

# RESCUE: Enabling green healthcare services using integrated IoT-edge-fog-cloud computing environments

Jaydeep Das<sup>1,2</sup>  | Shreya Ghosh<sup>3,4</sup>  | Anwasha Mukherjee<sup>5</sup>  |  
Soumya K. Ghosh<sup>3</sup>  | Rajkumar Buyya<sup>6</sup> 

<sup>1</sup>Advanced Technology Development Centre, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, India

<sup>2</sup>School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India

<sup>3</sup>Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, West Bengal, India

<sup>4</sup>College of Information Sciences and Technology, The Pennsylvania State University, State College, Pennsylvania, USA

<sup>5</sup>Department of Computer Science, Mahishadal Raj College, Garh Kamalpur, Mahishadal, West Bengal, India

<sup>6</sup>School of Computing and Information Systems, The University of Melbourne, Victoria, Melbourne, Australia

## Correspondence

Jaydeep Das, Advanced Technology Development Centre, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India.

Email: [jaydeep@iitkgp.ac.in](mailto:jaydeep@iitkgp.ac.in)

## Abstract

Internet of Things (IoT) has a pivotal role in developing intelligent and computational solutions to facilitate varied real-life applications. To execute high-end computations and data analytics, IoT and cloud-based solutions play the most significant role. However, frequent communication with long distant cloud servers is not a delay-aware and energy-efficient solution while providing time-critical applications such as healthcare. This article explores the possibilities and opportunities of integrating cloud technology with fog and edge-based computing to provide healthcare services to users in exigency. Here, we propose an end-to-end framework named *RESCUE* (enabling green healthcare services using integrated iot-edge-fog-cloud computing environments), consisting efficient spatio-temporal data analytics module for efficient information sharing, spatio-temporal data analysis to predict the path for users to reach the destination (healthcare center or relief camps) with minimum delay in the time of exigency (say, natural disaster). This module analyzes the collected information through crowd-sourcing and assists the user by extracting optimal path post-disaster when many regions are nonreachable. Our work is different from the existing literature in varied aspects: it analyses the context and semantics by augmenting real-time volunteered geographical information (VGI) and refines it. Furthermore, the novel path prediction module incorporates such VGI instances and predicts routes in emergencies avoiding all possible risks. Also, the design of development of a latency-aware, power-aware data-driven analytics system helps to resolve any spatio-temporal query more efficiently compared to the existing works for any time-critical application. The experimental and simulation results outperform the baselines in terms of accuracy, delay, and power consumption.

## KEYWORDS

cloud computing, edge computing, geospatial query processing, green computing, healthcare service, internet of things, spatio-temporal data

## 1 | INTRODUCTION

Internet of things (IoT) has manifested a dramatic revolution in all spheres of human lives by connecting billions of devices. It is estimated that by the end of 2025, more than 75 billion IoT devices will be connected to the web\*. The proliferation of IoT paradigm has a significant impact in different industries<sup>1</sup> and facilitated varied applications, such as time-critical applications, like healthcare, smart city, smart home, transportation, agriculture<sup>2-4</sup> and so forth. With the rapid development of IoT and other sensor technologies, this IoT paradigm has created new domains of research, namely, *Internet of Health Things (IoHT)*,<sup>5,6</sup> *Internet of Spatial Things (IoST)*,<sup>7</sup> *Industrial Internet of Things (IIoT)*,<sup>8,9</sup> *Internet of Military Things (IoMT)*.<sup>10,11</sup> The IoT devices need to send data to cloud servers frequently for processing and analyzing the accumulated data. However, this increases the delay, therefore affects the Quality of Service (QoS). Here, edge or fog nodes<sup>12,13</sup> extend the functionality of cloud computing by processing, analyzing, and storing the information at the edge of the network.

In the present decade, several IoT devices, such as Raspberry Pi<sup>†</sup>, SmartThings Hub<sup>‡</sup> and so forth, facilitate temporary storage, limited computation capability, and memory resources along with the conventional end-to-end connectivity *anywhere and everywhere*. These promising features have a significant impact on any large-scale IoT deployment, like smart city, smart healthcare, m-Health<sup>14</sup> and so forth. Further, the integration of edge or fog<sup>§</sup> nodes helps in taking an adaptive and dynamic decision based on the sudden changes of the environment and improves the efficacy of the IoT system. However, the computational power of these IoT devices is not sufficient for large-scale and compute-intensive analytics. Cloud computing is the only feasible solution where the processing is carried out in the cloud data centers. Nevertheless, the enormous amount of interconnected IoT devices generate a massive volume of data to be communicated and managed in the cloud server itself, and the cloud datacenters emit an enormous amount of greenhouse gases (due to the high energy consumption) taking a deep toll on the surroundings. On the other side, it is also observed that time-critical applications, such as healthcare, evacuation system, smart traffic monitoring, or defense applications, need real-time and latency-aware decision modules. Frequent communications with distant cloud servers increase the delay and may be challenging for time critical applications. Here, the fog or edge nodes can be made *intelligent* enough to analyze and adapt timely measures to reduce the intervention of cloud servers at each time. While fog or edge computing is not a replacement for cloud computing, the magnificent integration of these two booming technologies can efficiently facilitate delay, energy awareness, and real-time applications. In this work, we leverage the functionality of both cloud and fog/ edge computing to provide timely assistance to users in times of emergency. It is achieved by analyzing heterogeneous data sources, namely, health parameters, contextual information (mobility, environment temperature, air pressure, etc.), and real-time information (crowd-sourcing data). The word “Green” refers to low power; a system with low power or low energy consumption can be referred to as a green system. Due to the enormous volume of data analysis and transmission, accessing various applications through a mobile device has caused a tremendous amount of energy consumption not only by the cloud servers, network but also by mobile devices. Thus, low power, that is, green service provisioning, has become a significant challenge. The work also mathematically formulates the power consumption, latency, and compares them with the baseline methods to prove the eco-friendliness of the proposed framework in terms of low power and low latency.

