Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.

Internet of Things 16 (2021) 100459

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

CONFRONT: Cloud-fog-dew based monitoring framework for COVID-19 management

Anish Poonia [a], Shreya Ghosh [a,*], Akash Ghosh [b], Shubha Brata Nath [a], Soumya K. Ghosh [a], Rajkumar Buyya [c]

[a] Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India
[b] Department of Computer Science and Engineering, Jadavpur University, India
[c] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia

ARTICLE INFO

Keywords:
Internet of Things (IoT)
Cloud computing
Fog computing
Dew computing
Pandemic
COVID-19
Health monitoring

ABSTRACT

In the recent times, the IoT (Internet of Things) enabled devices and applications have seen a rapid growth in various sectors including healthcare. The ability of low-cost connected sensors to cover large areas makes it a potential tool in the fight against pandemics, like COVID-19. The COVID-19 has posed a formidable challenge for the developing countries, like India, which need to cater to large population base with limited health infrastructure. In this paper, we proposed a *Cloud-fog-dew based mOnitoriNg Framework foR cOvid-19 maNagemenT*, called CONFRONT. This cloud-fog-dew based healthcare model may help in preliminary diagnosis and also in monitoring patients while they are in quarantine facilities or home based treatments. The fog architecture ensures that the model is suited for real-time scenarios while keeping the bandwidth requirements low. To analyse large scale COVID-19 statistics data for extracting aggregate information of the disease spread, the cloud servers are leveraged due to its scalable computational and storage capabilities. The dew architecture ensures that the application is available at a limited scale even when cloud connectivity is lost, leading to a faster uptime for the application. A low cost wearable device consisting of heterogeneous sensors has also been designed and fabricated to realize the proposed framework.

1. Introduction

The COVID-19 pandemic has created an unprecedented challenge to the healthcare systems across the world. This is much more critical for the developing and under-developed countries, which often have very high population densities and limited healthcare infrastructure. Further, a large number of citizens are often needed to be quarantined or to be provided home-based treatments. Thus, there is need of a low-cost solution that may help in preliminary diagnosis/monitoring while the patients are in quarantine facility or in home isolation.

The *cloud computing* paradigm has risen to become a ubiquitous technology in the recent times. It is considered as a perfect solution to handle big data and provide resources on demand basis to the stakeholders. However, the amount of data, generated by the plethora of *Internet of Things (IoT)* sensors, especially for real-time applications has increased manifold. In such a case, it is not feasible to transmit such large amount of data all the way to the cloud as not only will it congest the network but will also result

* Corresponding author.
E-mail addresses: anishpoonia123@gmail.com (A. Poonia), shreya2015@iitkgp.ac.in, shreya.cst@gmail.com (S. Ghosh), akashkgp@gmail.com (A. Ghosh), nath.shubha@gmail.com (S.B. Nath), skg@cse.iitkgp.ac.in (S.K. Ghosh), rbuyya@unimelb.edu.au (R. Buyya).

https://doi.org/10.1016/j.iot.2021.100459
Received 1 March 2021; Received in revised form 21 May 2021; Accepted 12 September 2021
Available online 25 September 2021
2542-6605/© 2021 Elsevier B.V. All rights reserved.

in large amount of latency. For such applications, it is desired that the processing elements be brought closer to the sensors and actuators. This has been achieved by implementing a hierarchical architecture in which a *fog layer* has been introduced between the user devices and the cloud. The fog layer can be realized using network devices such as routers and servers which will perform part of the processing and storing tasks originally meant for the cloud layer.

In recent years, research in the area of fog technology in health sector has increased substantially. The role of automation to detect or predict diseases and ailments has opened new research avenues. The inherent ability of the ubiquitous sensors to collect data and transmit it to centralized locations has made them a potent tool in the fight against pandemic like situations. The COVID-19 crisis has brought out glaring lacunae in the health infrastructure to handle such a widespread pandemic, more so in developing and under-developed countries, where resources are often far and few. With exponential increase in the COVID-19 cases around the world, it is essential that home based remote health monitoring systems be used by health care professionals for monitoring the less critical patients. Such a system can also be used to regularly look for symptoms in individuals who have been quarantined on basis of their contact history and are susceptible to the infection in near future. Not only it will help in saving the precious time and effort, but also may ensure minimum physical contact between the suspect and health care workers and avoid possible virus transmission. Such an IoT based health monitoring framework can continue to be used in normal scenario in remote areas of the developing and third world countries, where people seldom go to health centres to get their medical check-ups done. Similarly, the elderly population of the society is often incapable to look after itself. In such scenarios, it may be prudent to provide citizens with a low-cost mechanism that can regularly collect information of vital body parameters and generate timely alerts.

In this paper, we have proposed a generic framework, namely CONFRONT, based on cloud-fog-dew architecture for monitoring subjects who are susceptible to COVID-19 like pandemic or other similar ailments. The framework collects data from wearable sensor device and alerts the user and health professionals on detection of any abnormality. The fog layer in the model reduces the latency and thus makes it suitable for real-time analysis which is a primary requirement for any health care applications. The proposed hierarchical infrastructure consists of three layers — IoT layer (consisting of sensors and actuators), fog layer and the cloud layer. The application for the framework has been designed in a modular fashion. The tasks can be added to, changed, or deleted from each module based on the service provider's application. The framework has a novel dew architecture which can be implemented using dew servers and dew databases. These dew components not only lead to better on-premise resource utilization but also improve the service uptime as the framework continues to be available to the user even if cloud connectivity is temporarily lost. This makes it even more suitable for use in remote and far-off places where seamless internet connectivity with the distant cloud data centres may not be available.

The major significance and the contributions of the work are as follows:

1. proposing a novel cloud-fog-dew architecture to remotely monitor health in case of pandemics, like COVID-19
2. fabricating a low-cost wearable device for remote monitoring of COVID-19 suspects without hampering their mobility
3. the proposed framework can significantly contribute in health care sector, especially of developing and under-developed countries, where the health infrastructure is often inadequate
4. the fog layer reduces latency and network usage by performing data processing at intermediate stages. The dew architecture improves the application uptime and resource usages
5. CONFRONT uses a Hidden Markov Model (HMM) architecture to effectively predict user's health status reducing the false-alarms

The rest of the paper is structured as follows. Section 2 discusses the existing research work in this field. Section 3 explains the proposed framework followed by its theoretical and simulation performance analysis in Sections 4 and 5. Finally, the conclusions are drawn in Section 6 along with the future scope of the work.

2. Related work

The term *fog computing* was introduced by Cisco[1] but was subsequently defined from several different perspectives. Fog Computing [1] is a paradigm that helps in achieving reduced latency in applications along with better mobility and scalability of heterogeneous sensors in order to achieve inter-operability. This is a promising paradigm to bring cloud applications closer to the IoT devices at the edge of the network [2]. In this work, the authors present *Fog Smart Gateway (FSG)* which is an intelligent gateway integrated with the capabilities of fog nodes. He et al. [3] proposed a private cloud six-layered architecture based on message queue at the cloud along with a plug-in algorithm so as to support concurrent requests from healthcare services. However, security issues are not dealt with and computing is centralized at cloud. Ghosh et al. [4] propose cloud-fog-edge based workflow management framework for emergency services such as healthcare.

Ahmad et al. [5] proposed a framework called Health Fog to be used as an intermediate layer between Cloud and end-users so as to reduce network usage. Cloud Access Security Broker has been used along with homomorphic encryption to provide security and privacy. Another work [6] presents how the inter-connectivity of IoT paradigm is beneficial for delivering healthcare services. Rahmani et al. [7,8] propose smart gateways at the edge of network to perform local storage, processing, etc. to exploit Fog Computing. This intermediate layer handled the load of the sensor and the distant cloud. A proof-of-concept design has been

