



SHUBHCHINTAK

An efficient remote health monitoring approach for elderly people

Ayan Banerjee¹ · Dibyendu Maji¹ · Rajdeep Datta¹ · Subhas Barman¹ ·
Debasis Samanta² · Samiran Chattopadhyay^{3,4}

Received: 18 October 2020 / Revised: 8 November 2021 / Accepted: 30 May 2022 /

Published online: 9 August 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

With the proliferation of IoT technology, it is anticipated that healthcare services, particularly for the elderly persons, will become a major thrust area of research in the coming days. Aim of this work is to design a fit-band containing multiple sensors to provide remote healthcare services for the elderly persons. An application has been designed to capture health data from the fit-band, pre-process the data and then send them to cloud for further analysis. A wireless Bluetooth enabled connection is proposed to establish communications between sensors and the application for data transmission. In the proposed application, there are three different front-end interfaces for three different users: system administrator, patient and doctor. The data collected from the patient's fit-band are sent to a cloud data storage, where the data will be analyzed to detect anomaly (e.g., heart attack, sleep apnea, etc.). A Convolution Neural Network (CNN) model is proposed for anomaly detection. For the classification of anomaly, a Long Short Term Memory (LSTM) model is proposed. In the presence of anomaly, the system immediately connects a doctor through a phone call. A prototype system termed as Shubhchintak has been developed in Android/iOS environment and tested with a number of users. The fit-band provides data tracking with an overall accuracy of 99%; the system provides a response with 3000 requests in less than 100 ms. Also, Shubhchintak provides a real-time feedback with an accuracy of 97%. Shubhchintak is also tested by patients and doctors of a nearby hospital. Shubhchintak is shown to be a simple to use, cost effective, comfortable, and efficient system compared to the existing state of the art solutions.

Keywords Remote health monitoring · Wireless sensors · Health anomaly for elderly people · Cloud-based application · Artificial intelligence approach to health data classification

✉ Subhas Barman
subhas.barman@gmail.com

1 Introduction

In today's world, there are nearly 703 million elderly persons (aged 65 years or above) who form a significant 9.2% of world population (World population on ageing, United Nations, 2019). In the developing countries, like India, the percentage of senior citizens is increasing day by day and the share of the population of elderly persons in India is projected to increase to 20% in 2050 (United Nations, 2019). Besides this, health technology is rapidly evolving and has become an important part of our lives. CISCO [30] reported that 70 million devices or things are already linked to the Internet (in 2020) throughout the world. But this rapidly growing technology has its adverse effects too. The younger generation is too busy with their personal lives. They have no time to take care of their guardians. According to the World Health Organization (WHO) [4], elderly persons need immense care as 39.5% (27.7 million people in 2019) elderly persons died of Cardio-Vascular diseases like hypertension, strokes and diabetes due to negligence, sedentary lifestyle and improper food diet. Number of people with diabetes have been rapidly increasing from 108 million in 1980 to 533 million in 2019. But 65% senior citizens all over the world are often neglected and they feel helpless when medical emergencies arise because they are alone.

Of late, wearable technologies and mobile applications have taken health and fitness industries by storm [5–7, 26, 28, 57] (<https://www.samsung.com/global/galaxy/galaxy-fit/>). There are plenty of software products (in terms of features and functionalities) are in the market which can keep a regular check on our real time health conditions. Several software companies have developed their software SDK's [23, 24, 27]. They have developed several new mobile health apps for Ionic and Versa smartwatches. Some of the apps are developed on Fitbit's SDK [19]. Their names, merits, demerits, and usage are shown in Table 1.

In addition, there exist several Android-based applications like Apple Watch, Fitbit trackers, Jawbone band, Google Fit, Moov Now, Mi band, GOQii band, Nike Running, Runtastic, etc. These products track steps, basic metabolic rate (BMR), heart rate through a fitness band and display them using an application which is supposed to alert only the users but not caregivers. Moreover, wearable devices are very clumsy sometimes and they cause sufficient irritation. In some cases, data collection needs an active network connection; so, if internet is not available, the monitoring system cannot work. According to the survey report of AIMS 2019 [35], 90% of the existing healthcare monitoring solutions are mainly focussed on tracking and monitoring data without any feedback and the rest 10% solutions

Table 1 Existing healthcare applications and their functionalities

Name	Merits	Demerits	Usage
Aetna [36]	Provides information on health conditions; gives them step-by-step guidance to treat conditions	One has to feed one's own data manually	Primary prevention
Healthify Me [54]	Determined calory, provide proper diet plan from nutritionists	Have to pay the experts for every solution	Elderly care
Diplomat pharmacy [3]	Provides medication, prescription reminders	Works based on the existing data without any classification	Oncology care management platform
Dexcom [49]	Real-time glucose monitoring	Decreases the comfort level in tracking time	Useful for diabetes patient

provide feedback directly from the experts. These better solutions (with feedback) are very costly and in most cases, they need some manual intervention to feed information.

But all these solutions are still unable to address some major challenges. What if a person gets a heart attack suddenly and he or she is unable to call the doctor? How can the caregivers assess current health condition of old persons if they lack specific medical expertise? How can a hospital be alerted if an elderly person is in critical emergency? If we think on a large scale, these problems also exist in hospitals situated in a rural area where there is a dearth of doctors, nurses, and health workers.

This work aims to overcome the above drawbacks. We design a fit-band which contains ECG, Temperature, Blood Pressure, Blood Oxygen Level Detector, Accelerometer, and Gyroscope sensors. Then, the collected data is sent to an Android/iOS platform through Bluetooth connectivity; so, there is no need for an active internet connection at all. After that, the process is divided into two parts: one for the non prime admin and another for the prime admin. The basic difference is non-prime users can monitor up to 5 elders at a time but for the prime users, there is no such restriction on the number of elders being monitored. The application has a trained Machine Learning (ML) model for anomaly detection and pre-trained NLP tool for feedback to help the non-prime users. For prime users, the system is expected to deal with huge data. So, the system first sends the data to a cloud and there is a pre-trained ML model for anomaly detection and to predict health conditions. There is also a pre-trained NLP tool for chatbot to help the users by giving helpful advice based on their health conditions. If any anomaly is detected, the admin will get a notification and an SMS alert about the health condition of the users. There is also an SOS alarm to draw the attention of the neighborhood to help the users. If any serious anomaly is detected (e.g., heart-attack, sleep apnea, etc), the system will also automatically call a doctor and inform the admin about the condition. The proposed system is called Shubhchintak. Figure 1 depicts the features of Shubhchintak.

The main contributions of this work are listed below.

1. A smart low-cost solution having several unique features for monitoring elderly people is proposed.
2. A fit-band which contains ECG monitor, temperature monitor, blood pressure monitor, blood oxygen level detector, accelerometer, and gyroscope sensors with Bluetooth connectivity has been designed.
3. The proposed system incorporates real-time anomaly detection and feedback mechanism.
4. Doctors call on emergency, comfort of the elderly people and 3-way monitoring (users, admin, doctor) are considered.

The rest of the paper is organized as follows. Section 2 gives a state of the art survey and highlights the gaps. Section 3 describes the detailed methodology and algorithms to develop the application. In Section 4, outcomes are investigated, several executions are assessed and a comparison with similar approaches are presented. Section 5 concludes with a discussion on future extensions of the work.

2 Literature review

For the past few decades, several computational approaches have been developed across the world to study remote health monitoring. Recently, advances in technology have introduced telehealth systems to monitor health conditions of aging adults at their own residences. Moreover, different sensors are used in the remote monitoring system to record the health

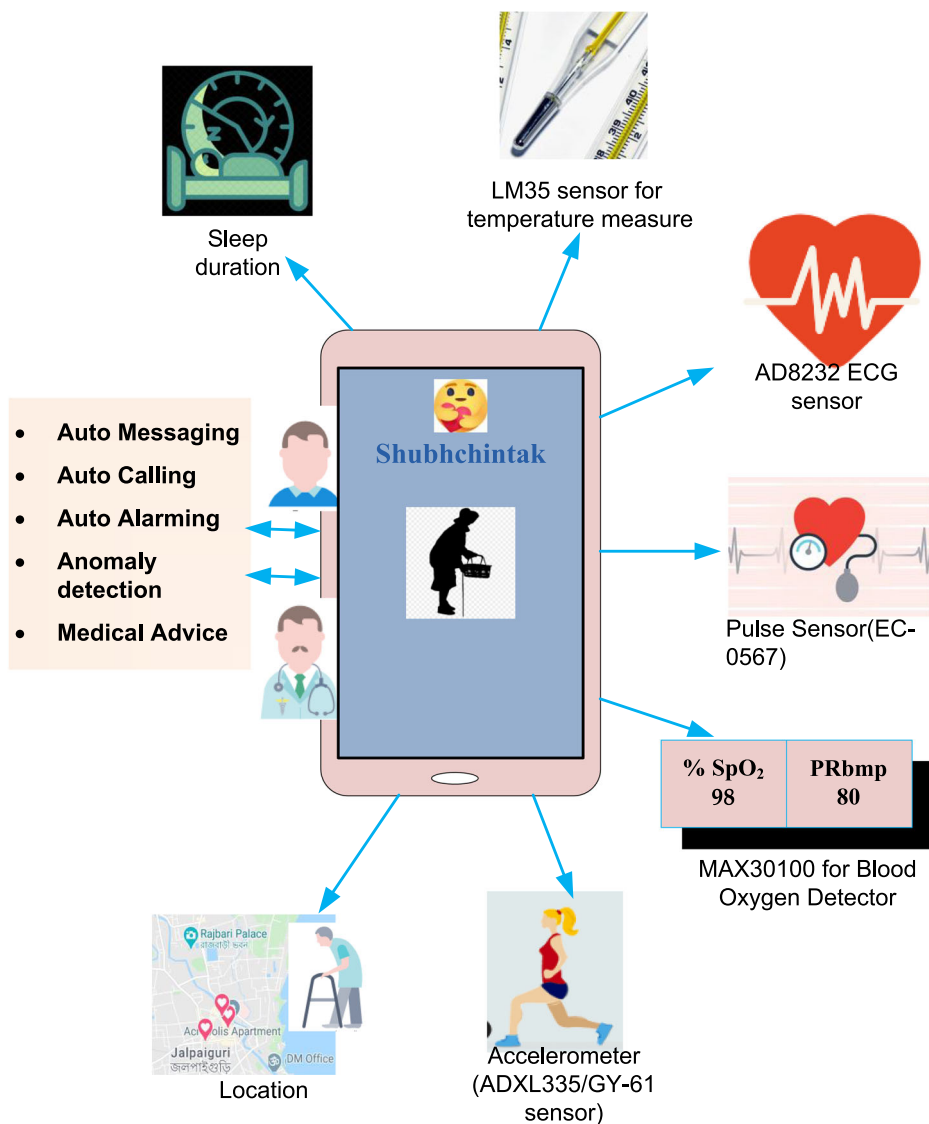


Fig. 1 Features of Shubhchintak healthcare application system

conditions of an aging adult for necessary measures. One's health condition depends on several factors like heart rate, blood pressure, body temperature, the oxygen level of blood etc. In 2003, Maiolo et al. [37] proposed a novel methodology to study the feasibility of the telemonitoring system for patients with severe respiratory illness. In that work, they investigated the feasibility of a telemonitoring service for patients with severe respiratory diseases. In 2004, Scalvini et al. [47] assessed the feasibility of nurse-led, home-based electrocardiography (ECG) monitoring for patients with chronic heart failure (CHF). This study considered seventy-four CHF patients and a fixed telephone line is used for ECG data transmission to a