In recent times, there is a growing need to query and analyze spatio-temporal datasets<sup>16</sup> to extract meaningful information and provide location-aware services, such as trip-planning, weather forecasting, and even health management. From its inception, *spatio-temporal data mining* has shown a significant impact on varied aspects of our lives. For instance, in the year 1854 in London, it was found in a spatio-temporal data analysis<sup>17</sup> that the source of Cholera was public pumps/tube-well and transmitted through contaminated drinking water. The finding was immensely helpful in combating the spread of Cholera. To this end, the internet of spatial things (IoST) combines IoT with spatial context,<sup>7</sup> where the location information of the objects plays an important role. Our proposed framework aims to provide proper assistance to users when emergencies occur, such as disaster or health emergencies. RESCUE assists users by finding a path to reach healthcare centers or other places postdisaster situations or when the patient’s health status is deteriorating. In the latter case, the cloud sends the alert to nearby fog nodes, and the fog nodes inform the ambulance service and nearest healthcare center as a preventive measure. The ambulance’s route or path is predicted by the cloud server, such that the

\*<https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>.

†<https://www.raspberrypi.org/>.

‡<http://www.smartthings.com/>.

§Fog (From cOre to edGe) term was coined by CISCO.<sup>15</sup>

ambulance can reach the healthcare facility with minimum delay. Therefore, an efficient module for analyzing a massive amount of real-time mobility and road information is required to assist the users in exigency situations.

## 1.1 | Motivation and challenges

There are varied real-life applications that can be facilitated through the RESCUE framework. Here, we have considered an exigency situation (say, a natural disaster like supercyclone) when substantial losses occur to human lives and public infrastructures like settlements, roads, electric supply, and so forth. While the normal lives of people are disrupted, getting a proper healthcare facility becomes a challenging issue. For instance, healthcare centers cannot be reached due to the inundation of the roads. Moreover, any emergency or disaster planning requires seamless information exchange and updating information about the affected regions and demands of the people (like healthcare or food facilities). *In this work, the proposed framework, RESCUE, aims to provide a better management framework in terms of providing preliminary healthcare, information collection, and sharing mechanisms, and finding routes to nearby healthcare centers while several regions are not-reachable due to the damage of the disaster.* If the communication backbone is disrupted, the computation is carried out in local nodes (edge) in a distributed fashion and assistance is provided to the users within the coverage of the edge node. It helps in proper postdisaster planning and improves overall urban sustainability and resilience. There are few challenges to provide such facilities in the time of emergency, such as,

1. How proper information about the affected regions can be accumulated quickly to take the recovery steps?
2. How these massive amounts of information can be stored, managed, and analyzed?
3. How to deploy a delay-aware system to assist users in the time of healthcare emergencies, when most of the roads and regions are affected and nonreachable?
4. How can we deploy an energy-efficient framework which provides all of these services?

It is important to note that the cloud paradigm provides the capability of storing, managing, and analyzing a massive volume of data. However, frequent communication with the cloud servers adds more delay and requires more energy consumption. All of these issues need to be addressed and efficiently resolved to provide adequate humanitarian relief and a sustainable environment.

## 1.2 | Contributions

The key contributions of this article are summarized as follows:

- We develop a hierarchical model that captures and accumulates data from heterogeneous IoT devices and performs preliminary data analysis in the edge of the network (i.e., in Fog nodes) to reduce the communication with the distant cloud servers. Further, the local data is stored in the temporary storage of the fog nodes, and only the required/aggregated information is sent to the cloud servers. The proposed framework leverages the VGI (volunteered geographical information) to accumulate information about the affected regions or any events in the surroundings to take countermeasures.
- We consider healthcare service as a prototype for this model, where mobility is an important aspect. RESCUE is conducive to analyzing the users' mobility pattern, present road conditions, and finds an appropriate, less time-consuming path to reach the destination. The model utilizes the *autoencoder* and *markov decision process* to model and predict path to users in less time. It is important to note that RESCUE is flexible enough to find out the paths from source to destination, which requires less fuel consumption by computing the distance of all possible routes efficiently, leading to fewer carbon footprints.
- We also provide a geospatial query processing service, where the user can get the required emergency service information in less execution time as all local geospatial data are stored nearby different fog nodes in a distributed fashion.
- The paper performs extensive simulation-based analysis using iFogSim<sup>55</sup> and real-time data analysis in Google cloud platform (GCP). The experiment results show that our proposed model decreases up to 81% for the indoor user device's

and up to 80% for the outdoor user device's power consumption and reduces the carbon footprints of the IoT, Fog devices, and Google cloud server, which moves a step towards green environment. The latency in healthcare service provisioning and finding routes to assist users is reduced up to 55% for the indoor user and up to 51% for the outdoor user in our proposed framework compared to cloud-only approaches.

The rest of the article is organized as follows. Section 2 represents the related works of the topic. Section 3 elaborates on our proposed RESCUE architecture with mobility and geospatial health query analysis. The latency and power calculations of user devices have been discussed in Section 4. In Section 5, the performance of our proposed architecture is measured with experimental setup details. The last section concludes our paper with future directions.

## 2 | RELATED WORK

Over a decade, several techniques have been adopted to efficiently manage and process a huge volume of spatio-temporal traces to facilitate location-based services. In this regard, Asghari et al.<sup>18</sup> propose a topic modeling framework specifically for spatio-temporal data for real-time processing and learning. A semisupervised learning method is proposed<sup>19</sup> in spatio-temporal social networks to discover POIs (point-of-interest) dynamically. The significance of spatio-temporal relations is addressed and such relations are extracted efficiently in Reference 20. Table 1 summarizes the relevant existing works along with their features.

Along with efficient information processing and management, it is also crucial to reduce the power consumption of the process and reducing overall carbon footprints. Albreem et al.<sup>32</sup> surveyed the existing green IoT research. They categorize the available green initiative into three parts. First part is working over Radio-frequency Identification (RFID) tags.<sup>33-35</sup> In the next part, they make the sensor network energy efficient with different types of routing algorithms,<sup>36-38</sup> clustering schemes.<sup>39,40</sup> The third is green Internet technology with hardware and software solutions for different types of services. A deployment scheme has been proposed by Reference 41. Two types of nodes are considered sensing nodes and relay nodes. Traffic loads are distributed from the sensing node to the relay node as it has direct communications among each other. It reduces the battery power consumption of nodes and increases the overall network lifetime.

