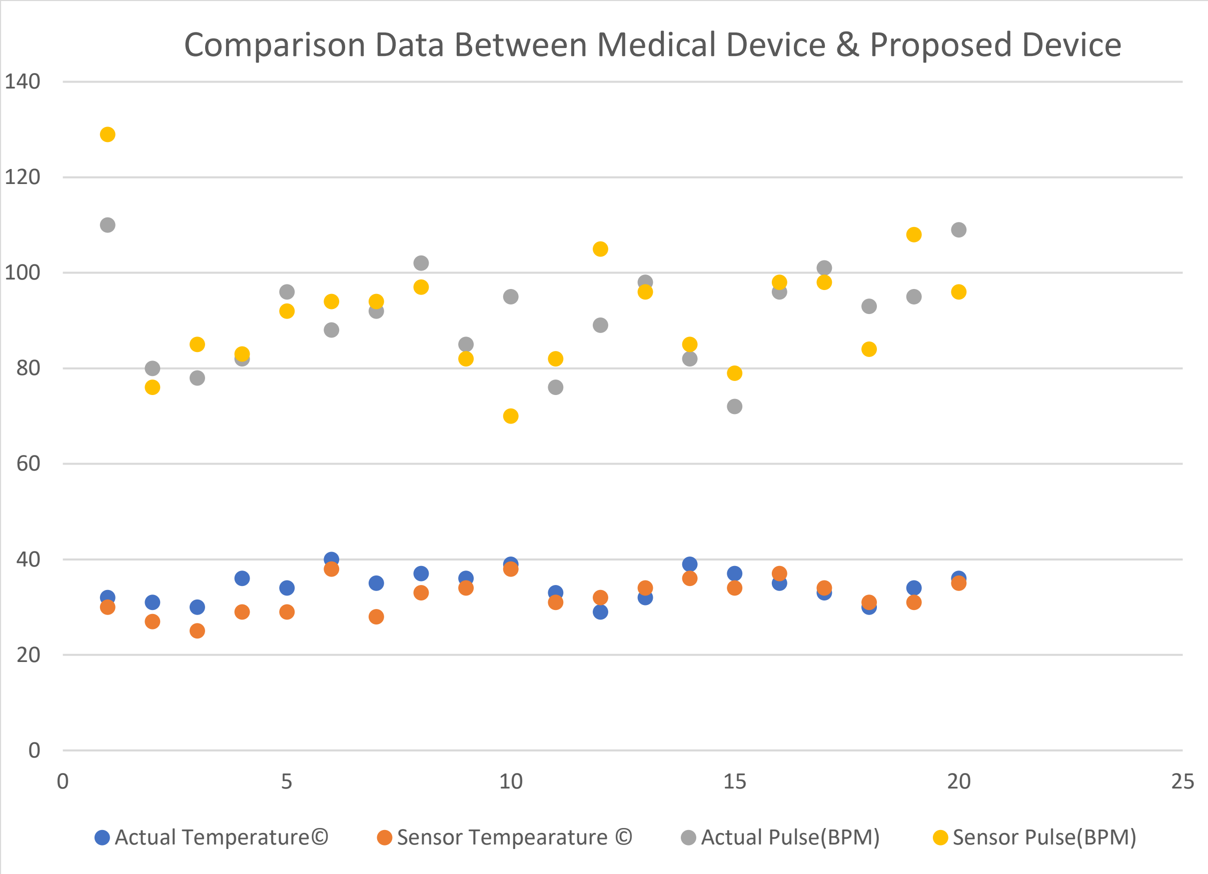


Figure 5.13: Comparison chart for both medical device and sensor generated temperature and pulse rate data

5.8.2 Performance Testing of Fog Node

After the sensor data are sent to the developed fog computer, it takes in, processes and sends the data to the fog server at a time. So, the whole 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. 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 II

PERFORMANCE EVALUATION IN TERMS OF COST, POWER, AND LATENCY

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

Table II 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 really quick and near real time throughout the layers. This determines that the developed health monitoring system is feasible in checking patient vital parameters effectively.