service station. The study concluded that following up CHF patients using a nurse-led telecardiology program is feasible and useful. In 2006, Vitacca et al. [53] proposed a methodology to study the feasibility of telemedicine for home monitoring of patients with chronic respiratory failure and with mechanical ventilation assistance. They worked with pulsed arterial saturation (pSat) data of 45 patients and the facility was connected to a receiving station using a telephone modem for human teleconsultation. In 2007, keyhole plan recognition model for Alzheimer's patients was proposed by Bouchard et al. [15]. They claimed that 82% calls successfully recorded the pSat data and the study concluded that it was feasible to home monitor the patients of chronic respiratory failure. This work is based on lattice theory and action description logic to formalize the plausible incoherent intentions of the patient. In 2008, Keong and Yuce proposed a Low data rate ultra wide band ECG monitoring system [33]. In 2009, Bansal et al. [10] described a novel approach as a computer-based wireless system for online acquisition, monitoring and digital processing of ECG wave forms. Later in 2010, Atoui et al. [8] proposed a neural-network model for deriving standard 12-Lead ECGs from serial three-lead ECGs. In 2011, Bergmann et al. [14] proposed Body-worn sensor design for patients and clinicians and in 2012, Barnwell et al. [13] presented a novel solution of image-guided optimization of the ECG trace in cardiac MRI.

In 2015 Bansal et al. [11] proposed a novel methodology for detecting cardiac disorders as a remote health monitoring system. Catarinucci et al. [16] proposed an IoT-aware architecture for a smart healthcare system. Matthew et al. [17] proposed an IoT-based affordable remote health monitoring system for the elderly people using smart mobile devices and Baig et al. [9] proposed a novel mobile healthcare application. Also, Khoi et al. [34] proposed IReHMo for remote health monitoring. In 2016, Mallick et al. proposed a novel methodology for heart rate monitoring using fingertips through an Arduino-based software. By refining this mechanism, Srinivasulu et al. [48] proposed a measurement and wireless data transmission-based heart rate monitoring using a pulse sensor. Ghosh et al. [21] proposed a remote health monitoring system using IoT. In 2017, remote health monitoring regained its popularity. Valliappan et al. [50] proposed a low-cost design for a wearable remote health monitoring and alert system for elderly heart patients. Majumder et al. [38] discussed and compared different health monitoring systems in terms of wearable sensors. Nienholed et al. [39] proposed a novel solution based on a sensor system to identify a set of prescribed patient activities for home care treatment of elderly people. Patient activities data for an entire day are treated as big data and these are processed using different data analysis techniques. Pardeshi et al. [42] published a review of Raspberry Pi-based remote health monitoring systems. Later, Garbhapu et al. [20] proposed a low cost IoT-based single sensor node remote health monitoring system. In 2018, Saha et al. [45] proposed an advanced IoT-based remote health monitoring system, integrated with home automation and an alarm system. Later, Verma et al. [52] published a novel article on Fog assisted-IoT enabled patient health monitoring system in smart homes. In 2019, Al-khafajiy et al. [2], proposed a novel solution for health monitoring of the elderly through wearable sensors. In that work, they basically focused on AAL (Ambient Assistive Living) and AML (Automated Machine Learning). In this article, specific disorders are detected by tracking and analyzing people's physiological data. It has a better cost redundancy but it is unable to provide feedback based on the analyzed data. Later, in 2019, Duran-Vega et al. [18] proposed a remote health monitoring system using an IoT-based system for elderly adults that comprises wearable devices and mobile applications. It uses a biometric bracelet connected to a mobile application to visualize the health condition. In this research, feedback and performance analysis are not reported.

In 2020, Yeri et al. [59] presented a detailed review on IoT-based real-time health monitoring systems to showcase the shortcomings of the existing methodologies and suggested future improvements. Based on that, Vedaiei et al. [51] proposed an IoT-based system to monitor the physical distance in post covid pandemic. Moreover, it tracks cough rate, body temperature, blood oxygen saturation, and respiratory rate. However, it does not work with a stable 4G/5G/Wi-Fi connection and the system does not have any feedback mechanism. Ghosh et al. [22] also proposed a real-time energy efficient health monitoring system. The fit-band used in this sytem can work upto 18 hours without recharge but the system is not sufficiently intelligent. Wu et al. [56] designed a healthcare application to measure different physiological signals like photoplethysmography (PPG), electrocardiogram (ECG), and body temperature with rigid flex structure. However, it is noted that a mobile gateway is essential to use this system. Huifeng et al. [29] proposed a continuous health monitoring system which is dedicated for a sports person; the system monitors heart rate, step counts, and BMR but does not have any feedback mechanism and lacks doctors-on-time feature. Hosseinzadeh et al. [25] proposed an IoT-based environment to collect vital health data through smart healthcare technology and the collected data are analyzed using machine learning algorithms to check vital signs and detect biological and behavioral changes of disabled or elderly persons. But, this system is not comfortable to users as they have to wear several sensors or have to lie down on a bed for remote tracking. Zhong and Li [60] designed and developed a health monitoring system to recognize the physical activities and monitor health conditions of college students through the connection of heterogeneous economically-efficient wearable gadgets in an open environment. But, it needs sufficient domain knowledge to use. Recently, in 2021 Saha et al. [46] obtained a working prototype having temperature sensor (DS18B20) and Arduino for health monitoring. However, temperature alone cannot provide details of all health conditions. Pandya et al. [41] proposed a smart aging wellness sensor networks for anomaly detection and generating alerts in near real time. But, the system is very bulky, it requires a continuous internet connection and anomaly detection accuracy is not good. Juyal et al. [31] proposed an IoT-based health monitoring system for skin diseases; however, sensors have a complex circuitry so that it is not comfortable to wear the gadget. Bansal et al. [12] proposed health care services using organ simulation though it is not in real-time; but it is the most developed application till date.

All methods discussed in this survey are mainly focused on accurate remote health monitoring. The challenges is that a lot of manpower is still required for continuous and

Table 2 Distinguishable features of Shubhchintak, the proposed application

Features	State of the art approaches	Shubhchintak
Manufacturing cost	High	Low
Data transfer	Internet/Wi-Fi	Bluetooth
Features	Heart rate, BMR, Sleep hours, Steps count	Heart rate, Step count, Blood oxygen level, BMR, Monthly health condition Temperature, ECG
Feedback mechanism	No	Yes
Anomaly detection	Very poor	Efficient
Doctors-on-emergency	No	Yes
SOS alarm	Yes	Yes
Offline simulation	No	Yes

uninterrupted monitoring. Otherwise, if an emergency occurs, patients will be unable to inform relevant persons and may die. To overcome this shortcoming, the proposed system Shubhchintak is attributed with some notable features. Besides monitoring, it also provides feedback. This feedback (based on the analysis of the monitored data) is real-time and works even when users are offline or when users are not wearing the fit-band. It also generates an SOS alarm to alert the neighbors, message the admin to inform him/her about the situation and call a doctor immediately (i.e., have the intelligence to tackle the emergency situation), to save his/her life in a medical emergency. So, there is no need for constant monitoring in our proposed system. This system also helps elders who need utmost care in a comfortable manner in COVID-19 pandemic. Shubhchintak is a very low-cost solution as well. The distinguishable features of Shubhchintak over the state of the art approaches are reported in the Table 2.

3 Proposed methodology

The proposed method to take care of the elderly people is as follows. It has three main phases namely, data collection, anomaly detection, and feedback generation. A brief overview of the proposed methodology is shown in Fig. 2. The phases are described in details in the following.

3.1 Data collection

In this work, a fitness band has been designed and developed to collect the information needed for proper monitoring and feedback. This fitness band contains an Accelerometer (ADXL335), Gyroscope (L3GD20H), Pulse Sensor (EC-0567), temperature sensor (ADA165), and a blood oxygen level detector. Figure 3 gives a brief overview of the sensors and Fig. 4 gives a brief overview of the circuit diagram of the proposed fitness band.

The circuit diagram (Fig. 4) shows the connection between six different sensors and NodeMCU module. The NodeMCU version 2 microcontroller is chosen for the smart band which is driven by ESP8266 microcontroller by Expressive systems. It is a low-cost module with integrated WIFI connectivity. In this circuit, the sensors and the module are charged using a 3v to 5v source based on requirements from external power source and necessities of the components. The data pins from the sensors are connected to the data pins in the NodeMCU module. Besides that, the NodeMCU module has 8 data pins (D1-D8) and a ground pin. The ground connection of all the 6 sensors is collectively connected to this ground pin of NodeMCU. The data provided by each sensor is connected to one of the 8 pins in the NodeMCU module. The data pin of the accelerometer sensor is connected to the D0 pin of the NodeMCU Module. Similarly, the pulse sensor, temperature sensor, gyroscope sensor, blood oxygen detector sensor, and the ECG sensor are connected to D1, D2, D3, D4, D5 data pins respectively of the NodeMCU module. The microcontroller unit (i.e., the NodeMCU module) performs a calculation based on the data received from the sensors and passes the result to the decision controller. It consists of predefined functions that take action based on the result of the module.

Different sensors used to design the fitness band are discussed in the following.