Bharti et al.<sup>42</sup> present a framework to recognize and classify complex activities at home using wearable devices. In order to reduce power consumption due to the continuous tracking of mobile devices, a method named *HARKE* has

TABLE 1 Comparative study of the existing works

| Existing works                | Integration of cloud-fog-edge-IoT | Mobility awareness | Crowd-sourcing approach (VGI) | Geospatial query processing | Delay awareness | Low power consumption | Time-critical application |
|-------------------------------|-----------------------------------|--------------------|-------------------------------|-----------------------------|-----------------|-----------------------|---------------------------|
| Kaur et al. <sup>21</sup>     | ✗                                 | ✗                  | ✗                             | ✗                           | ✗               | ✗                     | ✓                         |
| GeoFog4Health <sup>22</sup>   | ✗                                 | ✗                  | ✗                             | ✗                           | ✗               | ✓                     | ✓                         |
| HARKE <sup>23</sup>           | ✗                                 | ✗                  | ✗                             | ✗                           | ✗               | ✓                     | ✗                         |
| Goranson et al. <sup>24</sup> | ✗                                 | ✗                  | ✓                             | ✗                           | ✗               | ✗                     | ✓                         |
| Mooney et al. <sup>25</sup>   | ✗                                 | ✗                  | ✓                             | ✗                           | ✗               | ✗                     | ✓                         |
| Das et al. <sup>26</sup>      | ✓                                 | ✗                  | ✗                             | ✗                           | ✓               | ✓                     | ✓                         |
| IoHT <sup>6</sup>             | ✓                                 | ✓                  | ✗                             | ✗                           | ✓               | ✓                     | ✓                         |
| CLAWER <sup>27</sup>          | ✓                                 | ✓                  | ✗                             | ✗                           | ✓               | ✓                     | ✓                         |
| Spatio-Fog <sup>28</sup>      | ✗                                 | ✗                  | ✗                             | ✓                           | ✓               | ✓                     | ✗                         |
| Mobi-IoST <sup>29</sup>       | ✓                                 | ✓                  | ✗                             | ✗                           | ✓               | ✓                     | ✓                         |
| STOPPAGE <sup>30</sup>        | ✓                                 | ✓                  | ✗                             | ✗                           | ✓               | ✓                     | ✗                         |
| Kafhali et al. <sup>31</sup>  | ✓                                 | ✗                  | ✗                             | ✗                           | ✓               | ✗                     | ✓                         |
| RESCUE (proposed)             | ✓                                 | ✓                  | ✓                             | ✓                           | ✓               | ✓                     | ✓                         |

been proposed in Reference 23. Barik et al.<sup>43</sup> have proposed an ontology-based solution combined with statistical inferencing to recognize complex activities. On the other side, volunteered geographical information (VGI) provides a new opportunity for a better healthcare system by collecting a time-sensitive dataset from a huge number of subjects. Initially,<sup>44</sup> proposed the concept of VGI, where several examples are presented to illustrate the strength of VGI. A systematic overview of public healthcare research using VGI is presented in Reference 24. Modaresnezhad et al.<sup>45</sup> present information technology-enabled crowdsourcing to support several organization needs. Various quality measures and indicators for VGI are mentioned in Reference 46. Barik et al.<sup>22</sup> propose a *Fog-based SDI* framework (*GeoFog4Health*) for analysing big geospatial health data. However, the framework does not support user's mobility, and also does not provide any learning technique to model and predict location related tasks. Kafhali et al.<sup>31</sup> emphasize on personalized health monitoring and improving e-healthcare services using IoT-Fog-Cloud paradigm. The potentials of VGI in pervasive healthcare computing applications are presented in Reference 25, where the authors illustrate varied data sources using openstreetmap (OSM) in their case study. Martin et al.<sup>47</sup> propose an autonomic framework to migrate application modules considering the location-aware feature of fog computing.

There are also varied research works on the internet of health things (IoHT). Mukerjee et al.<sup>6</sup> present a framework for personalized healthcare in the IoT system. The techniques in the optimization of resources in the fog environment are presented in Reference 48. Spatial service orchestration in cloud for geospatial query execution is done in Reference 49. *Spatio-fog* framework<sup>28</sup> proposes an energy-efficient and delay-aware fog computing model for processing geospatial query. Energy-aware popular spatio-temporal data storage in fog devices is described in Reference 50. However, this work does not consider the mobility aspect of the users, which is an important factor in provisioning QoS-aware services. In this regard, Ghosh et al.<sup>29</sup> present a mobility-aware framework (*Mobi-IoST*) that considers the mobility information of the users in a region and assists them to reach the destination. Nevertheless, the framework only predicts route based on road network and user's location, it does not consider present condition of the region after disaster or any emergency. Another framework, namely *LOCATOR*,<sup>51</sup> provides an in-depth analysis of location information and facilitates efficient location-aware services to the users. However, these frameworks cannot handle a disaster scenario, which affects the underlying road network.