[1] https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

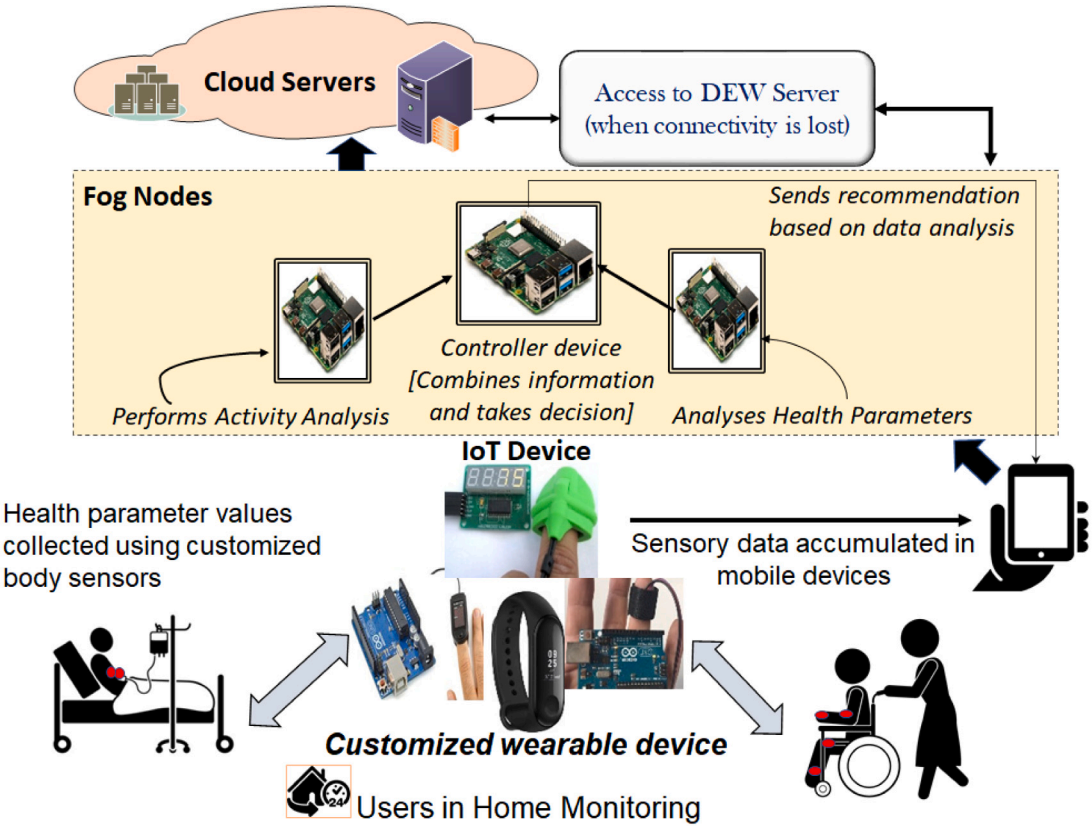


Fig. 1. Hierarchical structure of CONFRONT framework.

Table 1
Comparisons of existing works and CONFRONT framework for COVID-19 management.

Feature	Related works			CONFRONT
	[10,9,12]	[13]	[11]	
Cloud-fog-edge enabled framework	✓	✘	✘	✓
Dew computing	✘	✘	✘	✓
Health data analysis	✓	✓	✘	✓
Activity analysis	✘	✘	✓	✓
Design of wearable device	✘	✘	✘	✓

proposed to demonstrate the efficacy of the gateways. On the other side, [9,10] leverage cloud-fog paradigm for contact tracing, real-time drone based system and analysing the growth of the COVID-19 disease. Jean Louis et al. [11] presents a wearable activity tracker analysis in the context of home confinement during COVID-19. Another work [12] presents a novel spatio-temporal data analytics framework considering heterogeneous data sources, such as, mobility, travel statistics, population, literacy rate to effectively find out next hotspot zones and deploy zone-based lockdown measures.

Paul et al. [14] has proposed a context-sensitive healthcare system based on Fog layer which would weed out irrelevant data and thus reduce network usage, processing at the cloud and latency. Gill et al. [15] proposed another fog based information model that delivers healthcare service using IoT devices. Tuli et al. [13] proposed Health-Fog framework which used deep learning based techniques in the Edge devices in order to increase prediction accuracy and thus make the system useable in real-life situations. Yingwei Wang [16] formally defined dew computing as a potent technology to make independent and collaborative use of on-premise resources. A fog based Internet of Health things framework has been proposed in [17]. There are also works on analysing health status while user is in move [18] and home monitoring [19]. Yingwei Wang [20] suggest dew computing based architecture and potential methods of employing it as a global standard. The work [21] presents dew computing architecture in cyber–physical system and presents new features and functionalities of the architecture. The author claims that autonomy, independence and collaboration features of the proposed framework makes an edge over varied conventional paradigm such as edge computing systems, fog computing or cloudlet framework. Wang et al. [22] proposes a dew architecture based blockchain framework, named, *Dewblock* to enable enabled cryptocurrencies and enhance trust among several entities.

After the outbreak of COVID-19, many governments have come up with mobile based health platforms to effectively use user data for monitoring the health situation in the society. One such application is 'Aarogya Setu'[2] by the Government of India. Though the application is a great step to monitor the user's movement and contact history, however, it does not measure any health parameters. As a result, the application cannot monitor the health of the users. Our proposed model, which is based on a set of low-cost wearable sensors, collects vital body parameters and forwards it to fog and cloud devices to generate timely alert. Table 1 summarizes the features and existing works and *CONFRONT* to combat COVID-19. To the best of our knowledge, the proposed CONFRONT framework is unique and novel as none of the existing models or frameworks use dew architecture along with the cloud-fog hierarchy. This makes it more robust. At the same time, the detection algorithms and learning techniques can be replaced or modified any time by the service provider.

3. Proposed CONFRONT framework

This section presents the overview of the CONFRONT framework followed by the design and implementation process. Fig. 1 illustrates the hierarchical structure of the framework. At the bottom layer, the health parameters are collected using the customized wearable device which has several components to collect health parameters (blood pressure, heart-rate, oxygen level, body temperature) and movement parameters (accelerometer and GPS) at a given time-interval. These values are transmitted and accumulated in the users' mobile devices; which in turn transmitted to the fog layer. Then, the fog nodes perform the basic analysis on these sensory data in a distributed manner. For instance, as shown in Fig. 1, one fog node analyses the movement parameters and finds out the activity performed by the user in a particular time-interval. The other fog node simultaneously finds out whether the health parameters are within the normal range. Finally, the combiner node takes the decision about the health status of the users based on the data analytics, and sends the recommendation to the users' mobile devices accordingly. The cloud server performs any compute intensive task which is not feasible to run in the fog nodes due to the low computing power. Further, the cloud server stores the information about the users' health profiles. The computational and storage power of cloud servers is utilized for aggregate operations, such as, community health analysis and trend analysis of a region. The aggregate analysis performed in cloud servers provides recommendations of lockdown measures by extracting the risk of the infectious disease spread in a spatial region. CONFRONT framework also has a separate component named *Dew* server, which is utilized to cache the information in case the connectivity between the fog nodes and the cloud servers is lost. This dew server may be configured in the same physical device as the fog device or it can be placed on a separate system in close proximity (with fail-safe connectivity with the fog device). This will enable uninterrupted physical connectivity between the fog and dew server. The internet connectivity status at the fog layer will thus not impact fog-dew interaction. This ensures that the user can still connect/interact with the fog layer seamlessly in case the connectivity of fog-cloud is lost. It is assumed that the fog service provider will give dew services to the end users. The dew server will respond to the co-located fog device's users. The mobile devices send the user data to the static fog devices. The user is connected to the fog device by his/her mobile handset even when he/she moves around on their daily activities. It is evident that the CONFRONT framework is conducive to provision 24×7 home-health monitoring for ailing and elderly persons effectively.

3.1. Model design

A hierarchical structure and the application module implementing the proposed framework is shown in Fig. 2. The model proposed by Mahmud et al. [23] has been modified by adding an additional 'confirmatory module'.

Algorithm 1 : Working Model of CONFRONT

Input: Sensor data received from IoT devices
Output: Recommendation to users after processing the raw data

medical sensors embedded in the wearable device collect health parameters (blood pressure, heart-rate, oxygen level, body temperature) and movement sensors embedded in the wearable device collect movement parameters (GPS, accelerometer) in a defined time-interval
both health and movement variables are accumulated in the mobile-device
mobile device sends the data to the fog node
if fog node I is able to process all the health data **then**
    fog node processes the health parameters
    if all health parameters are within normal range **then**
        sends *OK* information to controller fog node
        controller fog node receives the information
        if connection with cloud can be established **then**
            sends $(OK, TimStamp)$ message to cloud server

[2] https://www.mygov.in/aarogya-setu-app/

```
11:         else
12:             discards the information
13:         end if
14:     else
15:         sends ALERT message to the controller node along with the abnormal health parameter value
16:         controller fog node receives the information
17:         stores the abnormal parameter value and timestamp
18:         go to step 44
19:     end if
20: else
21:     controller node checks the available resources
22:     if total available resources at fog layer ≤ computing resources required to analyse health information then
23:         controller node accumulates all information
24:         if connection with cloud can be established then
25:             sends the accumulated information along with user id to cloud server
26:             cloud server receives and analyses the information, and stores the result
27:             cloud server sends notification to controller fog node based on the analysis result
28:             go to step 44
29:         else
30:             sends the information in the Dew server
31:             Dew server stores the data
32:             controller fog nodes periodically checks the connectivity with the cloud servers
33:             if connectivity with cloud is restored then
34:                 go to step 26
35:             else
36:                 go to step 34
37:             end if
38:         end if
39:     else
40:         controller node distributes the analysis task among available fog nodes
41:         sends notification to the allotted nodes with the health data and analysis task
42:     end if
43: end if
44: Fog node II receives the movement parameters
45: activity detection module within fog node II identifies the activity user is performing at a given instance
46: sends the result activity, timeStamp to controller node
47: controller node analyses the results of health module (from fog node I) and activity detection module
48: if abnormality is detected then
49:     sends immediate notification to the user
50:     if connection with cloud can be established then
51:         sends the abnormal health status along with user id to cloud server
52:         cloud server stores the result and takes necessary action
53:     else
54:         Sends the information in the Dew server
55:         Dew server stores the data
56:         controller fog nodes periodically checks the connectivity with the cloud servers
57:         if connectivity with cloud is restored then
58:             go to step 51
59:         else
60:             go to step 56
61:         end if
62:     end if
63: end if
```

Each of these modules are placed on one of the hierarchy levels depending on the application. The red path represents flow of data. The client module is responsible for receiving the data from the wearable device which houses heterogeneous sensors such as blood pressure sensor, pulse sensor, pulse oximeter, body temperature sensor, accelerometer and GPS. The collected data is forwarded to the fog device where subsequent modules reside. The client module also receives the predicted result from event handler module and sends it to the user mobile for displaying. It may be noted that the client module is configured to always reside at the user mobile.