Accelerometer A thin, small, low power, 3 axis sensor named ADXL335/GY is used. By using tilt-sensing, it can measure static acceleration of gravity as well as dynamic acceleration that occurs due to motion, vibrations and shock. This sensor consists of 5 pins, which

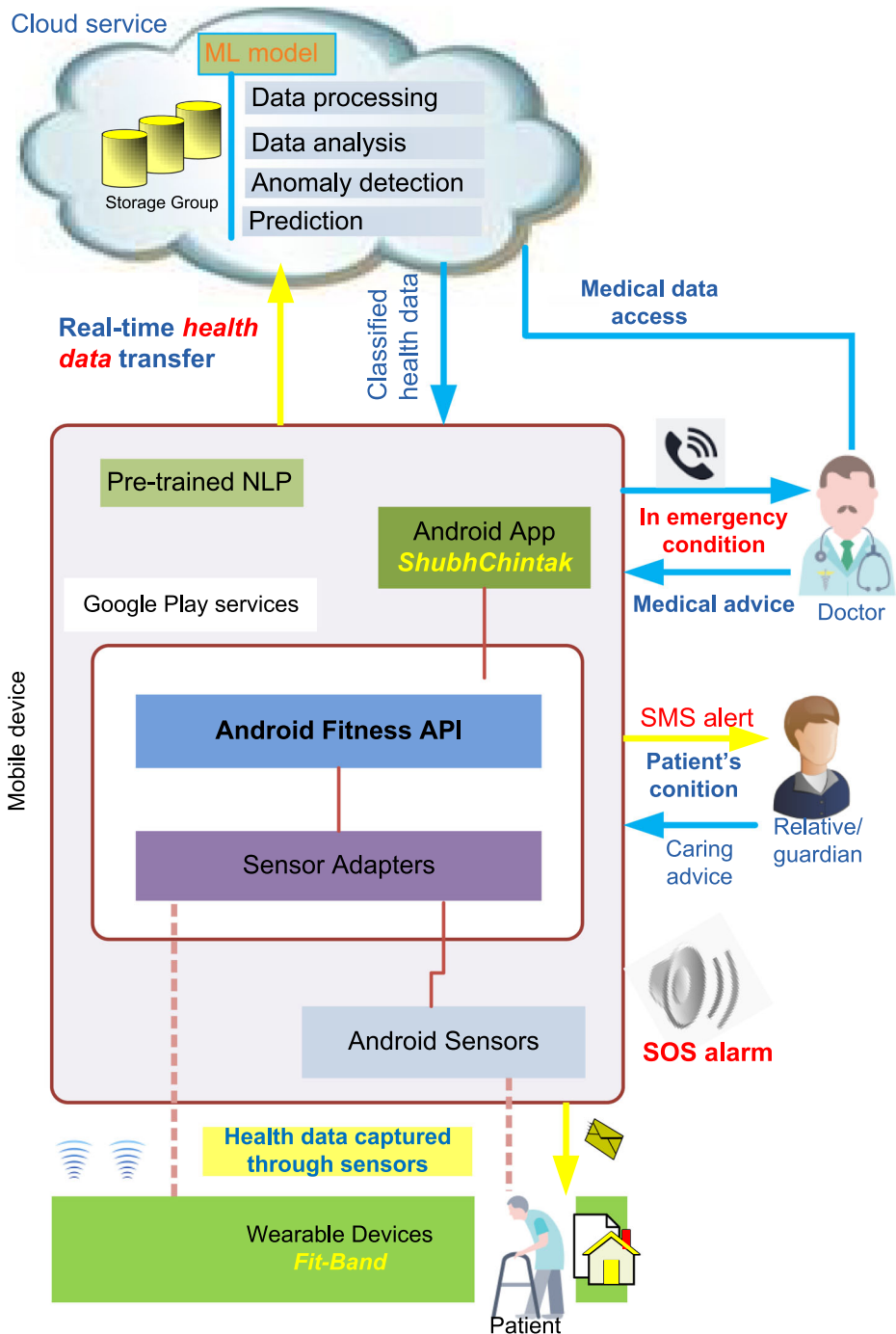


Fig. 2 A system-level architecture of Shubhchintak

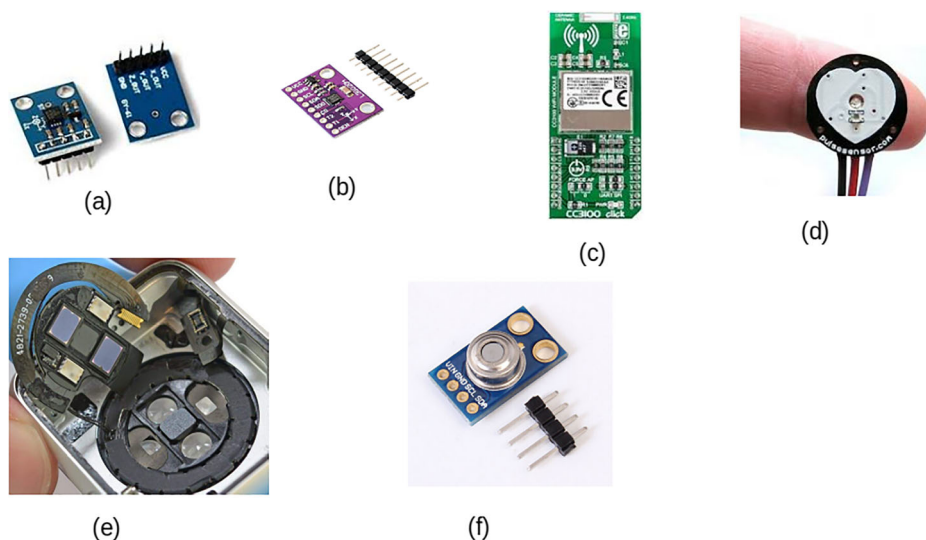


Fig. 3 Brief Overview of sensors (a) accelerometer, (b) gyroscope, (c) ECG sensor, (d) pulse sensor, (e) blood oxygen level detector, (f) human body temperature sensor

are: i) GND: this pin is connected to ground, ii) VCC: this is connected to + 3.3V, iii) X- This is connected to analog pin(A0), iv) Y-NIL and v) Z-NIL.

ECG sensor A small sensor named AD8232 ECG is used for recording electrical signals of the heart and the output can be obtained in an analog form. To obtain clear signal from the PR and QT intervals, the module acts as an op-amp. LO+ and LO- are connected to digital pins and the output pin provides analog signal; so, it is connected to the analog pin.

Temperature sensor LM35 sensor can measure temperature in the range of -55°C to 150°C . It provides an analog voltage proportional to the temperature. Higher the temperature, the higher is the output voltage. The output is converted to digital form using ADC for processing in the microprocessor.

Gyroscope sensor The sensor named LSM6DS3 is used for detecting motion. Its full scale acceleration ranges along $\pm 2/\pm 4/\pm 8/\pm 16\text{g}$ and angular rate ranges along $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000\text{ dps}$

Blood oxygen detector A sensor, named MAX30100 is used for blood oxygen detection. It works using 1.8V and 3.3V power supplies. It has two LEDs, one is red and the other is infrared for measuring oxygen levels in the blood. It uses the principle that oxygenated blood absorbs more infrared light and passes more red while deoxygenated blood absorbs red light and passes more infrared light. The sensor reads the absorption levels for both lights and provides an output.

3.2 Application development

Google Fit API consists of a Fitness Store, where data cloud stores health data of all specified users from their wearable devices and the android phones [27]. The sensor framework

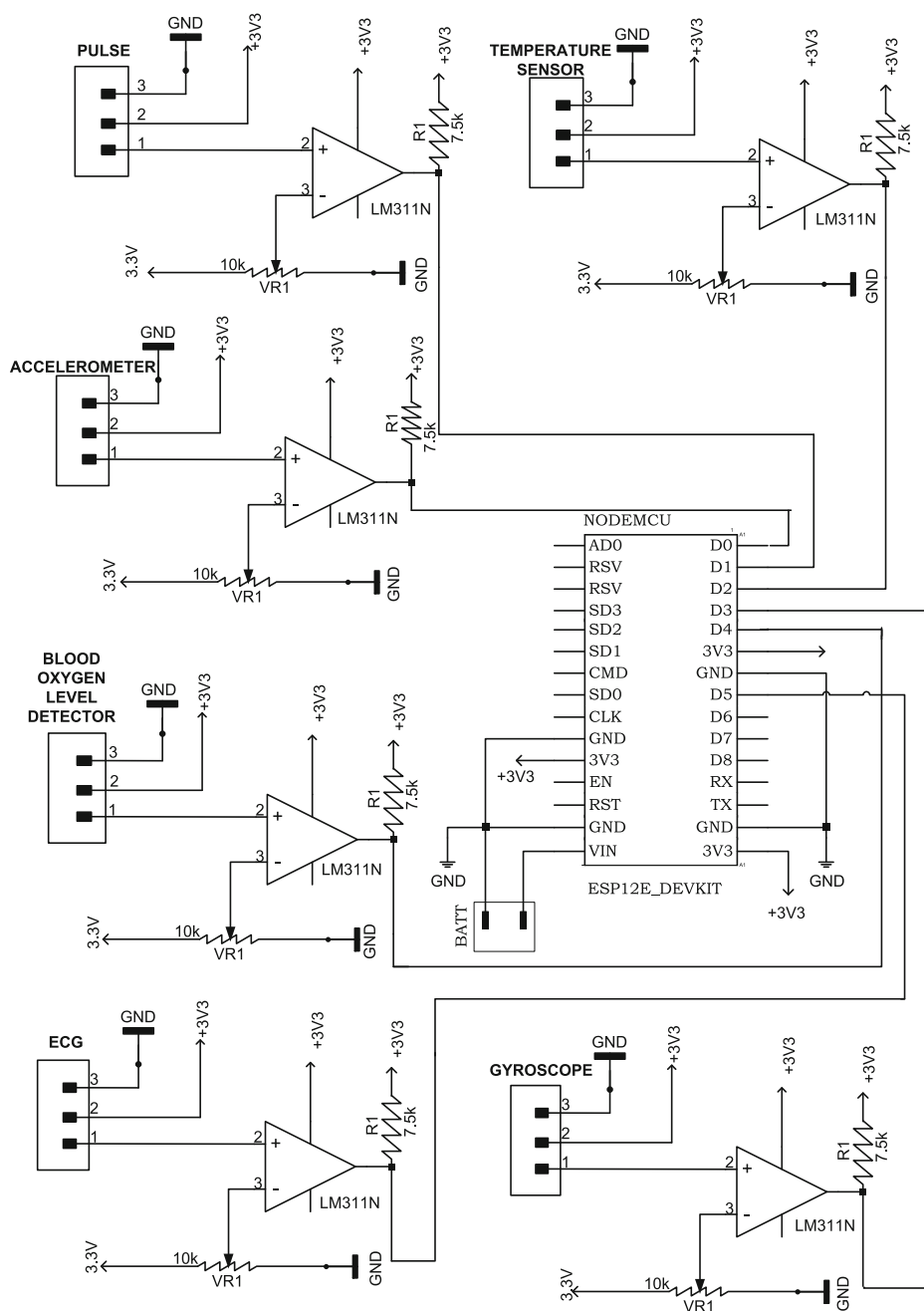


Fig. 4 A brief overview of the circuit diagram to design the fitness band

is a wholistic representation of sensors and fitness types like step count and heart rate, which has the ability to query and interact with the database. We have used Android Studio for the development of the proposed app using Google Fit API.

To build the application the following steps are to be carried out.