Ghosh et al.<sup>27</sup> discuss about *CLAWER* which facilitates an automated workflow among all stakeholders to provide healthcare services to the users in time of emergency. However, *CLAWER* cannot accumulate VGI or crowd-sourcing information and take proper countermeasures by analyzing the correlations of the spatio-temporal information in emergencies. Rajkumar et al.<sup>52</sup> present a smart surveillance system utilizing edge computing to detect and predict abnormal falling activity as well as reducing network bandwidth and latency. Since health data is sensitive in nature, novel blockchain technology is used for high security and integrity of clinical data repository and providing smart e-health system.<sup>53</sup> Healthcare data analysis with user's geolocation has been done in Reference 26. Another recent work<sup>54</sup> extracts the correlations of spatio-temporal events by proposing mobility-association rules. However, it falls short in reducing energy consumption as it only relies on cloud servers to process the data. Das et al.<sup>28</sup> propose *Spatio-Fog* framework which enables geospatial query resolution leveraging fog computing to reduce delay and energy consumption. However, the framework is unable to model user's mobility information and therefore can't predict optimal routes in exigency situation. Another interesting work is presented in Reference 30 where the authors have proposed a cloud-fog-edge enabled pandemic monitoring and management framework based on COVID-19 active cases, users' aggregated mobility and attempted to predict next probable hotspot zones leveraging deep learning method. However, the problem domain of *RESCUE* is different as stated in this paper. The utility of the proposed framework is manifold. Firstly, the deployed hierarchical model is beneficial to provision seamless analytics among several IoT sensors, edge, fog nodes, and cloud server. Further, the placement of services such as VGI, mobility analytics, query processing helps to facilitate effective healthcare service in the time of exigency. *RESCUE* outperforms other existing works in terms of latency, cost, accuracy, and stability. To the best of our knowledge, the proposed framework, *RESCUE*, overcomes all of these limitations as mentioned above of the existing works and provides green and delay-aware systems to assist users in the time of exigency.

### 3 | **RESCUE: PROPOSED ARCHITECTURE**

This section describes our proposed framework, namely *RESCUE*, which facilitates efficient health management and postdisaster recovery mechanisms efficiently in minimum time and power consumption. Figure 1 depicts the overall architecture of *RESCUE*. It is shown that there are two major modules: (i) public health awareness and (ii) home-health

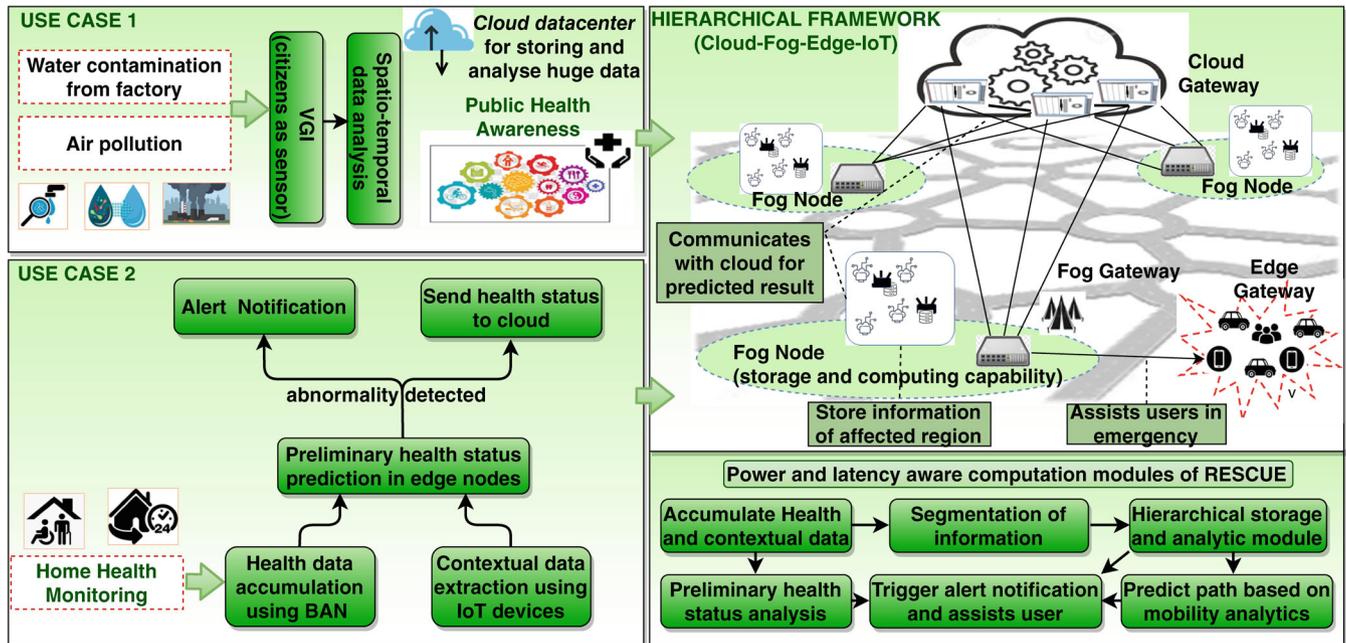


FIGURE 1 RESCUE framework

monitoring of users and assisting them. For the former case, RESCUE depends on crowd-sourcing information or VGI.

It accumulates the information and finds out the correlation of such reported events with spatial information. It helps in enhancing public health awareness and taking preventive measures. In the next case, the health of a user is being monitored using body area network (BAN), and in case any abnormality is detected, proper measures are taken. In both cases, IoT devices, edge nodes, fog nodes, and cloud servers communicate seamlessly to facilitate users' services. In case of a cellular network either small cell cloud enhanced eNodeB (SCcNB) (for the indoor region) or the road side unit (for the outdoor region) works as the fog node. For wireless local area network (WLAN) or wireless metropolitan area network (WMAN), the user is connected with the network through a Wi-Fi access point. In such cases, the switch/router or any other intermediate node acts as the fog node. The major working modules are also shown in the figure, and the modules are latency-aware and require less power consumption.

**Workflow:** The workflow of the modules has been presented with a sequence diagram (Figure 2). Initially, patient's health data, mobility data, and environmental data are collected from different IoT devices. These data are accumulated into the nearby fog devices and primary analysis of the patient's health status is performed here. The primary health report is sent to the patient. Any abnormality is observed at this stage, the primary report sends to the cloud server. Primary health data along with historical medical data of the patient forward to the medical practitioner for detailed analysis. The critical health data is shared with the nearby healthcare center for immediate healthcare facility that is, ambulance, bed booking provision to the patient. After getting the patient's location information from cloud, the ambulance sends for the patient. In the bottom part of the Figure 2, the workflow related to VGI data collection and analysis has been illustrated. When a user logs any event, the information of the event along with the location and temporal information are sent to the nearby fog node through the edge device. The fog nodes perform preliminary analysis and if any abnormality is detected, the information is sent to the cloud server for appropriate countermeasures. It is important to note that in our framework the region is segmented into different grids and the fog nodes cover a specific set of grids and accumulate information from the regions (grids) under its coverage. In this way, efficient storage and spatio-temporal data management and analytics are performed in RESCUE.