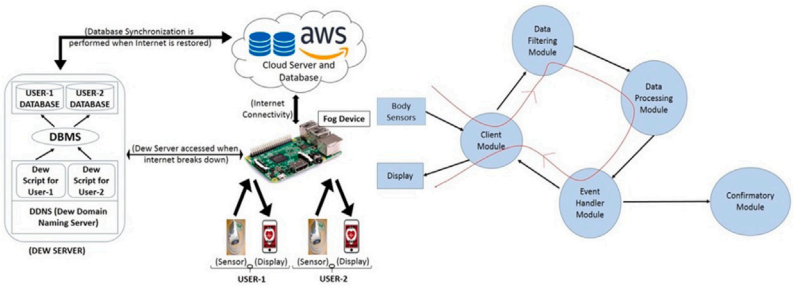


Fig. 2. Structure and application model implementing the proposed framework.

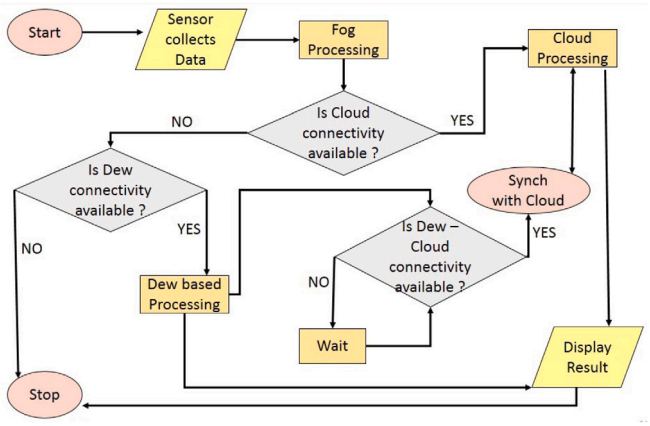


Fig. 3. Workflow of the CONFRONT framework.

Thus, the users must ensure that their mobile handsets should meet the minimum requirements to house this module. Data filtering module is responsible for noise filtering which then sends the filtered data to processing module. At the data processing module, features are extracted and data normalization is done and forwarded to event handler module. It is this module that implements the decision making algorithm and sends the result to be displayed to the user via the client module. Any positive case detected by the event handler module will be forwarded to the confirmatory module also, where human assisted decision may be made by a healthcare professional. If the manual diagnosis differs from the predicted result, an immediate correction can be forwarded to the user. This prediction error will be informed to the event handler module so that the prediction algorithm can learn and train itself continuously. The confirmatory module is always placed on the cloud. This has been done so that all the wrong predictions are known globally and can be used by all the fog nodes for training.

While the cloud-fog architecture ensures that the intermediate devices are used for processing tasks in order to bring computing closer to the edge, the architecture fails to access the cloud component if in case the internet connectivity goes down. The proposed framework can overcome this problem by employing the dew server. Fig. 2 and the workflow in Fig. 3 shows that if the internet connectivity between fog device and the cloud server snaps down, a connectivity with the dew server will be established.

To implement the dew server (which is co-located with the fog nodes through dedicated connection), the browser can be configured to redirect to the dew server address in case internet connectivity is lost. If a common dew architecture standard is accepted in future, the service provider can have their own dew-sites which may be mapped to the dew servers by the owner of the domain through the domain's registrar. Once the dew server is reached, the request is matched with the correct environment variable by the Dew DNS. This way the correct dew script is accessed. The user can then work on his/her data residing in the corresponding database using the installed database. The changes made to the database are retained in this dew server copy. The dew server database will be synchronized with the cloud database as soon as the internet connectivity between the dew server and cloud server is restored. Simultaneously, the fog device will also be now directly connected to the cloud server. At the back end, updates at the cloud server keep getting synchronized with the dew server. This will ensure that the fog device accesses the updated copy if the connectivity with the cloud breaks down again.

3.2. Activity analysis module to reduce false alarm in detecting abnormal health condition

In the context of COVID-19, the activity analysis of individual's is a crucial factor. The major objective of this work is to detect whether an user has any early symptoms of COVID-19 exposure. The commonly used screening tool for patients with COVID-19 is

the 6-minute walk test (6MWT).[3] This 6MWT is a measure of functional status or physical fitness. During this test, an user walks at her normal pace for six minutes. The heart rate and blood oxygen levels are measured before and after the test and these parameters are analysed. Specifically, these two parameters fluctuate if the user is infected. The test provides better understanding on probability of oxygen level decreasing, and will be beneficial for elderly and homebound users. Further, this test can be administered by a family member without any help from a paramedical/medical personnel. It may be noted that the activity analysis module of CONFRONT keeps track of the activity (walking for 6 min) of the user, and takes necessary action. For instance, information such as, start and stop alerts to the user, distance covered, heart rate and SPO2 change-rate — can be computed by CONFRONT.

The CONFRONT provides an excellent solution for this procedure. As discussed before, the embedded movement and medical sensors of our low-cost wearable device capture the sensory values of health parameters and movement features of an user. Now, the activity analysis module detects and assists user to take the 6MWT properly, and the medical sensors captures the blood pressure, heart-rate and oxygen saturation levels before, after and during the test at different time-intervals. Finally, the fog devices analyse the log and identifies if there is any risk present.

Another important feature of CONFRONT is that it is capable to detect health-status of users efficiently without less false-positives. Here, the *false-positive* of identification risk denotes that the module detect health abnormality of an user, however, the parameter changes are due to some environmental context, and the user is completely fine. This *false positive* results initiates unnecessary anxiety of the patients as well as overall hassle for the false-alarm to the health-caretakers. For instance, the heart rate of an user can be more than the normal range, when she works out in the gym, or the body temperature of an user may be more, when she is taking hot-bath. Therefore, these activity-contexts need to be identified in order to eliminate the false-alarms.

Environmental Context (EC): Environmental context (EC) is defined by an arraylist $EC^t = [ec_1^t, ec_2^t, \dots, ec_n^t]$, where each entry of the array-list contains a parameter of the surrounding region where the user is present at time t. Few examples of such parameters are *air temperature*, *humidity* and *light intensity*.

Activity (A): Activity of an user is represented by an ordered triplet (ap_i, TD_i, sT_i), where ap_i denotes the activity-name (such as, exercise, walking etc.) and the time-duration (TD_i) and start-time (sT_i) of the activity performed by the user.

Health Parameter (HP): Health parameter is represented by a pair $\langle (hs_1, hp_1^t), (hs_2, hp_2^t), \dots, (hs_n, hp_n^t) \rangle$, where hs denotes the health sensor/parameter (say, blood pressure) and the value of hs is presented by hp^t at time t.

Normal Range (NR): The normal ranges of the health parameter set (hp) is represented by $NR = \{nr_1, nr_2, \dots, nr_n\}$, where nr_i stores the normal range of health parameter hs_i. It may be noted that based on the age and other health-profiles of users, this NR value changes. Therefore, the fog nodes store different NR values for different users.

Next, we aim to identify the activities performed by the user in a given time-interval. We have categorized the activities in five broad classes, namely, *running*, *walking*, *standing*, *exercise* and *sitting*. The accelerometer is used to predict physical activity of the subject.

Accelerometer inputs

In our framework, the 3-axis accelerometer with high resolution (ADXL345) provides the X, Y and Z coordinates which is used to find the rate of change of position. This in turn is used to classify whether the subject is walking or running.

In the pre-processing step, the X, Y and Z dimensions are merged to give a single quantity denoted by 'a' using Euclidean distance formula as shown in Eq. (1). This will simplify the process of feature extraction.

$$\mathbf{a} = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

Noise filtering

The received signal is smoothed using a 5-point smoothing signal. Each signal is obtained as an average of four signals, two preceding signals and two succeeding signals. This will reduce some of the noise that may have got induced due to orientation shift of the sensor or accidental jerks in the subject activity.