1. Create a new project by adding the dependencies in the Gradle file. Add the dependencies 'play-services-fitness' and 'play-services-auth' in the apps Gradle file.
2. Create the API client as follows. Create a FitnessOptions instance, declaring the data types and access type. The Data Types needed for the proposed application are 'AGGREGATE-HEART-RATE-SUMMARY', 'TYPE-STEP-COUNT-CUMULATIVE', and 'TYPE-STEP-COUNT-DELTA'.
3. Get an instance of the Account object to be used with the API. Check if the user has previously granted the necessary data access, and if not, initiate the authorization flow. If the user is using the app for the first time then, they have to give access to their Gmail account.
4. The application has to subscribe to the required Fitness APIs. In our app, the following APIs are to be subscribed: a.AGGREGATE-HEART-RATE-SUMMARY b.TYPE-STEP-COUNT-CUMULATIVE.
5. Write a method 'queryFitnessData()' to get the heartbeat data from the fit-band. The heartbeat data is taken by reading the 'TYPE-HEART-RATE-BPM' DataType of the Google Fitness API after a constant interval of one second.
6. Write a method 'void readData-steps()' method to get the Steps Counter data from the cloud. The step-count data is taken by reading the 'TYPE-STEP-COUNT-DELTA' DataType of the Google Fitness API. The total steps in a day are calculated and displayed which is refreshed every 5 seconds.

ECG, temperature, blood pressure, blood oxygen level, accelerometer data, and gyroscope sensor data are collected through the connected bluetooth device. The data is then gathered and displayed to the users. To fetch the data from the bluetooth device, BluetoothAdapter is used in the following manner.

```
private BluetoothAdapter bAdapter = BluetoothAdapter.getDefaultAdapter();
Then the Individual data is received as follows:
//To get the BluetoothSocket input streams
InputStream dataIn= null;
dataIn= socket.getInputStream();
DataInputStream mmInStream = new DataInputStream(dataIn);
byte[] buffer = new byte[256];
// Read from the InputStream
bytes = mmInStream.read(buffer);
//Data is received as a Text Field.
String SensorData= new String(buffer, 0, bytes);
```

3.3 Anomaly detection and feedback

In this phase, two different mechanisms are used for two different types of users. For non-prime users there is a TensorFlow lite backend with this application and there is an in-built pre-trained model to detect the anomaly and classify the detection; then, this classified data is sent to a pre-trained NLP to give feedback based on this classification. But for prime users we have to send the data to a cloud because a large volume of data will be generated for prime users which may make the application slow. In the cloud, further processing and anomaly detection are performed and the feedback is generated. This feedback is returned to the application. Figure 5 gives an overview of anomaly detection in the case of non-prime users and prime users.

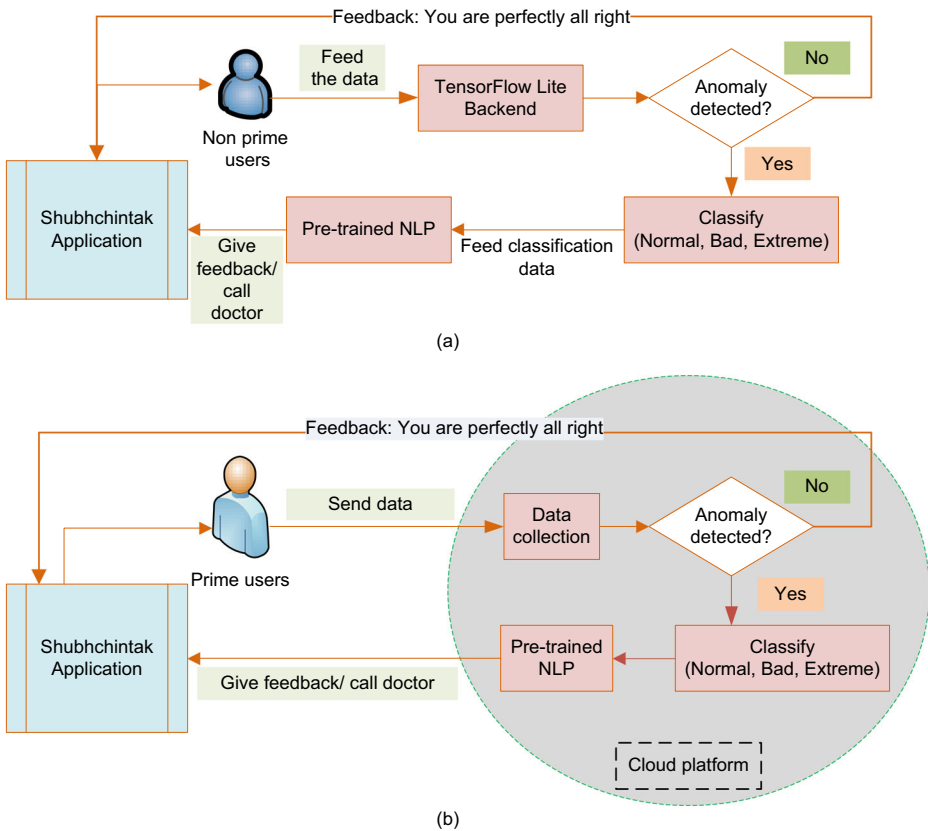


Fig. 5 Anomaly detection and feedback mechanism in case of (a) non-prime users and (b) prime users

3.3.1 Data storage in cloud

To handle the huge amount of data, a cloud-based mobile data storage application system is developed. We consider SAAS (Software as an administration) for data gathering. It is a cloud computing resource, specially designed for the administration of a software, without any knowledge of that software in the user device. It permits gathering and manipulating data through an Android portable platform and the goal is application developers can focus on the processing steps. A virtual machine (VM) instance is instantiated by running a container service in its own memory space so that the cloud server can gain multi-user ability with server virtualization.

3.3.2 Anomaly detection mechanism

For anomaly detection, a convolutional neural network (CNN) is used. The convolutional layers perform convolution with the stride size 1, and a set of 100 filters. where the kernel size is set to 3×1 for all 4 convolutional layers. It is shown in Fig. 6. At "Conv1," the size of the input vector is 263×1 . Upon convolution, the output obtained is passed through the ReLu activation, which performs element-wise nonlinear activation. The feature map thus

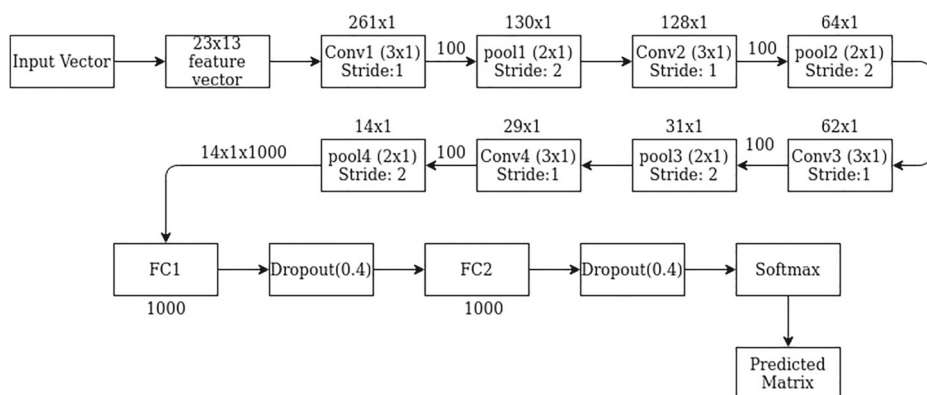


Fig. 6 Proposed CNN architecture with filter and kernel for each convolutional and pooling layer

obtained is of dimension 261×1 . This output feature map from Conv1 acts as the input to the first pooling layer “Pool1.”

The pooling size of 2×1 with 100 filters, and pooling stride of size 2 are considered for all pooling layers. At Pool1, the “max pooling function” is considered and it extracts the maximum element on the feature map, thereby resulting in a 130×1 output feature map. This output feature map becomes the input to the second convolutional layer “Conv2” with kernel size 3×1 and stride 1, comprising 100 filters, which then outputs a feature map of size 128×1 . This, in turn, acts as the input for “Pool2” with the same pooling size and pooling stride as considered in Pool1. Pool2 outputs a feature map of 64×1 vector. “Conv3” receives this as the input and performs the convolution over the vector to arrive at a 62×1 feature map. The “Pool3” layer performs “max pooling” over the feature map obtained from Conv3, resulting in a 31×1 output. The output feature map of Pool-3 is fed into “Conv4,” which performs convolutional operation with a 3×1 kernel and stride 1. Conv4 layer generates a feature map of size 29×1 , which is fed as input to “Pool4,” which in turn performs pooling similar to the previous pooling layers. Finally, we obtain a feature map of size 14×1 , which acts as the input to the fully connected layer “FC1.”

The fully connected layers FC1 and FC2 have 1000 neurons each and are connected with a “dropout layer”, which “drops-out” a set of activations randomly. Hence, it temporarily nullifies the contributions of the corresponding neurons in the forward pass. The dropout technique helps in overcoming the overfitting issue faced by CNNs, by allowing CNN to be less sensitive to the weights of the dropped out neurons, thereby providing the correct classification even in absence of those. The output layer consists of three neurons, i.e., one neuron for each class. A “softmax operator” is applied on FC2, which performs the final classification. The weights and biases for all layers are initialized with random values using a normal distribution of mean zero and unit standard deviation. The feature vectors act as inputs to the CNN, which go through the forward pass (convolution, ReLU, and pooling operations), and ultimately probabilities of the block belonging to individual classes are computed. Depending on the deviation of the actual probability distribution from the predicted, the weights and biases are updated in proportion to their contribution to the error via back-propagation. As CNN learns, the weights and biases get updated. “Softmax cross-entropy loss” is adopted as the loss function by us in this work. Adam optimizer is used as the optimizing algorithm for training the network, and the learning rate is set to 0.00001.

3.3.3 Trained NLP for feedback

Based on the predicted matrix, it classifies the anomaly into 4 categories i.e., normal, slightly abnormal, moderately abnormal, and extremely abnormal. Based on these 4 classes, an NLP is designed and trained. Unsupervised deep learning is used for this conversational chatbot. Figure 7 gives a brief overview of the NLP model. This model takes ABC as input and produces WXYZ as output. It stops predictions after outputting EOS (end of the sentence). The LSTM reads the input sentence in the reverse order for ease of optimization.

In this model, the recurrent neural network (RNN) [44, 55] goes through a natural generalization of feed-forward neural networks in terms of sequence to sequence mapping. Given a sequence of inputs ($g_1, g_2, \dots, g_n$), a standard RNN computes a sequence of outputs ($f_1, f_2, \dots, f_n$) by iterating on the following equation:

$$mid_t = \sum_{i=1}^n W^{midx} \cdot g_t + W^{midm} id \cdot mid_{t-1} \quad (1)$$

$$f_t = W^{ymid} \cdot mid_t \quad (2)$$

where, W denotes the weights of the neural networks, f_t = trainable features at t^{th} iteration, mid_t denotes the midpoint of all features of the t^{th} iteration, $midx = x$ variance of the midpoint and $midy = y$ variance of the midpoint and g_t is the gaussian normalization factor.