As illustrated in Figure 1, the RESCUE framework consists of several components such as *decision maker*, *predictor*, *load generator*, *data collector*, *preprocessing* and so forth. which help to provide effective healthcare service. For instance, data collection and preprocessing modules capture the health, ambiance, and mobility information at different time scales and remove erroneous or duplicate points, and stores the information. The load balancer module of query processing

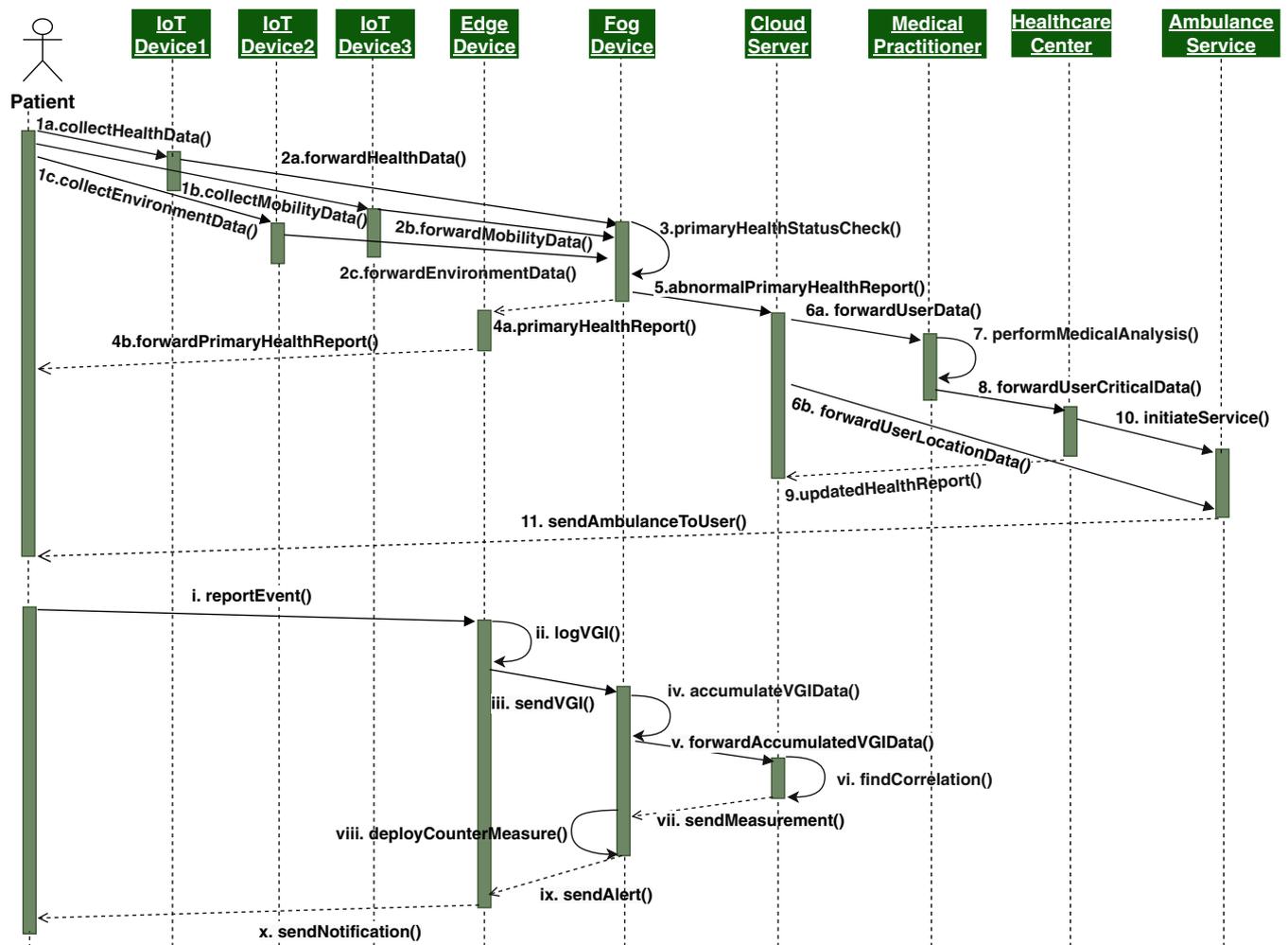


FIGURE 2 RESCUE sequence diagram

is useful to reduce the computational load of spatio-temporal query processing. Further, decision-maker and predictor modules help to extract whether there is some abnormal health condition of the user and predicts the optimal path in case of emergency. In brief, all of these components contribute to the optimality of the end-to-end framework and facilitate efficient time-critical application, say, healthcare.

### 3.1 | Public health management: Volunteered geographic information (VGI) approach

In the era of sensor network development, VGI can be viewed as *human as sensor* or *citizen as sensor*, since a number of independent individuals (human/ citizen) can provide information about their surroundings. VGI has a great potential in public health monitoring. For instance, VGI could assist emergencies like the recent outbreak of COVID-19. The users can log information about the suspected cases (such as having symptoms or having a recent travel history from the affected regions in the world). The countries with high population densities, like India, can also benefit by collecting information like a availability of essential commodities or medicines in a lockdown situation.