Feature extraction

The following features were extracted from the filtered signal using a sample window size of 256 samples and an overlap of 50%:

(a) Maximum Amplitude
(b) Minimum Amplitude
(c) Mean Amplitude
(d) Standard Deviation in Amplitude
(e) Energy in Time Domain
(f) Energy in Frequency Domain

[3] https://www.physio-pedia.com/Six_Minute_Walk_Test_/_6_Minute_Walk_Test

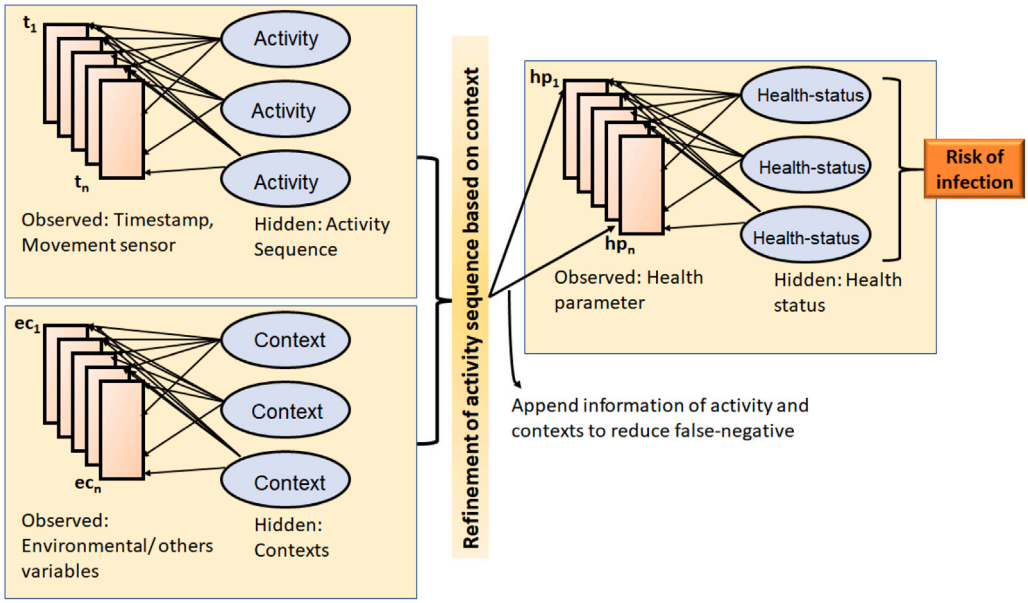


Fig. 4. Health status prediction of CONFRONT framework.

Feature normalization

The values obtained for the features were then normalized in the range [0,1]. The formula for normalization is as given in Eq. (2):

$$\mathbf{y} = \frac{x - \min}{\max - \min} \tag{2}$$

The data so obtained is used for training the KNN classifier, The testing data is also normalized in similar way and would then be used for prediction by the classifier.

KNN classifier

K Nearest Neighbour (KNN) [24] classifier has been used for classifying the activities as one of the five classes. The value of $k = 3$ was found to give the best result. This best result was obtained using GridSearchCV[4] when 5 fold cross validation was performed on the normalized training set. The nearest distance was calculated on based of Euclidean distance.

The activity classification coupled with data collected by all other sensors helps in predicting if the patient can be classified as a potential or expected positive case of COVID-19 or pandemic of similar nature. For example, a patient with oxygen levels less than 90%, high body temperature and abnormal pulse readings will be classified as positive case of COVID-19 by the used disease prediction algorithm. However, if the accelerometer indicates a strenuous activity (i.e. running) in hot/humid ambient temperature, the predicted result may change as low oxygen levels, high body temperature and pulse readings may now be attributed to the physical activity. It may be noted that the disease prediction algorithm employed is for research demonstration purpose only. The positive case results are notified back to respective users and healthcare authorities by the fog node so that confirmatory tests can be initiated at the health care centre for such individuals and adequate preventive measures are taken at the earliest. If the confirmatory test results are negative, the same can be informed to the prediction algorithm so that it can learn better and improve its prediction accuracy.

3.3. Health data analysis

In this section, we discuss about the health parameter analysis module (HAM). Here, the inputs of the module are health parameter (HP), activity (AP) and environmental context (EC) at different timestamps. The objective of the module is to identify whether the person is at risk of infection or any other abnormal health conditions.

The method to identify an users' health status is not straight forward, as it depends on several factors. Firstly, the health parameter values depend on the age, gender and pre-existing disease of the user. Also, the environmental contexts plays an important role to

[4] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

understand whether the user is actually at risk or not. In this regard, we model the problem of identifying health-problems of users based on the accumulated data using *Hidden Markov Model*. In brief, Hidden Markov Model is a statistical Markov model in which it is assumed that the system is modelled using Markov process. Markov process assumes that future predictions are dependent on most recent observations. Based on the changes of health parameters along with other variables over time, HMM can detect whether a person is having any abnormal health condition efficiently. CONFRONT uses the hidden Markov model to predict a individual's health status due to its ability to consider various influencing factors as an unobserved parameter.

We aim to model user's activity, environment variables and collected health parameter values to predict the health status of the user. Typically HMM consists of two kinds of stochastic variables, *state variables (hidden)* and *observable variables*. Our architecture to identify user's health status is illustrated in Fig. 4. It represents the architecture of the HMM-based health status prediction module where each nodes represent a random variable at a given time (t). In the left side, two separate layers are present. t_i represents the observed variable (timestamp and movement sensor), and the hidden state is activity sequences of the user. Specifically, this layer constructs the basic activity sequences (standing, walking, sitting etc.) from movement sensory information. The left bottom layer extracts the context (hidden state) from observed environment variables (air temperature, humidity, low light/sound intensity etc.). This layer extract the hidden context of the activity sequences. Next, the basic activity sequences are refined based on the extracted contexts. For instance, heart rate may increase while exercising, or heart rate may decrease while sleeping etc. The body temperature of an user may increase/decrease while having bath. We also append user's pre-existing disease/medical history in the second layer of the model. This is beneficial for identifying the health-status of users more efficiently.

Given a sequence of observations, the health-status prediction task (i) associates a set of parameters with each observed contexts; (ii) detects abnormal (or sporadic health parameter values) given the contexts and (iii) learns the parameters of the HMM module in several layers based on the user's dataset.

Hidden States: These are defined by the health-status of the user. For example, *(high blood pressure, t_1–t_6), (low oxygen level, t_{10}–t_{20})* are two hidden states of an user's medical record.

Observable States: The sensory information, such as, accelerometer sensors, activities performed by the user and health parameter values are considered as *observed* variables. All of these variables can be easily accumulated from our low-cost customized wearable device.

The proposed model or *HAM* (health parameter analysis module) is formally defined as $\Theta = \{\langle H, \kappa\rangle, \langle O, \kappa\rangle, \chi, \kappa\}$, where $\langle H, \kappa\rangle$ represents the set of hidden variables of HAM. The layers are represented by κ. $\langle O, \kappa\rangle$ denotes the observed variables obtained from various sensory information. The state transition probabilities and observation probabilities at different layers of *HAM* are denoted by χ.

Next we extract several inferences from the HAM of CONFRONT about the users' health status. For *evaluation* or computation of the likelihood of an observed sequence ($P(O|\Theta)$), HAM utilizes *forward-algorithm along with a k-order Markovian assumption*. Inspired from the work of Ghosh et al. [25], we use a variable a^k to extract k-length sequence of observed variables. It helps to model HAM from historical sequence of k-length, which is beneficial for monitoring the health status of an user during a medical test (say, 6MWT). Thus, the observation probability can be represented from *forward algorithm* as:

$$P(a^k|\Theta) = \sum_{t=1}^{length_{max}} P(a^k|h_t^k) * P(h_t^k) \tag{3}$$

where, the maximum length of hidden states are $length_{max}$, h_t^k denotes sequences of hidden states within k-length. It may be noted, that we have used the idea of *k-order* Markov chain to predict the output depending on k recent sequences. After aggregation, we get:

$$\begin{aligned} P(a^k|\Theta) = \sum_{t=1}^{length_{max}} [\prod_{i=1}^{k} P(a(i)|h_t(i)) \\ * P(h_t(i)|h_t(i-1), h_t(i-2), \ldots, 1)] \end{aligned} \tag{4}$$

The *decoding problem* finds out the most likely sequence of hidden states. It is evident that each observation is related to a sequence of hidden states in different layers of HAM. We have used a variant of *Viterbi algorithm* using time-relationships among the possible sequences. HAM maximizes $p(h_{t:T,\kappa}|\Theta, o_{t:T,\kappa})$ over all possible hidden state sequences. Finally, in parameter learning, an iterative version of *forward backward* algorithm is utilized. Algorithm 2 briefs the steps of the health data analysis module (HAM) of CONFRONT. Fig. 5 illustrates the sequence diagram of the underlying sequence of the processes of the algorithm. The health parameter values are accumulated after a specific time-interval (t) and sent to the mobile device (m). Similarly, environment parameters are also sent to mobile device from the smart-home sensors (es). The smart phone extracts context information such as movement of the user, acceleration, light intensity etc. The information are processed in the worker fog nodes (wfn) based on health data, environmental condition and activity data. The controller fog node (cfn) performs the aggregate analysis and learns the parameter of the model to identify abnormal health condition. In case abnormal condition is detected, alert is send to the cloud for further intimation to the health-care centres and the caregivers of the user. Subsequently, notification is sent through mobile device as well to the user.