Then, decoding and rescoring are done. A large deep LSTM is trained based on anomaly classification. This training is done by maximizing the log probability of a correct translation X given the source sentence Y .

$$\log prob_{\max} = \frac{1}{S} \sum_{i=1}^S \log_{10} P(X_i | Y_i) \quad (3)$$

Where Y is the training set. Once training is completed, the translations are produced by finding the most likely translation according to the LSTM:

$$X' = \prod_{i=1}^n \max(P(X_i | Y_i)) \quad (4)$$

The advantage of these models is their simplicity and their generality, with little need for domain-dependent rules.

4 Experimental results and analysis

Experiments are carried out based on real-time data collected from volunteers who have worn the fitness band. As a result, accuracy of the proposed model depends heavily on the

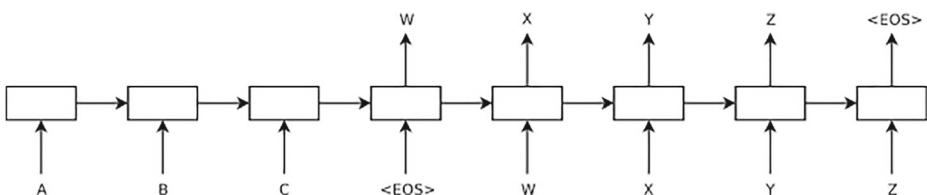


Fig. 7 Overview of NLP model

Table 3 Performance comparison of sensors used in the proposed system with ISO standard

Sensors	ISO			Shubhchintak		
	min_TPR	Max_FPR	Accuracy	TPR	FPR	Accuracy
Accelerometer	0.8000	0.1500	0.9500	0.9010	0.1113	0.9905
Gyroscope	0.8000	0.1500	0.9500	0.9054	0.1170	0.9944
ECG sensor	0.8000	0.1500	0.9500	0.9381	0.1274	0.9978
Blood pressure sensor	0.8000	0.1500	0.9500	0.9377	0.1354	0.9982
Blood oxygen level sensor	0.8000	0.1500	0.9500	0.9504	0.1137	0.9911
Human body temperature sensor	0.8000	0.1500	0.9500	0.9054	0.1157	0.9914

accuracy of the sensors. So, we used high-quality sensors for this project. The performance analysis of the individual sensors is presented in Table 3. The International Standard Organization (ISO) has fixed the minimum true positive rate (TPR), maximum false positive rate (FPR), and minimum accuracy level requirements for any sensor [58] to be considered as an acceptable sensor. So, a liberal comparison is also performed in Table 3. In the following, the experimental setup and the context of the experiments are described.

4.1 Data description

Shubhchintak deals with patients data directly similar to [2]. A number of volunteers were engaged for the experiments. These volunteers are drawn from the patient data-base of a local hospital. These volunteers wore fitness band and then, data were collected. The data so collected was used as a dataset. Thus, the data set on which experiments were carried out are real-life and much more reliable than any benchmark data set. The detail of the experimental setup is explained in the following subsections.

4.2 Experimental setup and environment

Firstly, the fitness band is designed according to the Fig. 4 with the required components shown in Fig. 3. Not only that, to show the adaptative capability of the design, this experiment is also carried out through Mi fitness band. Once the fit-band design is implemented, the next phase is to design the android application. The Shubhchintak application is developed using the Android Studio IDE. The application is run on an Intel i5 Core processor with 8GB RAM and 2GB NVIDIA Graphics Card. In the next step, the sensors' data are sent to Google Cloud for anomaly detection and feedback generation through pre-trained NLP. The details of this cloud architecture is mentioned in Table 6. All machine learning models have been developed using python tensor-flow framework. An android smartphones are used to test the developed architecture.

4.3 Android application

The application is developed based on the descriptions of Section 3.1. It collects data from the fitness band and processes the data in real-time. It contains 4 main pages. The first is the login page. The user enters the username and the password. Based on the saved logins they can use prime and nonprime features. Prime users have to pay charges for cloud processing. Features of the android application are shown in Figs. 8 and 9.

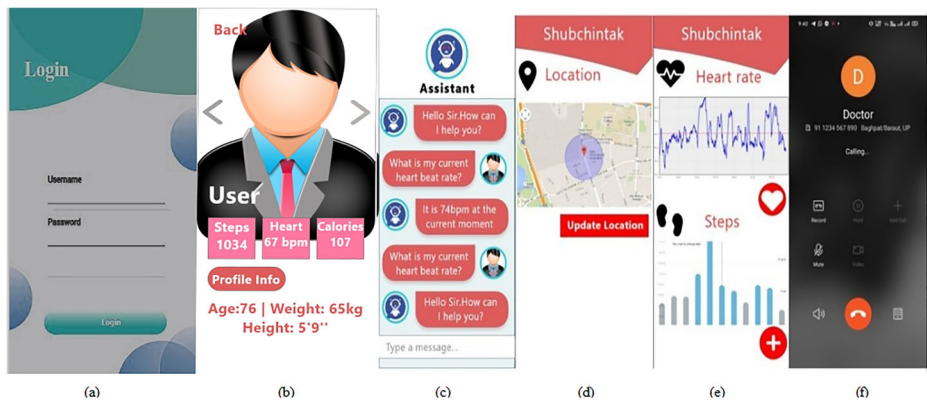


Fig. 8 (a) Login page (b) Main page (c) Chatbot assistant (d) Geo-Sensing (e) Health data (Heart Rate and Step Count) and (f)Auto calling screen of the Shubhchintak application

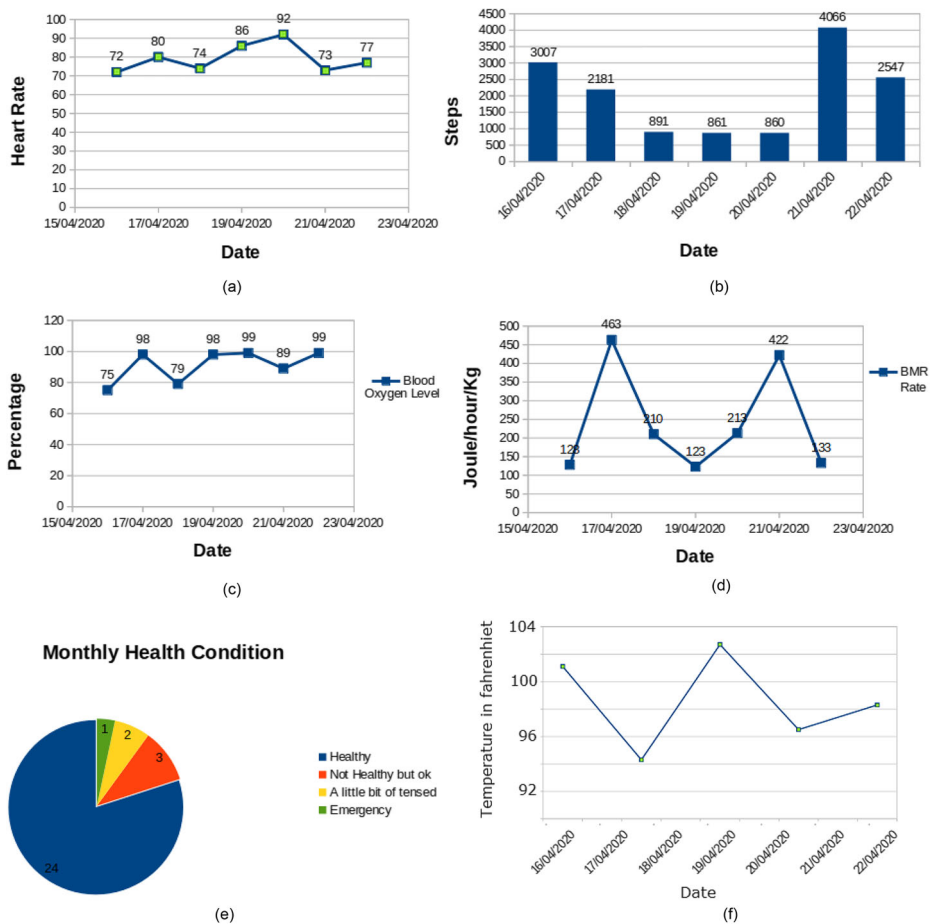


Fig. 9 Health data of the shubhchintak application. (a)Heart Rate,(b) Step Count (c) Blood Oxygen Level, (d) Basic Metabolism Rate, (e) Monthly Health Condition (f) Temperature

Table 4 Performance analysis of the anomaly detection model

Block size	Min_Accuracy	Accuracy (Shubhchintak)	Min_F1-score	F1-score (Shubhchintak)	Min_Success rate	Success rate (Shubhchintak)
16 × 1	0.8500	0.9843	0.9000	0.9617	0.9000	1
32 × 1	0.8500	0.9854	0.9000	0.9666	0.9000	0.9999
64 × 1	0.8500	0.9824	0.9000	0.9611	0.9000	0.9988
128 × 1	0.8500	0.9888	0.9000	0.9699	0.9000	1

4.4 Anomaly detection

Based on the collected data from Shubhchintak application, anomaly detection is performed as described in Section 3.3.2. The predicted matrix gives the true positive (TP), false positive (FP), true negative (TN), and false-negative value (FN) values. From these values, accuracy (ACC), precision, recall, F1 score, and success rate (SR) are calculated as follows:

$$ACC = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \quad (5)$$

$$Precision = \frac{|TP|}{|TP| + |FP|} \quad (6)$$

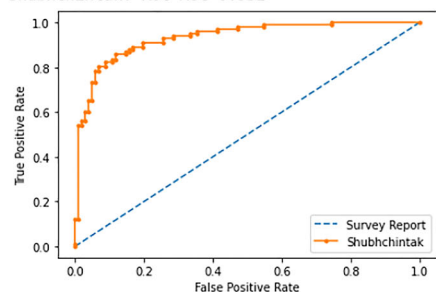
$$Recall = \frac{|TP|}{|TP| + |FN|} \quad (7)$$

$$F1Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (8)$$

$$SR = \sum_{i=1}^N \frac{\delta_{F1Score}}{N} \quad (9)$$

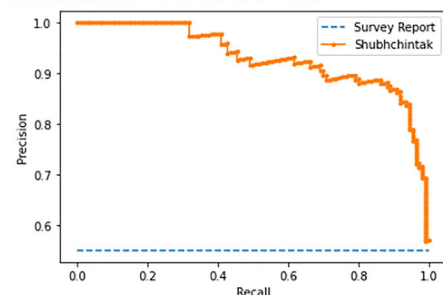
Based on the above equations, performance analysis of the anomaly detection model is calculated which is presented in Table 4. In [1], researchers made a survey using data mining techniques and concluded that the acceptance of any machine learning algorithm is based on the above mentioned parameters. So, if any model obtains higher accuracy, F1-score, and success rate, the model has a higher acceptance rate.