5.8.3 Performance Testing of ECG Signal

As Electrocardiogram (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 actual medical ECG machine. The ECG diagnosis produces some other results like features of ECG signals such as PR interval, P/QRS/T/QT intervals, QT complex, and heart rate in BPM (beats per minute). The resultant ECG waveforms are illustrated in figure 5.14 below:

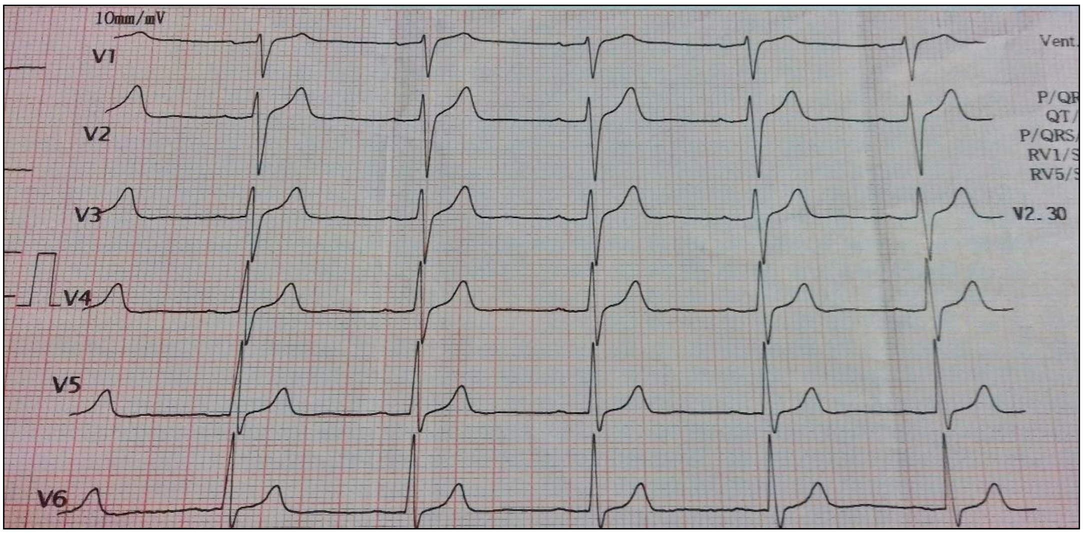


Figure 5.14: Graphical representation of 12-Lead ECG Signals (6 precordial leads) measured with clinical ECG machine

According to the figure 5.14, the 12-lead ECG signal measurement has been considered for comparing with our ECG sensor generated waveforms. For convenience, we consider one signal V4 from these six precordial signals of 12-lead ECG waveforms. Now the ECG sensor generated signal that is displayed in the Arduino serial plotter is shown below:

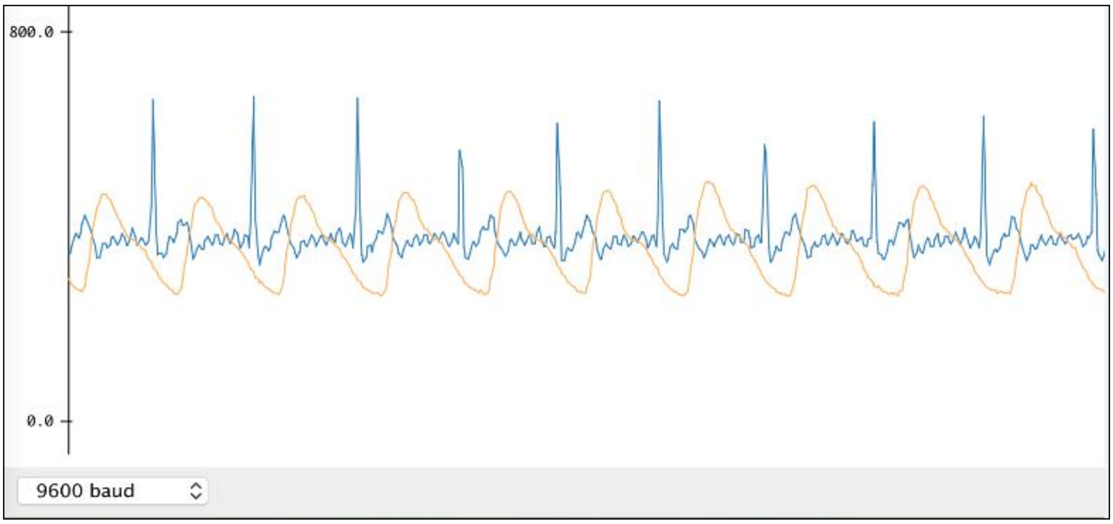


Figure 5.15: Graphical representation of generated ECG signals displayed in Arduino serial plotter

The figure 5.15 shows the ECG sensor generated signals that are displayed at Arduino serial plotter. This signal has been measured from the same demo patient by placing the electrodes and running the programs in the Arduino. Now considering a standard single lead signal V4 from figure 5.16, we observe that there is dissimilarity between the clinically diagnosed actual ECG signal and the sensor generated ECG signal. 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.