Algorithm 2 Health Data Analysis for Recommendation

Input: Health parameters ($HP = \{hp_1, hp_2, \ldots, hp_n\}$) and Sensor values for activity sensing ($AS = \{ap_1, ap_2, \ldots, ap_m\}$)
Output: Recommendation regarding Health Profile

```
function GenerateHAM(u)                                   ▷ Where HAM - Health analysis module, u - user id
    HAM(H[ ], O[ ]) ← NULL; AP ← NULL; EC ← NULL; HP ← NULL   ▷ Initialization of the variables
    for l = 1 to κ do
        for i = 1 to T do
            Accumulate_data(EC_i ← (data, time))
            Accumulate_data(HP_i ← (data, time))
            Accumulate_data(AP_i ← (data, time))
        end for
        Create_node(O_l) ← accumulated_data(EC, AP)
        Generate_Model(Θ) ← O_l
        χ ← parameterLearning(Θ)                          ▷ Model update
        H ← activitySequenceGen(Θ, χ)
    end for
    for j = 1 to m do
        for i = 1 to m do
            flag ← checkRange(HP_i, NR_j)                 ▷ Check the normal range of the parameters
            if flag == 0 then                             ▷ Health parameter values NOT within normal range
                p ← analyseActivity(HP_i, H, t)
                count=0
                while p! = 1 and count ≤ 5 do
                    TEMP ← Extract all sensor data after (t + ζ) time − interval
                    p ← RepeatAnalysis(TEMP)
                    count = count + 1
                end while
            end if
            if flag == 0 or p == 0 then
                Alert(Health Status Critical)
            else
                Alert(Health Status Normal)
            end if
        end for
    end for
end function
```

3.4. Design and implementation

The proposed model has been implemented using actual sensors. A low-cost wearable device (see Fig. 6) has been fabricated by removing the LCD panel of the blood-pressure and pulse sensor. The created space has been used to accommodate other sensors such as blood-pressure sensor, pulse meter, accelerometer (ADXL-345), NZ-GPS and wifi module in the plastic body case itself and utilize it as a single wearable sensor.

The device has been further customized to output serial data at 9600 baud rate (8 bits data, No parity, 1 stop bits) in ASCII format. In addition, body temperature sensor and pulse oximeter sensor can provide critical additional information. This information is critical as pandemics like COVID-19 often show symptoms such as fever [26] and low blood oxygen levels [27] which may result in shortness in breath and rapid pulse rate.

ADXL345 has been used as an accelerometer with an aim to infer the amount of physical activity being undertaken by the subject wearing the sensor. This would help to ascertain if abnormality in the collected values is attributed to increased physical activity of the subject, low oxygen level or some other unknown reasons.

The NZ-mini GPS module is fitted so that the subject's geographical location is continuously monitored. This will not only help in understanding endemic factors while making inferences from the collected data but can be used to enforce strict quarantine on suspected COVID-19 or similar pandemic patients. The collected data can also be used to trace the movement history of such patients who have been under monitoring using our proposed wearable device. In addition, containment areas or zones can be identified where the number of COVID-19 cases are high.

All these sensors have been integrated, programmed and burnt onto a single chip which has been integrated with the blood pressure and pulse sensors in the same wearable body case. Employment of ECG sensors is desirable in early detection of COVID-19. However, it would need medical expertise for data collection, severely reduce the mobility and comfort level of an asymptomatic user apart from significantly increasing the wearable device cost. Thus, ECG sensor has not been integrated in the present design. However, it is technically feasible to integrate the ECG sensor. This wearable case is then connected using ESP8266 wifi chip 12f

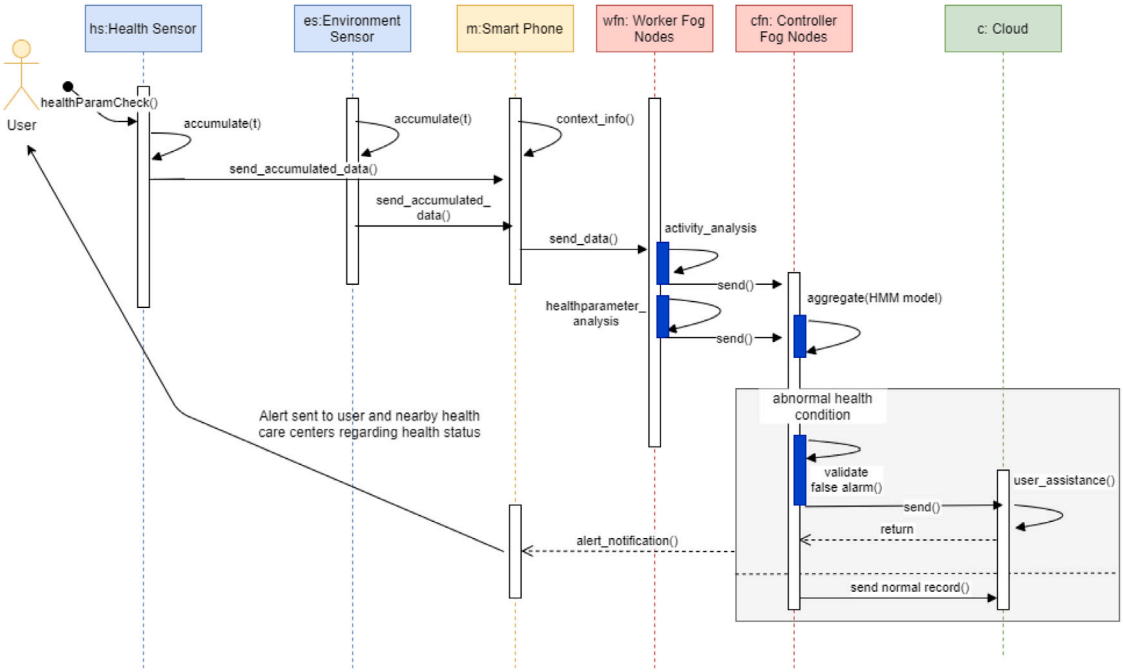


Fig. 5. CONFRONT framework: sequence diagram.

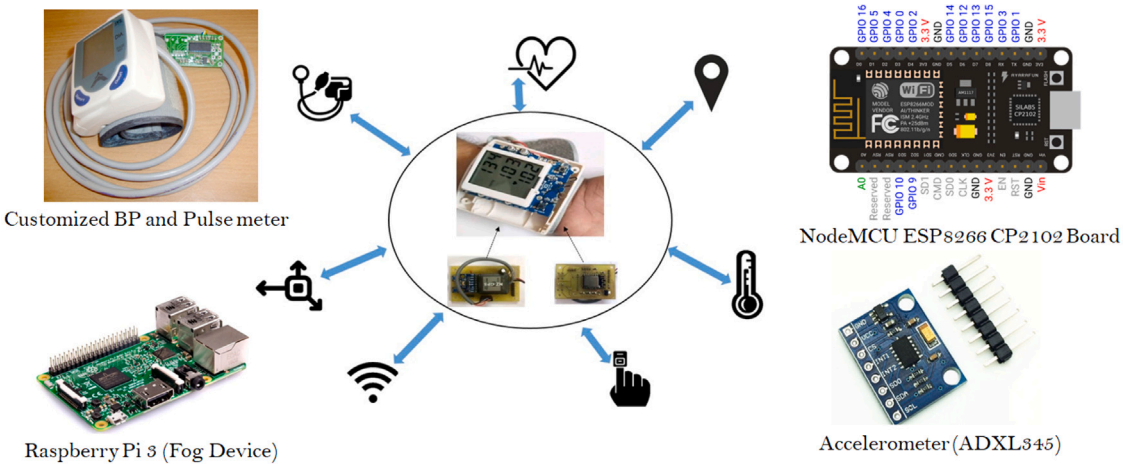


Fig. 6. Low-cost wearable sensor (designed and fabricated in the lab of IIT Kharagpur for this research work).

to the RPi which is further connected to the cloud. All the collected data is sent at continuously from the wearable device to the user mobile handset using ESP8266 wifi chip 12f and MQTT protocol. The data is then forwarded over wifi network by the client module on the user mobile handset to a Raspberry Pi 3 which acts as fog node where all the modules are placed except confirmatory module which as mentioned previously will always remain on cloud. In a real world scenario, the user mobile handset can easily connect to internet using the existing network of Base Transceiver Stations and then forward it to designated fog nodes. The activity detection algorithm [28] as well as the disease prediction algorithm are also implemented in the fog device. We have also deployed activity-modelling using activity-time graph [29] for profiling users based on their habitual preferences at different time-scales.

4. Power and energy consumption of CONFRONT: Theoretical model

In case of continuous health monitoring, the health, activity and movement data are collected using the wearable device, and the mobile device sends the data to fog nodes. The data processing is performed inside these fog devices before forwarding to the cloud. Inside the fog nodes, the data-instances are processed and then the predicted recommendations are forwarded to the mobile device.