Survey Report: ROC AUC=0.500
Shubhchintak: ROC AUC=0.932



(a)

Shubhchintak: f1=0.972 auc=0.981



(b)

Fig. 10 True positive rate vs false positive rate and precision-recall curve for anomaly detection model

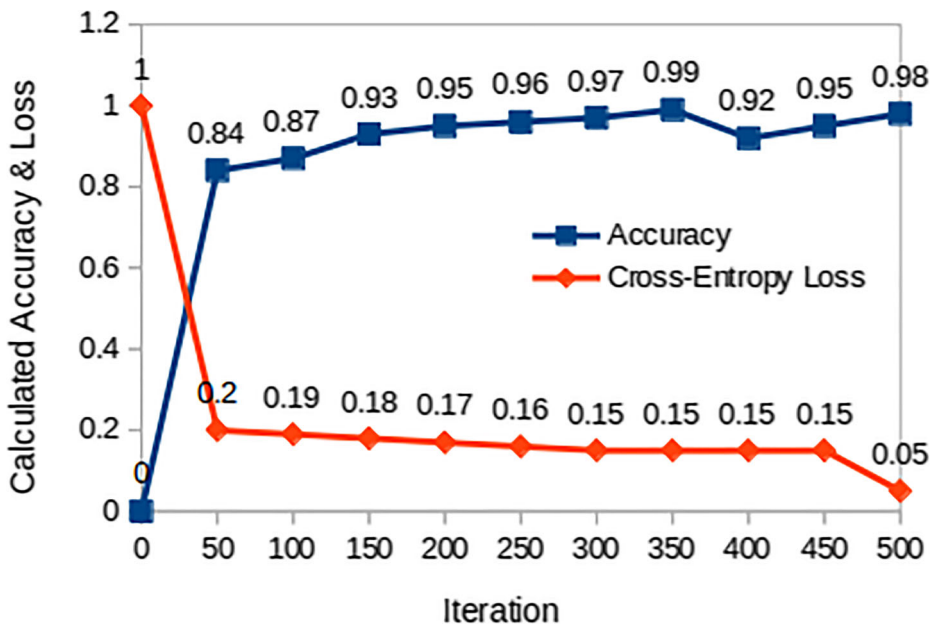


Fig. 11 Classification accuracy and cross-entropy loss of anomaly detection model

The comparison is graphically depicted in Fig. 10. This comparison is based on the true positive rate vs false positive rate and precision-recall curve. The performance of the classification model (anomaly detection model) is measured within terms of cross-entropy loss, or log loss. Cross-entropy loss increases as the predicted probability deviates from the actual one. The cross-entropy loss and accuracy are shown in Fig. 11.

4.5 Trained NLP and feedback

Based on the anomaly classification, a trained chatbot is developed in line with the description of Section 3.3.3. The performance analysis of this chatbot is shown in Table 5. When the ML model classifies the data as normal, the chatbot returns “You are absolutely fine” and this message is also delivered to the user. If the ML model classifies the data as slightly abnormal then the chatbot returns “The health condition is not too worried but take medicines in time.” If the ML model classifies the data as moderately abnormal, the chatbot directly informs the admin about users’ health condition in detail and raises an SOS to alarm

Table 5 Performance analysis of the chatbot feature

Block size	Min. accuracy	Accuracy (Shubhchintak)	Min. F1-score	F1-score (Shubhchintak)	Min. success rate	Success rate (Shubhchintak)
16 × 1	0.8500	0.9843	0.9000	0.9617	0.9000	1
32 × 1	0.8500	0.9854	0.9000	0.9666	0.9000	0.9999
64 × 1	0.8500	0.9824	0.9000	0.9611	0.9000	0.9988
128 × 1	0.8500	0.9888	0.9000	0.9699	0.9000	1

to the neighbors. In extreme cases, the chatbot directly calls the doctor/home physician through the application.

Performance analysis of a chatbot is depicted in Fig. 12.

4.6 Comparison with state of the art approaches

We have carried out a survey of a large number of elderly people some of whom are patients of a local hospital (Jalpaiguri Super Speciality Hospital). From their experience, we can conclude that this application is very simple to use and the hardware is comfortable as well. From the doctors' point of view, the system has a very less response time and also provides data with the required accuracy. The proposed approach is compared with Al-Khafajiy et al. [2], Durán-Vega et al. [18], Paganelli et al. [40], Prabha et al. [43], and Kassem et al. [32]. The parameters of the comparison are as follows: (a) request response time, (b) the number of messages per latency, (c) scalability and communication cost, and (d) data storage cost in cloud infrastructure.

4.6.1 Request response time

In this application, request-response time is dependent on hardware as well as software components. Time to transfer fitness data from phones or connected sensors in wearable devices to cloud storage affects response time calculation and it is directly dependent on hardware components. Similarly, captured fitness data through different sensors need to be processed before sending it. Storing the data and its analysis using the ML model for decision making (i.e., anomaly detection) consumes maximum time. In our experiment, the accumulation and processing of health data are carried out by our proposed Shubhchintak system for 24

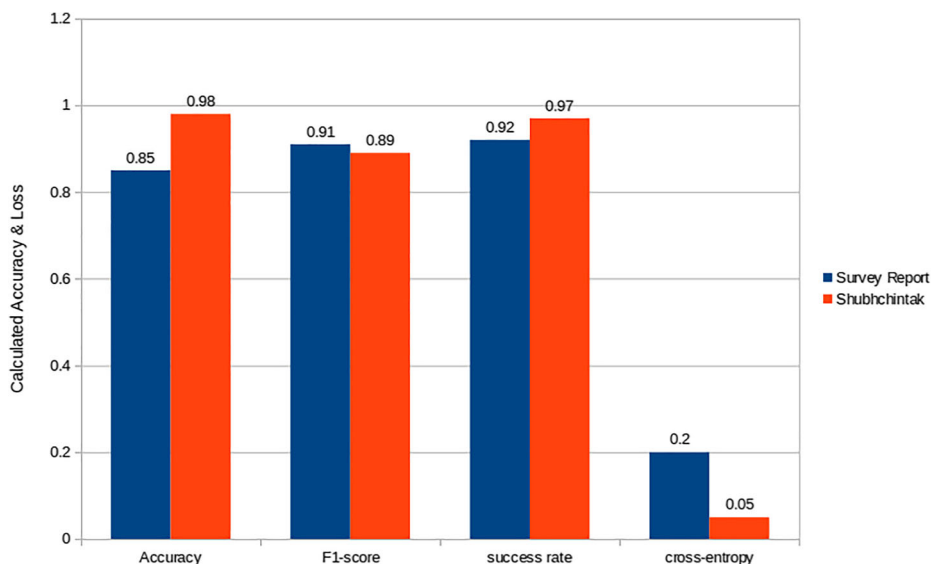


Fig. 12 Performance analysis of the chatbot

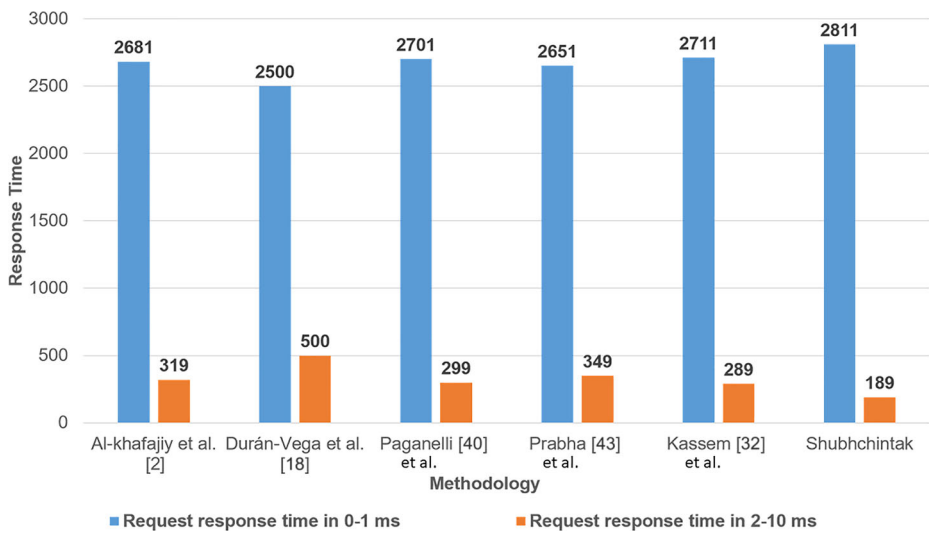


Fig. 13 Comparison analysis based on request-response time

hours. The overall system response time is computed for 3000 requests. Even, we compared the response time of our proposed system with the reported response times of some existing systems and the comparison is plotted in Fig. 13. It is observed from the comparison that the request-response time of the proposed application (i.e., Shubhchintak) is lesser than the existing applications [2, 18, 32, 40, 43]. Our proposed application Shubhchintak makes it possible by using optimized network protocol and virtual cloud machine. It comes with

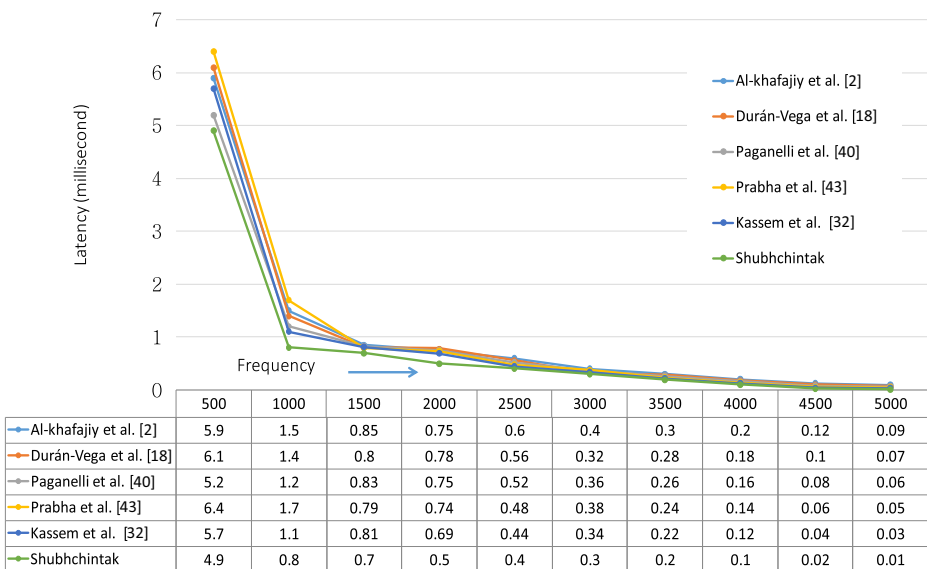


Fig. 14 Comparison analysis based on the number of messages per latency

some useful features such as: (a) multiuser visualization, (b) network simplicity, and (c) reduced data packet fault.