In short, VGI refers to the use of smart devices to assemble, modify, and share geographical information provided by users voluntarily. Now, the volunteers can give a large amount of data at different spatio-temporal resolutions. However, there are few challenges: (a) since the data is collected from different people, the accumulated data can be heterogeneous, (b) the reliability of data quality needs to be maintained. In this work, we mainly focus on public health status. To restrict the heterogeneous nature of the collected data, we provide a list of events ( $ev$ ) such as accidents, water contamination,

excess household wastage, or a sudden outbreak of a disease. The volunteers can select any option from the list and mark the severity ( $Se_{ev}$ ) of the event. Furthermore, they also log the location information for such events. For maintaining the data quality, we have deployed a *hierarchical approach*, where RESCUE relies on a reduced group of trusted individuals (who act as moderators). Furthermore, when a large amount of data is collected from a region and such moderators are not present, in that case, we follow *Crowdsourced approach*, where the convergence on the reliability of the data is fully dependent on the crowd (or volunteers) by identifying and correcting errors collectively. Once the dataset is collected, we form different spatial clusters with the collected dataset. Each cluster consists of the information  $\langle ev_i, Se_{ev_i}, cardinality \rangle$ . Here,  $ev_i$  is an event  $i$  with severity  $Se_{ev_i}$ . The *cardinality* is computed by aggregating the number of data entries with the same value. Next, we generate a *heatmap* using this information from all of the places and find out the correlation with other contextual parameters.

---

**Algorithm 1.** Extracting routes from source to destination in time of exigency
 

---

**Input:** VGI, mobility and POI data of  $N$  connected regions ( $R$ ) of graph  $G(V, E)$ 
**Output:** Route  $\langle Route(S, D, Y) \rangle$ 
 $\triangleright$  Route from source  $S$  to destination  $D$ 

```

1:  $V, E, Y \leftarrow NULL$ ;
2: for each regions  $l_j \in N$  do  $\triangleright$  Check all regions for accumulated VGI data of events
3:   for each GPS point  $p_i \in R$  do
4:     for each accumulated VGI  $v_k \in V$  do
5:        $flag \leftarrow checkAuthen(v_k, p_i)$ 
6:       if  $flag == 1$  then
7:          $S_j \leftarrow ComputeCardinality(v_k)$   $\triangleright$  Compute the number of times an event is reported in a particular
           location and store using hierarchical indexing
8:          $S_j \leftarrow geotagg()$   $\triangleright$  Associate location information with the reported event
9:          $h_j \leftarrow heatmapGen(S_j)$   $\triangleright$  Generate a heatmap and analyse the correlation value
10:         $S.insert(h_j)$   $\triangleright$  Append the event in the datastore
11:       end if
12:     end for
13:   end for
14: end for
15: for each mobility trace  $tr \in T$  do  $\triangleright$  Analyse the mobility information in the region
16:   for each event information  $h \in S$  do
17:      $GE.append(h)$   $\triangleright$  A new node in the affected region is generated
18:      $GE.buffer(50)$   $\triangleright$  Buffer with 50 meter for each affected region is generated
19:      $path \leftarrow autoencoder(GE, POI, M)$   $\triangleright$  Learn representation of data and context using autoencoder
20:      $path \leftarrow refinement(GE, v_{risk}, path)$   $\triangleright$  Extract routes using MDP
21:      $Y : route \leftarrow$  Add nodes from  $path$ 
22:      $t \leftarrow extractTemporal(path)$   $\triangleright$  Extract path with less commute time
23:      $Y.append(Edge(G, t))$   $\triangleright$  Append edges between the nodes to complete the route from source to destination
24:     Print  $Y$   $\triangleright$  Print the path
25:   end for
26: end for

```

---

RESCUE also can analyze such historical records, if available. For instance, in some villages of India, in *rainy season*, few infectious diseases (e.g., Cholera or Dengue) occur on a large scale <sup>¶</sup>. Likewise, several diseases depend on temperature, humidity, and rainfall patterns. If these relations can be found apriori, early preventive measures can be adapted to prevent widespread of diseases. Thus, we can find out the hotspots of such diseases in different spatial and temporal scales. Now, in the VGI approach, the time complexity of the process will be  $O(M)$ , where  $M$  denotes the number of sources from where the data are being collected.

<sup>¶</sup><https://bit.ly/3gZ000D>.

### 3.2 | Mobility analysis to find the routes

In this section, we describe the process of finding a path to reach the destination in minimal time, avoiding the regions with risk (or affected areas). First, we model the study region in a *graphical structure* where the nodes are the POIs, and the edges are the road segments. The region is segmented into uniform grids, and the fog nodes store the road-network information as well as the POI information of the region under its coverage area. The fog nodes are capable of communicating with each other and sends the information to the cloud gateway, which forwards it to the cloud. The route finding method requires a large set of data analysis. Hence the cloud will perform the data analysis and extract the optimal path. We begin the discussion by defining the preliminary concepts as follows:

1. Road network ( $R(V, E)$ ): The underlying road network is defined by a directed graph, where the edges  $e_i \in |E|$  are the road-segments and the intersections of the edges are represented by nodes  $v_i \in |V|$ .
2. POI ( $P$ ): The POI or Point-of-Interest depicts the landmarks of a region, such as residential area, commercial area, and so forth. We build a tree-based structure to store all of these POI information.
3. Route ( $S, D, Y$ ): The route represents a subgraph of  $R(V, E)$  where the two end-points are the source ( $S$ ) and destination ( $D$ ) nodes and the intermediate edges and nodes represent the optimal path to reach the destination.

It may be noted that the volume of data including road-network structure, POIs, and movement information is huge, and increases along with the spatial range of the study region. It is not possible to store all such information in a centralized server. To this end, we build an efficient indexing scheme and deploy it. Here, we have used the large cell base stations (road side unit or RSU) as the fog nodes. The RSUs have temporary storage and computing capabilities. These are divided into two categories (i) macro RSU and (ii) micro RSU. The macro RSU has a coverage area of 1–20 km and the micro RSU has a coverage area of 200 m–1 km. All the POI information and other mobility-related information are stored in these RSUs. We segregate the study regions into hexagonal grids of uniform area. Each of the center points of the grids is extracted and geo-hash code is generated. The list of the geo-hash codes is stored in the cloud server. The fog nodes store the POIs within its coverage using a layered hashing scheme. The base idea is taken from Reference 54. RESCUE uses three layers of hashing, where each layer stores more granular spatial information.