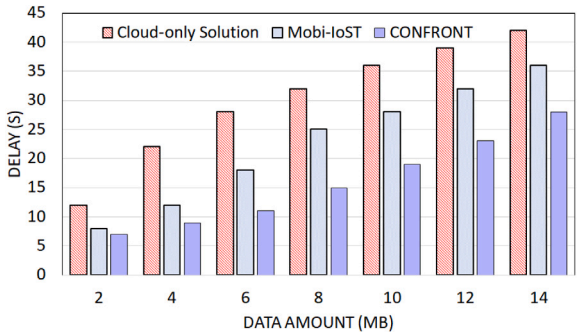


Fig. 7. Comparison of latency in receiving recommendations from fog nodes and cloud.

In this regard it has to be noted that the fog nodes have limited resources and storage capabilities. The recommendations are sent to the user based on his/her preferences. As the fog nodes store the recommendations itself, the user can receive them at lower latency than the conventional paradigm where the recommendations are sent to the user from the cloud. Let the recommendation data amount is $Data_{rec}$, data transmission rate from cloud to fog and from fog to mobile device are R_{cs} and R_{sm} respectively, and the link failure rate from cloud to fog and from fog to mobile device are f_{u1} and f_{u2} respectively. The latency in receiving recommendation from cloud L_{cm} is given by,

$$L_{cm} = (Data_{rec}/R_{cs})(1 + f_{u1}) + (Data_{rec}/R_{sm})(1 + f_{u2}) \tag{5}$$

The latency in receiving recommendation from fog node L_{sm} is given by,

$$L_{sm} = (Data_{rec}/R_{sm})(1 + f_{u2}) \tag{6}$$

Comparing Eqs. (5) and (6) it is observed that $L_{sm} < L_{cm}$. In Fig. 7 the latency in receiving recommendations by the users from cloud and fog nodes are shown. It may be observed from the theoretical analysis that the latency in receiving recommendations from fog nodes as in the proposed framework is $\sim 45\%$ lower than receiving recommendations from cloud as in the conventional scheme. In this regard it may be noted that in link failure cases, dew servers cache the information and processes the data. Therefore, the users will receive the result faster than receiving them from cloud (when connectivity issue arises). Let us consider the number of IoT devices sending data to the mobile device is N_{iot}, the number of data packets received by the mobile device from an IoT device D_j is P_j, where $1 \le j \le N_{iot}$, the power consumption of a mobile device in receiving a data packet p is P_{rp}, the power consumption in transmitting a data packet p is P_{tp}, the power consumption for accumulating data packets is P_a, the power consumption for analysing data packets is P_{en}, N_{tmen} is the number of data packets transmitted by a mobile device to the fog device.

The total number of data packets N_{rm} received by the mobile device from N_{iot} number of IoT devices is given as,

$$N_{rm} = \sum_{j=1}^{N_{iot}} P_j \tag{7}$$

The power consumption of a mobile device P_{rm} in receiving N_{rm} data packets is given as,

$$P_{rm} = \sum_{p=1}^{N_{rm}} P_{rp} \tag{8}$$

The power consumption of a mobile device P_{tmen} in transmitting N_{tmen} data packets is given as,

$$P_{tmen} = \sum_{p=1}^{N_{tmen}} P_{tp} \tag{9}$$

Therefore, if dew servers are not present, the total power consumption of the mobile device for data transmission, reception, accumulation and encoding is given by,

$$P_{totmen} = P_{rm} + P_{tmen} + P_a + P_{en} \tag{10}$$

The power consumption of a mobile device in transmitting N_{tm} data packets is given as,

$$P_{tm} = \sum_{p=1}^{N_{tm}} P_{tp} \tag{11}$$

If dew server is present, the total power consumption of the mobile device for data transmission, reception and accumulation is given by,

$$P_{totm} = P_{rm} + P_{tm} + P_a \tag{12}$$

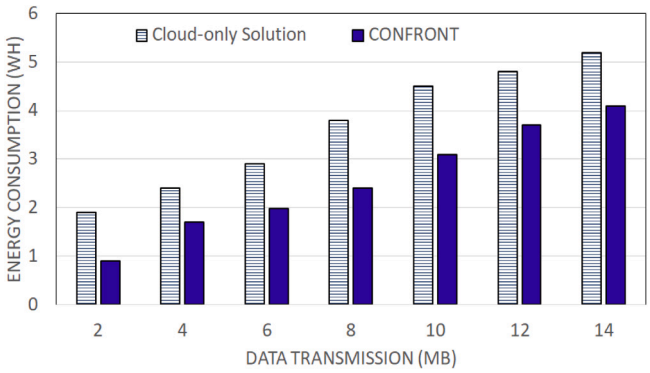
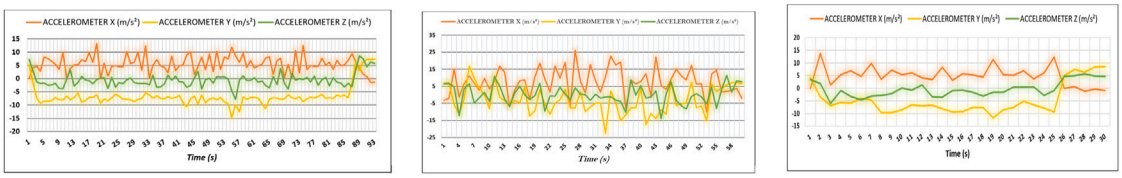


Fig. 8. Comparison of energy consumption of CONFRONT.



(a) Walking (b) Running (c) Climbing Upstairs

Fig. 9. Snapshot of accelerometer profile of three activities.

Table 2
Comparison of accuracy measure for identifying activity and health condition.

Activity	Bayesian model	KNN	DT	SVM	NN	CONFRONT
Walk	68.8%	77.3%	71.05%	81.52%	73.09%	**86.02%**
Run	73.2%	82.08%	74.6%	84.03%	79.63%	**87.80%**
Climb upstairs	71.04%	78.56%	70.48%	81.2%	75.28%	**85.71%**
Climb downstairs	75.46%	85.2%	72.15%	84.8%	77.08%	**89.11%**
Health condition						
Normal health and high external temperature	65.12%	67.8%	58.19%	71.72%	72.32%	**96.11%**
Normal health and strenuous activity	61.06%	68.21%	52.62%	73.81%	74.72%	**95.04%**
Abnormal health status	59.08%	62.16%	80.61%	77.50%	79.61%	**95.83%**

In Fig. 8 the power consumption of CONFRONT is presented and compared with the cloud-only solution. This is observed that use of dew server reduces the power consumption by ∼ 35%. It also outperforms the baselines in terms of latency as well. Therefore, the proposed framework, *CONFRONT provides a delay-aware and energy-aware health recommendations to users.*

5. Performance evaluation

In this section, we present the performance evaluations of the *CONFRONT* framework to demonstrate the efficacy of the system. Specifically, we evaluate the framework with few real data-instances collected during the study. Further, the scalability of *CONFRONT* is evaluated using a simulation study in iFogSim toolkit.

5.1. Experimental results

During this study, we collect user's activity, health and other contextual datasets from the Kharagpur (22.31454, 87.306) and Kolkata (22.5379, 88.3682) region of India. The collected dataset includes all parameters captured by our designed wearable sensor at different time-intervals for a duration of 14 days. Specifically, we have collected health data voluntarily from a small set (40) of students including both under graduate and graduate students (including Ph.D. scholars and research staffs of the laboratory) and professors. Amongst them, the age and sex distribution are as follows: 8 (age range: 17–21), 10 (age range: 22–30) and 14 (age range: 31–40), 8 (age range: 41+), 25 (Male), 15 (Female). Among the volunteers, twelve people reported pre-hypertension in their health profiles. Different health parameter (body temperature, blood pressure, pulse rate, SPO2 etc.) values and the normal range are logged in a dictionary object based on the age and sex of people. The normal range is identified by the help of the medical

Table 3
Placement for application modules.

Application module	Placement in CONFRONT	Placement in cloud-only solution
Client module	Mobile	Mobile
Data filtering module	Area gateway	Cloud
Data processing module	Area gateway	Cloud
Event handler module	Cloud	Cloud
Confirmatory module	Cloud	Cloud

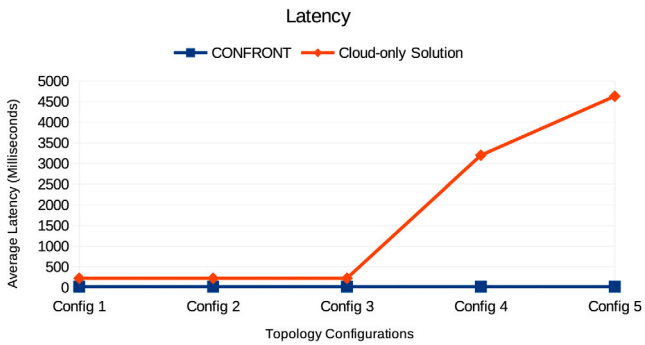


Fig. 10. Model performance during simulation: latency.

practitioner. In a given condition, say the user is doing normal activities (walking, sitting, sleeping etc.), if all the health parameters sensed by the wearable device fall into the defined range, then the user is considered having "normal health condition". Here, "Strenuous activity" represents running, physical exercise, swimming, cycling, - in other words, when our health parameter values change due to the activities performed. "High external temperature" represents the ambient temperature is more than the normal weather condition due to some external stimulus (hot water), and that may too effect the body temperature of the user. Finally, the "abnormal health status" represents that without any external impact the health parameter values are outside the range of the normal values. The dataset is captured at varied environment conditions as well to validate our proposed methodology. Fig. 9 shows a snapshot of an user's accelerometer profiles while the user is engaged in different activities, such as, walking (see Fig. 9(a)), running (see Fig. 9(b)) and climbing upstairs (see Fig. 9(c)).