4.6.2 Number of messages per latency

Latency means the gap between time of issuing of an instruction for data transfer and the time when actual transfer of data begins. If the messaging protocol has too much latency, it reduces the fluency of the overall solution. We count the number of messages per latency of the proposed application and compare that with the same number of other existing similar applications. The comparison result is plotted in Fig. 14. In Fig. 14 the yellow line denotes the latency of the Shubhchintak application and it is always found to have a lower latency than the existing methodologies.

4.6.3 Scalability and pricing

Next, we check the scalability and price of our application. As we know, a proposed solution has to be scalable with various types of users. But we have to keep the pricing in the mind.

Table 6 Comparison of the state-of-the-art methods using Google Cloud

Approaches	Number of patients	Time needed to build the model (h)	Total cost (\$)	Cost per year (\$)
Al-Khafajiy et al. [2]	500	1:05	2.88	34.56
	1000	2:15	5.76	69.12
	1500	3:14	8.64	103.68
	2000	4:10	11.52	138.24
Duran-Vega et al. [18]	500	1:01	2.66	31.92
	1000	2:05	5.55	66.6
	1500	3:12	8.22	98.64
	2000	4:05	11.1	133.2
Paganelli et al. [40]	500	1:21	2.76	31.72
	1000	2:25	5.75	66.77
	1500	3:22	8.72	98.74
	2000	4:25	11.71	133.72
Prabha et al. [43]	500	1:31	2.86	31.52
	1000	2:35	5.85	66.47
	1500	3:32	8.82	98.64
	2000	4:35	11.81	133.32
Kassem et al. [32]	500	1:41	2.16	31.72
	1000	2:55	5.35	66.97
	1500	3:72	8.92	100.74
	2000	4:65	11.71	143.72
Shubhchintak	500	0:55	2.22	26.64
	1000	1:50	5.11	61.32
	1500	2:55	7.55	90.6
	2000	4:00	9.88	118.56

Data centre location = Iowa; Number of vCPU = 1; RAM Size = 8 GiB; EBS Bandwidth = Upto 10 Gbps; NVIDIA[®] Tesla[®] P4 GPU cost = \$0.60; 1 year commitment price = \$0.378 per hour

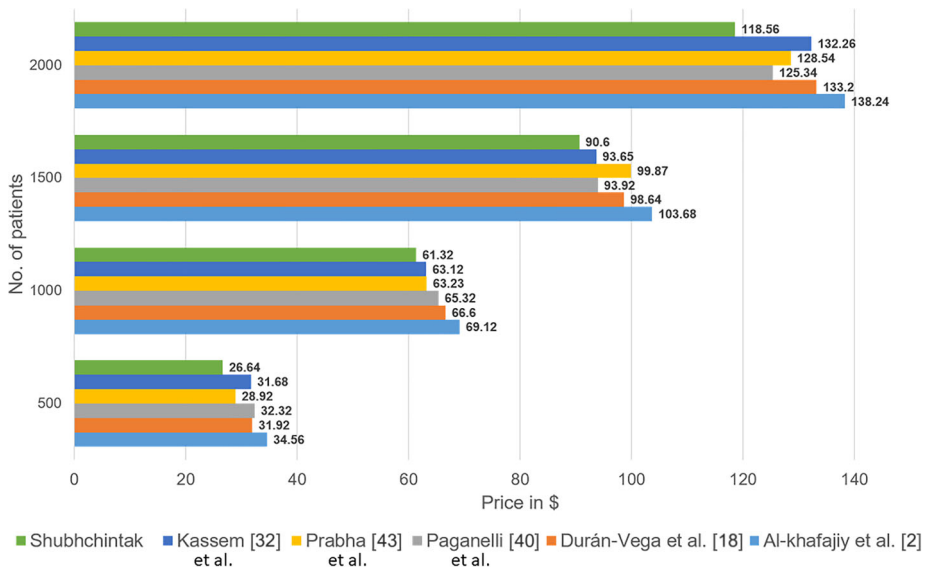


Fig. 15 Comparison of the scalability and pricing with state-of-the-art-approaches

As scalability increases, pricing also increases. People need a low-cost product with higher scalability. Shubhchintak meets such needs. Though we have used our own fitness band, this application is also compatible with other fitness bands like MI Band [57], HONOR band [28], Samsung band (<https://www.samsung.com/global/galaxy/galaxy-fit/>), Apple band [6] etc. And we use the Cloud only for non-prime users; this policy reduces cost for many users. We estimate the prices, based on 500, 1000, 1500, and 2000 connected patients and our estimated costs are compared with the prices for the same number of connected patients for existing prototypes as reported in [2, 18, 32, 40, 43]. As we are unable to know the configuration of the cloud of the other state of the art approaches, we prepare their model as suggested in [2] and [18] and run in the cloud services available to us. Table 6 and Fig. 15 help to understand the scalability and pricing of our implementation in google cloud configuration and provides a comparison with state-of-the-art approaches.

4.6.4 Data storage cost

Last but not the least, we need proper data storage to store the data for future uses. But the storage cost has to be kept in mind. The number of users is strictly increasing as well as the volume of data. But we cannot lose a single bit of data as health data is very sensitive. To achieve low-cost data storage, virtual storage provides a huge database with globalization and storage center. Here data are distributed to plurality nodes of multiple disks. This virtual storage also provides high-speed read-write service and as it reduces the hardware usage, it saves the hardware cost. In comparison with the existing methodologies, the cost reduction of the Shubhchintak application is shown in Fig. 16.

From Fig. 13, it is observed that Shubhchintak has very less response time than other two recently proposed methods. Also in Fig. 14, it is observed that Subhachintak has a lower latency as the yellow line converges much faster than the blue and orange line and from Fig. 15, we can conclude that the Shubhchintak application has higher scalability and

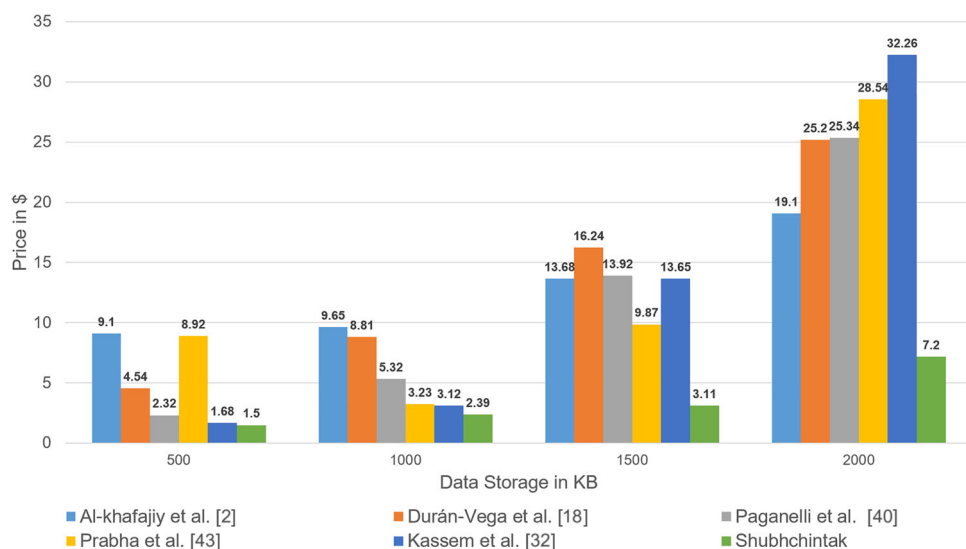


Fig. 16 Comparison of data storage cost

less cloud pricing than the existing methods [2, 18, 32, 40, 43]. Figure 16 states that the Shubhchintak application uses less storage than others. So, from these comparisons, we can clearly conclude that Shubhchintak is a cost effective, more accurate, and more efficient solution than other existing applications.

5 Conclusions

In this article, we have implemented a novel framework for remote health management of elderly people. The proposed system monitors their health parameters and provides real-time feedback according to their health condition. This solution is an IoT-based app that tracks the health conditions of the old age people in real-time and inform their guardians in case of any problems. It can also remind the patients about taking timely medicines and taking a walk in the evening as suggested by the doctor. The app also functions as an SOS alarm when old people faint anywhere, so as to call people nearby for rescue. If there is a minor problem, the app will itself provide a solution through the chatbot. The novelty is that the app can easily be combined with any fit band over Bluetooth and provide real-time analysis of the problems faced by the Elders. Our implementation is very easy to use and cost effective.

Moreover, our application, Shubhchintak has very less response time, better latency, higher scalability with less cloud pricing and it incurs less storage cost than recently reported applications. Even, our experimental results state that Shubhchintak gives a much more accurate and better performance than other existing applications.

The proposed application may be improved further by making it more energy efficient, secured towards medical data and enabling it to detect conditions like depression level which is very important in the times of covid-19 pandemic.

Declarations

Conflict of Interests The authors declare no conflict of interest in this work.