RESCUE uses *Markov Decision Process (MDP)* to recommend the routes from a given source to a destination. Typically, MDP is used as it has a fast convergence speed and it provides a global optimal route instead of only considering the local benefits. Initially, autoencoder based reduction method is deployed such that complex nonlinear relationship can be retained. The input datasources include road-network structure, road properties (length, road-type, lane-type), crowd-level, traffic density, type of neighborhood areas and so forth. When a exigency situation occurs, we need to find out the safe routes from source to destination region. For such scenario, the similarity between two regions (grids) are important, and the reduced data should capture this similarity. Also, the impact of different factors should be assigned different weight in our problem scenario. The input is high dimensional data which is converted into a input vector of size 30, which is further reduced layer by layer using multilayer perceptron until the vector length is 2. RESCUE uses fully connected layer at each stage. Next, *cross entropy loss* is computed using reduced data and the source data. For optimizing data reduction process, *back propagation* is used, where different weights for different dimensions of input data is allocated to capture the importance of different features.

Next, MDP is represented as 5-tuple:  $(POI, S, Pr, Path_{re}, \vartheta)$ . Here,  $POI$  is the set of regions, where user can move from one region to another following one of the actions ( $S$ ).  $Pr$  is the corresponding probability, and  $Path_{re}$  represents the reward to move to the next region.  $\vartheta$  is the discount factor to select the global optimal route from source to destination. The *reward function*( $path_{re}$ ) of the process is defined as:

$$Path_{re}(POI_a, POI_b, Seg) = \frac{Risk_d(POI_a, POI_b)}{NW_{dis}(POI_a, POI_b)^{\frac{Risk_d(POI_a, POI_b)}{|Risk_d(POI_a, POI_b)|}}}. \quad (1)$$

This is the reward function when a particular grid ( $Seg$ ) is selected in the path. It may be noted that when the user is in a grid, it is only possible to visit any of its neighboring grids in the next step. The  $NW_{dis}$  is the *Haversine distance* between  $POI_a$  and  $POI_b$ . It is assumed that the user wants to visit  $POI_b$  starting from  $POI_a$ . The difference of risk in moving from  $POI_a$  to  $POI_b$  is represented by  $Risk_d(POI_a, POI_b)$ . These risk values are computed from the available data sources like crowd-sourcing data. The heatmap is used to compute the risk value in a range of [0,1] for any particular location.

Next, we represent *policy* as  $\mu$  which is basically distribution over actions given states and defines the overall behavior of the agent in the model. Next, we define the cumulative reward to  $POI_a$  as  $v(POI_a)$ :

$$v(POI_a) = E(Path_{re_{POI_a}} + \vartheta v_{POI_b}). \quad (2)$$

$$v_i(POI_a) = Path_{re_i} + \vartheta Path_{re_{i+1}} + \vartheta^2 Path_{re_{i+2}} + \dots = \sum_{t=0}^{\infty} \vartheta^t Path_{re_{i+t+1}}. \quad (3)$$

It can be observed that if discount factor is set larger, then the recommended route will be longer. The policy ( $\mu$ ) is defined as:

$$\mu(POI_a) = \arg \max_s (Path_{re_{POI_a}^s} + \vartheta v_{i+1}(POI_b)). \quad (4)$$

Algorithm 1 represents is the basic steps of modeling such networks along with the movement information. Here, we have used the autoencoder and Markov decision process to model the road network and related information and subsequently finding the path to reach the destination from the source. A Road network is defined by a directional graph with road segments as edges and the intersection points of the road segments as nodes. In the time of the disaster, information from heterogeneous sources needs to be augmented which makes it a high-dimensional dataset and the process becomes time-consuming. For this reason, here RESCUE uses an *autoencoder* to learn the representation of data related to an emergency situation by dimensionality reduction. Here, we have used autoencoder for dimensionality reduction in the context of mobility data representation and path finding. The major motivation behind using autoencoder instead of PCA is that it is important to capture the nonlinear relationship in the feature space. As it can be understood that the features (considering location, mobility and other contextual information) have nonlinear relationship with each other, autoencoder is better suitable to compress the related information into low dimensional latent space utilizing the capability to model complex nonlinear functions. On the other side, PCA features are projections onto the orthogonal basis, therefore unable of modeling complex nonlinear functions accurately.

Here, we deploy two-phase processing: offline and online. In the offline process, we compute and store the optimal paths between different pairs of locations, which are accessed frequently. The extraction of *optimal path* can depend on various aspects, like, minimum commute time, shortest distance, or minimum fuel consumption. In the *online* phase, when a path-query is processed, the fog nodes take the computed route from the given source to destination and checks the feasibility of the solution. Next, feedback is sent to the cloud server, in case the route is not optimal. In our method, we have selected the size of the grid is 8 m (length of one side of hexagonal grid). In the storage, each temporal layer (hashing) stores information of 30 min time-interval and each spatial bucket stores information of 36 closest neighbor grids.

### 3.3 | Geospatial query and services

Geospatial service helps to retrieve the geospatial data from the different databases seamlessly. There are many types of open geospatial consortium (OGC<sup>#</sup>) standardize spatial services available. Web feature service (WFS<sup>||</sup>) extracts the featured geospatial data from the database. Web processing service (WPS<sup>\*\*</sup>) processes the geometrical operations, that is, overlap, buffer, intersection, cross, and so forth. over existing geospatial data. Web coverage service (WCS<sup>††</sup>) accesses the multidimensional geospatial data from the server database. Web map service (WMS<sup>‡‡</sup>) displays the map with the user's point of interest(POI). Catalog service (CSW<sup>§§</sup>) keeps the information about the available geospatial data along with its

<sup>#</sup><https://www.ogc.org/>.

<sup>||</sup><https://www.ogc.org/standards/wfs>.

<sup>\*\*</sup><https://www.ogc.org/standards/wps>.

<sup>††</sup><https://www.ogc.org/standards/wcs>.

<sup>‡‡</sup><https://www.ogc.org/standards/wms>.