To evaluate the efficiency of the proposed framework, we set-up different conditions and measure the accuracy of identifying activity and actual health condition. We compare *CONFRONT* with five baselines and report the *accuracy* for all of these conditions. Table 2 presents the accuracy measure of CONFRONT along with the baseline methods namely, *Bayesian Model*, *KNN*, *Decision Tree (DT)*, *SVM* and *NN*. The parameter for KNN is selected as 3. We have chosen *radial basis function (RF)* as the activation function in NN. A linear kernel is selected for SVM. The results for different runs are captured and average accuracy measure is reported. In the experiment, we evaluate the accuracy of basic activities, as well, we also capture the accuracy of the methods in different contexts, such as, *when user is performing any strenuous activity*, or, *when the ambient temperature is high*. It is observed that *CONFRONT* performs exceptionally well in identifying health conditions of users compared to the baseline methods. It has outperformed other methods in a significant margin of $\approx 24.8\%$. The key reason of this observation is that the refinement layer of the proposed *HAM* helps in removing the false-positives and identifies the health status of the users efficiently.

5.2. Simulation results

The proposed framework's performance has been evaluated using iFogSim [30] for five different configurations in a purely cloud based architecture as well as cloud-fog architecture on the basis of three parameter viz. average latency, network usage, and cost of execution at the cloud. The five configurations have one, two, four, eight, and sixteen area gateways (acting as fog nodes) respectively with four user mobiles connected to each area gateway. The modules were placed as mentioned in Table 3. This simulation work has been carried out in order to test the correctness and efficacy of the suggested hierarchy and the application model before investing in actual hardware.

5.2.1. Average latency

The latency of the *CONFRONT* framework and cloud-only solution are illustrated in Fig. 10. Average latency is an indicator of how well the application will respond in real time. The lesser the latency, better is real time response of the application. The delay for cloud-fog model remains low as the application modules get placed at respective area gateways. Thus for every end user, the response is coming from its corresponding parent area gateway. Though the confirmatory module exists at the cloud in all the configurations, however, it contributes to the delay only when a positive case is classified by the event handler module (see Fig. 2).

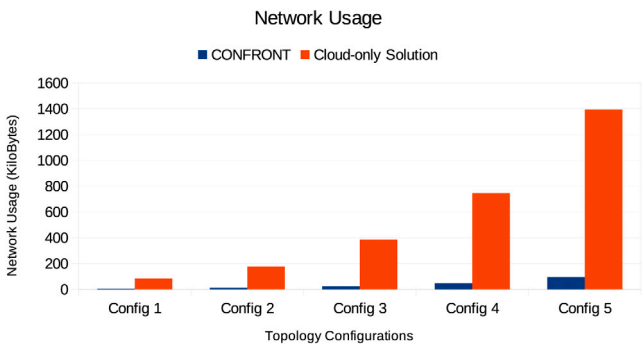


Fig. 11. Model performance during simulation: network usage.

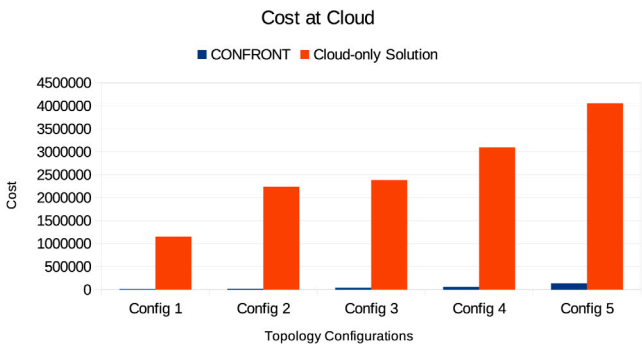


Fig. 12. Model performance during simulation: cost at cloud.

Table A.4
List of acronyms used in the paper.

Acronym	Full form
COVID-19	Coronavirus Disease 2019
BAN	Body Area Network
BP	Blood Pressure
AWS	Amazon Web Services
ECG	Electro Cardio Gram
GPS	Global Positioning System
IoHT	Internet of Health Things
IoT	Internet of Things
QoS	Quality of Service
SPO2	Saturation of Peripheral Oxygen
6MWT	6-minute walk test (6MWT)
DNS	Domain Name System
MQTT	Message Queuing Telemetry Transport
LCD	Liquid Crystal Display
ADM	Activity Detection Module
HAM	Health parameter Analysis Module

5.2.2. Network usage

We show the network overhead in Fig. 11. The stark difference in network usage values of cloud-fog based placement and cloud only placement is owing to the location of application modules in both the frameworks. The fog based placement policy has only the event handler module and the confirmatory module placed at the cloud. However, in case of cloud based model, all the critical modules are placed at the cloud (Table 3), which results in all the data being pushed to the cloud and thus increasing the network usage.

5.2.3. Cost of execution

The incurred cost value in case of fog based model is less as cloud resources are used only when the confirmatory module is accessed which happens in case of positive detection only. All the other application modules are placed either on the mobile device

or the intermediate fog devices. However in case of purely cloud based model, all the modules except client module are placed on the cloud itself. This results in large processing requirement from the cloud and thus increasing the cost phenomenally. It can thus be clearly seen that cloud-fog architecture has outperformed the pure cloud architecture on all three parameters in simulator based performance evaluation. The cost of execution of *CONFRONT* framework and cloud-only solution are shown in Fig. 12.

5.3. Discussions

Dataset collection and experimentation have been performed under several restrictions due to COVID-19. To make the system more reliable, we have extensively tested the modules based on both real-life data (collected through volunteers) as well as simulated traces generated from MATSIM[5] toolkit. The major observations of the work are summarized as follows.

- The proposed framework provides $\sim 24\%$ better health-status identification compared to the baselines, which proves that the entire framework has higher accuracy in assisting users regarding their health-status (refer Table 2).
- The proposed method has $\sim 25 - 45\%$ less latency and power consumption than the existing methods.
- A wearable and low-cost device has been designed and developed which is capable to accumulate health parameters, activity, movement data of user. On the top of that our AI-enabled technology can produce better health-status prediction result considering heterogeneous data-instances (refer Fig. 6).
- The dew-based architecture supports no internet connectivity situation, and the data is cached until the connectivity with the cloud servers is restored (refer Figs. 2 and 3).
- The overall hierarchical cloud-fog-dew framework along with the activity and health analytics module ensure that when any abnormal health situation arises, an immediate measure can be taken (Figs. 2 and 5).

Therefore, we can conclude that the proposed CONFRONT system is a faster health assistance system with higher accuracy level. The higher accuracy also refers that the probability of false recommendation generation is very less in the proposed system with respect to the existing methods. We can assure the proposed CONFRONT framework can be utilized for medical assistance, however, a thorough inspection of the normal range values of health parameter from medical practitioner is recommended. The modules and implementation are scalable enough to modify any such pre-defined health parameter values.

6. Conclusions and future work

This paper proposes a novel framework, *CONFRONT*, which facilitates an efficient COVID-19 in-home health monitoring framework. The hierarchical framework having Cloud-Fog-Dew layers reduces the network usage and cost of execution as well as reduces the latency. Further, the Dew architecture has helped in improving the uptime of the health care model.

The scope of the proposed solution is wide; it not only assists in early identification of individual COVID-19 suspects but can also identify the zones that may require intervention to control the spread. The low-cost wearable sensor can be used to constantly monitor home quarantined patients and timely inform the requirement of intensive hospitalized care. This may significantly reduce the pressure on health resources in the time of pandemic which is a major challenge in developing countries. The proposed CONFRONT framework can be modified and used with other models to develop low-cost solution for clinical diagnosis of different diseases, in future. The framework will continue to be relevant even when there is no existing threat of pandemic as it can be used to perform remote and continuous monitoring of senior citizens' health. As the suggested framework employs GPS sensor, it may be utilized for identifying disease patterns and their endemic nature. Once such patterns are identified, medical supply chains can be automated much more efficiently. Accelerometer based detection can be modified to even detect falls/slips, which may prove fatal in case of an elderly person. When coupled with telemedicine, the proposed framework and its underlying architecture can prove to be a disruptive health care technology.

Declaration of competing interest

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

Acknowledgements

The work is partially supported by Research Grant from Department of Science and Technology, Government of India under Geospatial Chair Professor Award, and TCS PhD Research Fellowship. The authors are thankful to the local medical authority and other stakeholders for supporting this work.