CHAPTER 6

CONCLUSION

In this work, we have successfully incorporated the fog computing concept in IoT-based health monitoring system architecture by establishing a fog-IoT driven health monitoring application. To procreate a simple, low-cost, low-energy, and highly efficient fog-based monitoring system, we introduced a combination of Raspberry Pi and Arduino UNO as fog computing node which will fetch, process, and store biosensor generated data from the sensor up to the server. The fog computers developed here are highly efficient in terms of latency, data processing, cost, energy, and network usage, thereby justifying the general concept of the fog computing system. Our accomplished application can effectively demonstrate all the requirements of the IoT-based healthcare system along with delivering the pervasiveness and ubiquity of the fog computing concept. The fog paradigm offers a relatively simpler, user-friendly, highly compatible architecture in any healthcare monitoring and management domain and newer establishments based on health information technology. 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. Besides, this system can highly assist medical caregivers, doctors, nurses by providing a way to monitor their respected patients remotely in real-time. In this way, the implemented patient health-monitoring system based on the 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 along with healthcare data processing, storing, managing and maintaining simultaneously.

REFERENCES

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

[2]. Khan, R. A., & Pathan, A. S. K. 2018. The state-of-the-art wireless body area sensor networks: A survey. *International Journal of Distributed Sensor Networks*, *14*(4), 1550147718768994.

[3]. Yigitoglu, E., Mohamed, M., Liu, L., & Ludwig, H. 2017, June. Foggy: A framework for continuous automated iot application deployment in fog computing. In *2017 IEEE International Conference on AI & Mobile Services (AIMS)* (pp. 38-45).

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

[5]. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. 2012, August. 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).

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

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

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

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

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

[11]. Pawade, A. S., & Jamgekar, R. S. 2018. MediCloud: Cloud Computing Services to Health Sector. *International Journal of Computer Applications*, *975*, 8887.

[12]. Cao, Y., Hou, P., Brown, D., Wang, J., & Chen, S. 2015, June. 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).

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

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

[15]. Maksimović, M. 2018. Implementation of Fog computing in IoT-based healthcare system. *Jita-Journal Of Information Technology And Aplications*, *14*(2).

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

[17]. Liu, Y., Dong, B., Guo, B., Yang, J., & Peng, W. 2015. Combination of cloud computing and internet of things (IOT) in medical monitoring systems. International Journal of Hybrid Information Technology, 8(12), 367-376.

[18]. Borthakur, D., Dubey, H., Constant, N., Mahler, L., & Mankodiya, K. 2017, November. 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).

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

[20]. Yi, S., Li, C., & Li, Q. 2015, June. A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data* (pp. 37-42).

[21]. Feldman, B., Martin, E. M., & Skotnes, T. 2012. Big data in healthcare hype and hope. *Dr. Bonnie*, *360*, 122-125.

[22]. Paul, A., Pinjari, H., Hong, W. H., Seo, H. C., & Rho, S. 2018. Fog computing-based IoT for health monitoring system. *Journal of Sensors*, *2018*.

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

[24]. Gia, T. N., Jiang, M., Sarker, V. K., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. 2017, June. 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).

[25]. Biswas, S. A Novel Fog and Cloud based Holistic Secure Framework for Smart Healthcare. *International Journal of Computer Applications*, *975*, 8887.

[26]. Barik, R. K., Dubey, H., Misra, C., Borthakur, D., Constant, N., Sasane, S. A., & Mankodiya, K. 2018. Fog assisted cloud computing in era of big data and internet-of-things: systems, architectures, and applications. In *Cloud computing for optimization: foundations, applications, and challenges* (pp. 367-394). Springer, Cham.

[27]. Chiang, M., & Zhang, T. 2016. Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, *3*(6), 854-864.

[28]. Al-Doghman, F., Chaczko, Z., Ajayan, A. R., & Klempous, R. 2016, October. A review on Fog Computing technology. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 001525-001530).

[29]. Dubey, H., Monteiro, A., Constant, N., Abtahi, M., Borthakur, D., Mahler, L., & Mankodiya, K. 2017. 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.

[30]. Gill, S. S., Arya, R. C., Wander, G. S., & Buyya, R. 2018, August. 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.

[31]. Hajji, W., & Tso, F. P. 2016. Understanding the performance of low power Raspberry Pi Cloud for big data. *Electronics*, *5*(2), 29.

[32]. Maksimović, M., Vujović, V., & Perišić, B. 2015, June. A custom Internet of Things healthcare system. In *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-6).

[33]. Chen, S., Zhang, T., & Shi, W. 2017. Fog computing. *IEEE Internet Computing*, *21*(2), 4-6.

[34]. Gia, T. N., Jiang, M., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. 2015, October. 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).

[35]. Dubey, H., Yang, J., Constant, N., Amiri, A. M., Yang, Q., & Makodiya, K. 2015. Fog data: Enhancing telehealth big data through fog computing. In *Proceedings of the ASE bigdata & socialinformatics 2015* (pp. 1-6).

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

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