References

1. Agrawal S, Agrawal J (2015) Survey on anomaly detection using data mining techniques. *Procedia Computer Science* 60:708–713
2. Al-Khafajiy M, Baker T, Chalmers C et al (2019) Remote health monitoring of the elderly through wearable sensors. *Multimed Tools Appl* 78:24681–24706. <https://doi.org/10.1007/s11042-018-7134-7>
3. Andres J, Lott S, Qureshi K (2018) Eight-week outcomes of ledipasvir/sofosbuvir in noncirrhotic treatment-naïve patients with hepatitis c: analysis of pharmacy-based data. *Journal of Managed Care & Specialty Pharmacy* 24(1):23–28
4. American Diabetes Association (2019) 4. Comprehensive medical evaluation and assessment of comorbidities: standards of medical care in diabetes—2019. *Diabetes care* 42. Supplement 1:S34–S45
5. Apple Inc. (2016) Introduction to HealthKit
6. Apple Inc. (2020) Apple watch, available (online). <https://www.apple.com/watch/>. Accessed 26 April 2020
7. Apple Inc. (2016) HealthKit API Reference
8. Atoui H, Fayn J, Rubel P (2010) A novel neural network model for deriving standard 12-Lead ECGs from serial three-lead ECGs: application to self-care. *IEEE Trans Inf Technol Biomed* 14(3):883–890
9. Baig MM, GholamHosseini H, Connolly MJ (2015) Mobile healthcare applications: system design review, critical issues and challenges. *Australasian Physical & Engineering Sciences in Medicine* 38(1):23–38. Remote health monitoring system for detecting cardiac disorders
10. Bansal D, Khan M, Salhan AK (2009) A computer-based wireless system for online acquisition, monitoring, and digital processing of ECG waveforms. *Comput Biol Med* 39(4):361–367
11. Bansal A, Kumar S, Bajpai A, Tiwari VN, Nayak M, Venkatesan S, Narayanan R (2015) Remote health monitoring system for detecting cardiac disorders. *IET Syst Biol* 9(6):309–314
12. Bansal M, Chopra T, Biswas S (2021) Organ simulation and healthcare services: an application of IoT. In: 2021 6th international conference on inventive computation technologies (ICICT). IEEE, pp 205–208
13. Barnwell J, Klein J, Stallings C, Sturm A, Gillespie M, Fine J, Hyslop W (2012) Image-guided optimization of the ECG trace in cardiac MRI. *Int J Cardiovasc Imaging (formerly Cardiac Imaging)* 28(3):587–593. <https://doi.org/10.1007/s10554-011-9865-7>
14. Bergmann J, McGregor A (2011) Body-worn sensor design: what do patients and clinicians want? *Ann Biomed Eng* 39(9):2299–2312. <https://doi.org/10.1007/s10439-011-0339-9>
15. Bouchard B, Giroux S, Bouzouane A (2007) A keyhole plan recognition model for Alzheimer's patients: first results. *Appl Artif Intell* 21(7):623–658
16. Catarinucci L, De Donno D, Mainetti L, Palano L, Patrono L, Stefanizzi ML, Tarricone L (2015) An IoT-aware architecture for smart healthcare systems. *IEEE Internet Things J* 2(6):515–526
17. Clark M, Lim J, Tewolde G, Kwon J (2015) Affordable remote health monitoring system for the elderly using smart mobile devices. *Sensors & Transducers* 184(1):77
18. Durán-Vega LA, Santana-Mancilla PC, Buenrostro-Mariscal R, Contreras-Castillo J, Anido-Rifón LE, García-Ruiz MA, Montesinos-López OA, Estrada-González F (2019) An IoT system for remote health monitoring in elderly adults through a wearable device and mobile application. *Geriatrics (Basel)* 4(2):34. <https://doi.org/10.3390/geriatrics4020034>
19. Fitbit Inc., Fitbit SDK, available (online). <https://dev.fitbit.com/getting-started/>. Accessed 27 April 2020
20. Garbhapu VV, Gopalan S (2017) Iot based low cost single sensor node remote health monitoring system. *Procedia Computer Science* 113:408–415
21. Ghosh AM, Halder D, Hossain SA (2016) Remote health monitoring system through IoT. In: 2016 5th international conference on informatics, electronics and vision (ICIEV). IEEE, pp 921–926
22. Ghosh A, Raha A, Mukherjee A (2020) Energy-efficient IoT-health monitoring system using approximate computing. *Internet of Things* 9:100166
23. Google Inc. (2016) Fit SDK overview
24. Google Inc., Fit APIs, available (online). <https://developers.google.com/fit/android>. Accessed 26 April 2020

25. Hosseinzadeh M, Koohpayehzadeh J, Ghafour MY, Ahmed AM, Asghari P, Souri A, Pourasghari H, Rezapour A (2020) An elderly health monitoring system based on biological and behavioral indicators in internet of things. *J Ambient Intell Humaniz Comput*:1–11
26. <https://www.apple.com/in/watch/>. Accessed 26 April 2020
27. <https://developers.google.com/fit/overview>. Accessed 26 April 2020
28. Huawei Consumer Business Group, HONOR band, available (online). <https://www.hihonor.com/global/products/accessories/honorband5/>. Accessed 27 April 2020
29. Huifeng W, Kadry SN, Raj ED (2020) Continuous health monitoring of sportsperson using IoT devices based wearable technology. *Comput Commun* 160:588–595
30. Index, Cisco Visual Networking (2019) Forecast and trends, 2017–2022 white paper. Cisco: San Jose, CA, USA
31. Juyal S, Sharma S, Shukla AS (2021) Smart skin health monitoring using AI-enabled cloud-based IoT. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2021.01.074>
32. Kassem A, Tamazin M, Aly MH (2021) A context-aware IoT-based smart wearable health monitoring system. In: 2020 International conference on communications, signal processing, and their applications (ICCSPA). IEEE, pp 1–6
33. Keong HC, Yuce MR (2008) Low data rate ultra wideband ECG monitoring system. In: Dumont G, Galiana H (eds) 30th annual international conference of the IEEE engineering in medicine and biology society, Vancouver, BC, 20–25 Aug 2008. IEEE, USA, pp 3413–3416
34. Khoi NM, Saguna S, Mitra K, Åhlund C (2015) IReHMo: an efficient IoT-based remote health monitoring system for smart regions. In: 2015 17th international conference on e-health networking, application & services (HealthCom), pp 563–568. <https://doi.org/10.1109/HealthCom.2015.7454565>
35. Kole A et al (2019) Epidemiologic features of heart failure patients in Rural Central India. *Circulation* 140(Suppl_1):A15732–A15732
36. Kong D, Cen L, Jin H (2015) Autoreb: automatically understanding the review-to-behavior fidelity in android applications. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security
37. Maiolo C, Mohamed EI, Fiorani CM, De Lorenzo A (2003) Home telemonitoring for patients with severe respiratory illness: the Italian experience. *J Telemed Telecare* 9(2):67–71
38. Majumder S, Mondal T, Jamal Deen M (2017) Wearable sensors for remote health monitoring. *Sensors* 17(1):130
39. Nienhold D, Dornberger R, Korkut S (2016) Sensor-based tracking and big data processing of patient activities in ambient assisted living. In: 2016 IEEE international conference on healthcare informatics (ICHI). IEEE, pp 473–482
40. Paganelli AI, Velmovitsky PE, Miranda P, Branco A, Alencar P, Cowan D, Endler M, Morita PP (2021) A conceptual IoT-based early-warning architecture for remote monitoring of COVID-19 patients in wards and at home. *Internet of Things*:100399
41. Pandya S, Mistry M, Kotecha K, Sur A, Ghanchi A, Patadiya V, Limbachiya K, Shivam A (2021) Smart aging wellness sensor networks: a near real-time daily activity health monitoring, anomaly detection and alert system. In: Proceedings of second international conference on computing, communications, and cyber-security. Springer, Singapore, pp 3–21
42. Pardeshi V, Sagar S, Murmurwar S, Hage P (2017) Health monitoring systems using IoT and Raspberry Pi—a review. In: 2017 international conference on innovative mechanisms for industry applications (ICIMIA). IEEE, pp 134–137
43. Prabha D, Darshini B, Soundariya K (2021) IoT application for safety and health monitoring system for construction workers. In: 2021 5th international conference on trends in electronics and informatics (ICOEI). IEEE, pp 453–457
44. Rumelhart D, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323(6088):533–536
45. Saha J, Saha AK, Chatterjee A, Agrawal S, Saha A, Kar A, Saha HN (2018) Advanced IOT based combined remote health monitoring, home automation and alarm system. In: 2018 IEEE 8th annual computing and communication workshop and conference (CCWC). IEEE, pp 602–606
46. Saha R, Biswas S, Sarmah S, Karmakar S, Das P (2021) A working prototype using DS18B20 temperature sensor and arduino for health monitoring. *SN Computer Science* 2(1):1–21
47. Scalvini S, Zanelli E, Volterrani M, Martinelli G, Baratti D, Buscaya O, Baiardi P, Glisenti F, Giordano A (2004) A pilot study of nurse-led, home-based telecardiology for patients with chronic heart failure. *J Telemed Telecare* 10(2):113–117. <https://doi.org/10.1258/135763304773391576>
48. Srinivasulu A (2016) Measurement and wireless data transmission of heart rate using pulse sensor. *Indian Journal of Mednodent and Allied Sciences*:126–131

49. Taleb N et al (2016) Comparison of two continuous glucose monitoring systems, Dexcom G4 Platinum and Medtronic Paradigm Veo Enlite System, at rest and during exercise. *Diabetes Technology & Therapeutics* 18(9):561–567
50. Valliappan S, Mohan BPR, Kumar SR (2017) Design of low-cost, wearable remote health monitoring and alert system for elderly heart patients. In: 2017 International conference on iot and application (ICIOT), Nagapattinam, pp 1–7. <https://doi.org/10.1109/ICIOTA.2017.8073612>
51. Vedaiei SS, Fotovvat A, Mohebbian MR, Rahman GM, Wahid KA, Babyn P, Marateb HR, Mansourian M, Sami R (2020) COVID-SAFE: an IoT-based system for automated health monitoring and surveillance in post-pandemic life. *IEEE Access* 8:188538–188551
52. Verma P, Sood SK (2018) Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet of Things Journal* 5(3):1789–1796
53. Vitacca M, Assoni G, Pizzocaro P et al (2006) A pilot study of nurse-led, home monitoring for patients with chronic respiratory failure and with mechanical ventilation assistance. *J Telemed Telecare* 12(7):337–342
54. Wani RT (2019) Lifestyle medicine and use of technology in current healthcare. *BMJ Innovations* 5(4):135–135
55. Werbos PJ (1990) Backpropagation through time: what it does and how to do it. *Proc IEEE* 78(10):1550–1560
56. Wu T, Wu F, Qiu C, Redouté JM, Yuce MR (2020) A rigid-flex wearable health monitoring sensor patch for IoT-connected healthcare applications. *IEEE Internet of Things Journal* 7(8):6932–6945
57. Xiaomi Inc., Mi Band, available (online). <https://www.mi.com/in/miband2/>. Accessed 27 April 2020
58. Yao J, Warren S (2005) Applying the ISO/IEEE 11073 standards to wearable home health monitoring systems. *J Clin Monit Comput* 19(6):427–436
59. Yeri V, Shubhangi DC (2020) IoT based real time health monitoring. In: 2020 second international conference on inventive research in computing applications (ICIRCA). IEEE, pp 980–984
60. Zhong C-L, Li Y-L (2020) Internet of things sensors assisted physical activity recognition and health monitoring of college students. *Measurement* 107774:159. <https://doi.org/10.1016/j.measurement.2020.107774>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Affiliations

Ayan Banerjee¹ · Dibyendu Maji¹ · Rajdeep Datta¹ · Subhas Barman¹  · Debasis Samanta² · Samiran Chattopadhyay^{3,4}

Ayan Banerjee
ab2141@cse.jgec.ac.in

Dibyendu Maji
dibyendumaji415@gmail.com

Rajdeep Datta
rajdeep2898@gmail.com

Debasis Samanta
debasis.samanta.iitkgp@gmail.com

Samiran Chattopadhyay
samirancju@gmail.com

¹ Department of Computer Science & Engineering, Jalpaiguri Government Engineering College, Jalpaiguri 735102, India

² Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur 721302, India

³ Institute for Advancing Intelligence, TCG CREST, Salt Lake, Kolkata 700091, India

⁴ Department of Information Technology, Jadavpur University, Salt Lake City, Kolkata 700106, India