Appendix. Acronyms used in this paper

The acronyms along with the full forms are reported in Table A.4 for better understanding of the work.

[5] MATSIM: https://www.matsim.org/.

References

[1] A.A. Alli, M.M. Alam, The fog cloud of things: A survey on concepts, architecture, standards, tools, and applications, IEEE Internet of Things J. 9 (2020) 100177.

[2] P. Maiti, H.K. Apat, B. Sahoo, A.K. Turuk, An effective approach of latency-aware fog smart gateways deployment for IoT services, IEEE Internet of Things J. 8 (2019) 100091.

[3] C. He, X. Fan, Y. Li, Toward ubiquitous healthcare services with a novel efficient cloud platform, IEEE Trans. Biomed. Eng. 60 (1) (2012) 230–234.

[4] S. Ghosh, J. Das, S.K. Ghosh, R. Buyya, CLAWER: Context-aware cloud-fog based workflow management framework for health emergency services, in: 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGRID, IEEE, 2020, pp. 810–817.

[5] M. Ahmad, M.B. Amin, S. Hussain, B.H. Kang, T. Cheong, S. Lee, Health fog: A novel framework for health and wellness applications, J. Supercomput. 72 (10) (2016) 3677–3695.

[6] S. Zeadally, O. Bello, Harnessing the power of Internet of Things based connectivity to improve healthcare, IEEE Internet of Things J. (2019) 100074.

[7] A.M. Rahmani, T.N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, P. Liljeberg, Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach, Future Gener. Comput. Syst. 78 (2018) 641–658.

[8] A.-M. Rahmani, N.K. Thanigaivelan, T.N. Gia, J. Granados, B. Negash, P. Liljeberg, H. Tenhunen, Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems, in: 2015 12th Annual IEEE Consumer Communications and Networking Conference, CCNC, IEEE, 2015, pp. 826–834.

[9] A. Kumar, K. Sharma, H. Singh, S.G. Naugriya, S.S. Gill, R. Buyya, A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic, Future Gener. Comput. Syst. 115 (2021) 1–19.

[10] S. Tuli, S. Tuli, R. Tuli, S.S. Gill, Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing, IEEE Internet of Things J. 11 (2020) 100222.

[11] J.L. Pépin, R.M. Bruno, R.-Y. Yang, V. Vercamer, P. Jouhaud, P. Escourrou, P. Boutouyrie, Wearable activity trackers for monitoring adherence to home confinement during the COVID-19 pandemic worldwide: Data aggregation and analysis, J. Med. Internet Res. 22 (6) (2020) e19787.

[12] S. Ghosh, A. Mukherjee, S.K. Ghosh, R. Buyya, STOPPAGE: Spatio-temporal data driven cloud-fog-edge computing framework for pandemic monitoring and management, 2021, arXiv preprint arXiv:2104.01600.

[13] S. Tuli, N. Basumatary, S.S. Gill, M. Kahani, R.C. Arya, G.S. Wander, R. Buyya, HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments, Future Gener. Comput. Syst. 104 (2020) 187–200.

[14] A. Paul, H. Pinjari, W.-H. Hong, H.C. Seo, S. Rho, Fog computing-based IoT for health monitoring system, J. Sens. 2018 (2018).

[15] S.S. Gill, R.C. Arya, G.S. Wander, R. Buyya, 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, Springer, 2018, pp. 1376–1383.

[16] Y. Wang, Cloud-dew architecture, Int. J. Cloud Comput. 4 (3) (2015) 199–210.

[17] A. Mukherjee, D. De, S.K. Ghosh, Fogioht: A weighted majority game theory based energy-efficient delay-sensitive fog network for internet of health things, IEEE Internet of Things J. 11 (2020) 100181.

[18] S. Ghosh, A. Mukherjee, S.K. Ghosh, R. Buyya, Mobi-IoST: Mobility-aware cloud-fog-edge-IoT collaborative framework for time-critical applications, IEEE Trans. Netw. Sci. Eng. 7 (4) (2020) 2271–2285, http://dx.doi.org/10.1109/TNSE.2019.2941754.

[19] A. Mukherjee, S. Ghosh, A. Behere, S.K. Ghosh, R. Buyya, Internet of Health Things (IoHT) for personalized health care using integrated edge-fog-cloud network, J. Ambient Intell. Humaniz. Comput. (2020) 1–17.

[20] Y. Wang, Definition and categorization of dew computing, Open J. Cloud Comput. 3 (1) (2016) 1–7.

[21] M. Gushev, Dew computing architecture for cyber-physical systems and IoT, IEEE Internet of Things J. 11 (2020) 100186.

[22] Y. Wang, A blockchain system with lightweight full node based on dew computing, IEEE Internet of Things J. 11 (2020) 100184.

[23] R. Mahmud, F.L. Koch, R. Buyya, Cloud-fog interoperability in IoT-enabled healthcare solutions, in: Proceedings of the 19th International Conference on Distributed Computing and Networking, 2018, pp. 1–10.

[24] S. Kaghyan, H. Sarukhanyan, Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer, Int. J. Inf. Models Anal. 1 (2012) 146–156.

[25] S. Ghosh, S.K. Ghosh, Modeling individual's movement patterns to infer next location from sparse trajectory traces, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2018, pp. 693–698.

[26] W.-j. Guan, Z.-y. Ni, Y. Hu, W.-h. Liang, C.-q. Ou, J.-x. He, L. Liu, H. Shan, C.-l. Lei, D.S. Hui, et al., Clinical characteristics of coronavirus disease 2019 in China, N. Engl. J. Med. 382 (18) (2020) 1708–1720.

[27] Y. Liu, L.-M. Yan, L. Wan, T.-X. Xiang, A. Le, J.-M. Liu, M. Peiris, L.L. Poon, W. Zhang, Viral dynamics in mild and severe cases of COVID-19, Lancet Infect. Dis. 20 (6) (2020) 656–657.

[28] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, F. Chouireb, Fall detection and human activity classification using wearable sensors and compressed sensing, J. Ambient Intell. Humaniz. Comput. 11 (1) (2020) 349–361.

[29] S. Ghosh, S.K. Ghosh, R.D. Das, S. Winter, Activity-based mobility profiling: A purely temporal modeling approach, in: Proceedings of the the Web Conference 2018, 2018, pp. 409–416.

[30] H. Gupta, A. Vahid Dastjerdi, S.K. Ghosh, R. Buyya, iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments, Softw. - Pract. Exp. 47 (9) (2017) 1275–1296.

Anish Poonia completed his M.Tech. [2018–2020] from the Computer Science and Engineering department of Indian Institute of Technology (IIT) Kharagpur, India. He is presently working on multiple defence related projects of Govt of India. His research interests include cloud computing, machine-learning and cyber security.

Shreya Ghosh received the B.Tech. degree [2015] from the Department of Computer Science and Technology at Indian Institute of Engineering Science and Technology, Shibpur (IIEST Shibpur), India, and Ph.D. [2016–2021] from the Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Kharagpur, India. She is currently a senior research fellow with the Department of Computer Science and Engineering, IIT Kharagpur, India working towards developing AI-enabled solutions for smart-city and pandemic management application. Her current research interests include machine learning, trajectory data mining, cloud computing and Internet of Spatial Things. Shreya is the recipient of the prestigious TCS fellowship. She is the member of AnitaB.org and student member of IEEE and ACM.

Akash Ghosh has completed his Bachelor of Engineering [2017–2021] from the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India. He is currently working towards his MS (by Research) degree at the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. He is a senior JBNSTS (Jagadis Bose National Science Talent Search) scholar. His areas of interests include natural language processing, machine learning, artificial intelligence and IoT. He is a student member of ACM.

Shubha Brata Nath received the B.Tech. degree in Computer Science and Engineering from West Bengal University of Technology, India, and the M.E. degree in Software Engineering from Jadavpur University, Kolkata, India. He is currently working towards the Ph.D. degree at the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. His research interests are cloud computing and fog computing. He is a student member of the IEEE.

Soumya K Ghosh received the M.Tech. and Ph.D. degrees from the Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Kharagpur, India. He is currently a Professor with the Department of Computer Science and Engineering, IIT Kharagpur. He was with the Indian Space Research Organization, India. He has authored or coauthored more than 300 research papers in reputed journals and conference proceedings. His current research interests include spatial data science, spatial web services, and cloud computing. He is the recipient of National Geospatial Chair Professor award from Department of Science and Technology, Govt. of India. He is senior member of IEEE and member of ACM.

Rajkumar Buyya is a Redmond Barry distinguished professor and the director of the Cloud Computing and Distributed Systems Laboratory (CLOUDS) Laboratory, University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He is one of the highly cited authors in computer science and software engineering worldwide (h-index = 150, g-index = 322, and 117,400+ citations). Recently, Dr. Buyya is recognized as a "Web of Science Highly Cited Researcher" for five consecutive years since 2016 by Thomson Reuters, a Fellow of IEEE, and Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier for his outstanding contributions to Cloud computing.