<sup>§§</sup><https://www.ogc.org/standards/cat>

source. Some examples of healthcare system related geospatial query where the geospatial services are applied. Geospatial Query 1 ( $GQ_1$ ): *List out the hospitals where MRI facility available of city C.* SQL syntax of  $GQ_1$  is as follow:

```
SELECT H.name, H.time, H.price
FROM Hospital H
WHERE H.facility='MRI' AND City='C';
```

In this geospatial query, one layer(hospital facility) is involved. A geospatial filter operation (facility="MRI") is applied over the layer. The query parse tree of  $GQ_1$  is presented in Figure 3. To retrieve hospital data from multiple hospital data sources, the essential geospatial web services for  $GQ_1$  are as follows.

1. getFeature feature service is needed to retrieve hospital data with MRI facility availability in city C.
2. getMap service displays the result of  $GQ_1$  on a map.

Geospatial Query 2 ( $GQ_2$ ): *Find out the hospital details which are available within  $r$  kilometers radius of a point  $P(x,y)$  in ascending order by distance.  $x$  and  $y$  are latitude and longitude of point  $P$ .* SQL syntax of  $GQ_2$  is as follow:

```
SELECT H.name, H.address, H.contact, H.distance
FROM Hospital H
WHERE Overlap (H.shape, Buffer('P(x,y)', r))=1 ORDER BY H.distance ASC;
```

In this geospatial query, two thematic layers, Land use land cover (LULC) and Hospital, are involved. Distance calculation between the user location ( $x,y$ ) and different hospitals is possible only after the integration or overlying of these two layers. In general, the Euclidean distance is calculated between two points. But, in a real scenario, there may not have a path between these two points. The shortest Euclidean distant hospital may have no path to reach there. Whereas, the Euclidean distance-wise comparatively far hospital has good communication and reach there quickly. So, only one or two thematic layers are not capable enough to consider all real-life factors. Many thematic layers' involvement is required. But, the increment of layers also increases the computational complexities and requires more resources and energy to compute these. The query parse tree of  $GQ_2$  is presented in Figure 4. To retrieve geospatial data from multiple data sources, the essential geospatial web services for  $GQ_2$  are as follows.

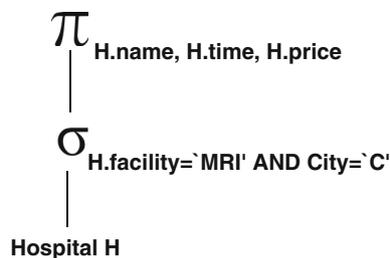


FIGURE 3 Query parse tree of  $GQ_1$

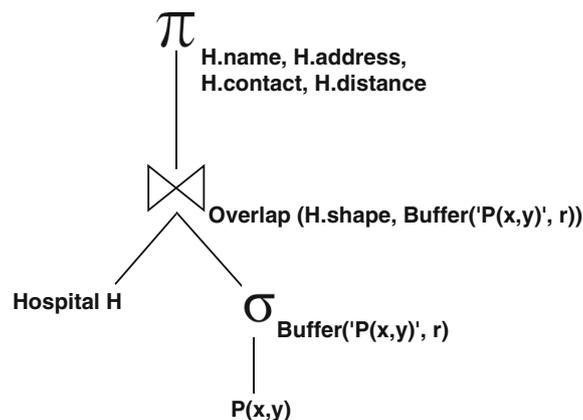


FIGURE 4 Query parse tree of  $GQ_2$

1. BufferFeatureCollection processing service is needed for " $r$ " km radius buffer.
2. IntersectionFeatureCollection processing service is needed for overlapping of point  $P(x,y)$  and hospital layer.
3. getMap service displays the result of  $GQ_2$  on a map.

#### 4 | LATENCY AND POWER CONSUMPTION OF USER DEVICE DURING HEALTH STATUS DETECTION

In the proposed framework, it is observed that the sensor nodes do the data collection, and the accumulation is done inside the smartphone, which then forwards the data to the fog device. In case of abnormal health conditions, the fog device forwards the data to the cloud. Further data analysis is performed inside the cloud, and notification is sent in case of an emergency. Here, to calculate the latency and power consumption of the user's smartphone in this entire process, two scenarios are considered: indoor and outdoor. In case of the indoor region, either small cell cloud enhanced enodeB (SCcNB), which is a small cell base station equipped with storage and computational resources (if the user is registered under a cellular network), or the switch, router working as fog device (if the user is connected with Wi-Fi) is used for connecting the smartphone with the network. In the case of the outdoor region, the user is usually connected with the network through the road side unit (RSU). Now, for these two scenarios, the latency and power consumption of the user device during the entire process of health status detection will be determined. The parameters used in latency and power consumption calculations are defined in Table 2.

**TABLE 2** Parameters used in latency and power calculation

| Parameter | Definition   |
|-----------|--|
| $D_{sj}$  | Data amount collected by sensor $j$  |
| $D_m$     | Data amount accumulated inside the smart phone and transmitted to the fog device         |
| $D_c$     | Data amount transmitted by the fog device after processing to the cloud through Gateway  |
| $F_s$     | Link failure rate from sensor to smart phone   |
| $F_{mi}$  | Link failure rate from smart phone to SCcNB/ fog device                                  |
| $F_{mo}$  | Link failure rate from smart phone to RSU  |
| $F_{fi}$  | Link failure rate from SCcNB/ fog device to cloud  |
| $F_{fo}$  | Link failure rate from RSU to cloud  |
| $L_{qpi}$ | Latency in processing query if the device is at indoor region                            |
| $L_{qpo}$ | Latency in processing query if the device is at outdoor region                           |
| $N_s$     | Number of sensor nodes collecting health, movement, environmental data                   |
| $P_t$     | Power consumption of the smart phone per unit time in data transmission mode             |
| $P_r$     | Power consumption of the smart phone per unit time in data reception mode                |
| $P_a$     | Power consumption of the smart phone per unit time during data accumulation              |
| $P_i$     | Power consumption of the smart phone per unit time in idle mode                          |
| $R_{sm}$  | Data amount transmitted per unit time from sensor node to smart phone                    |
| $R_{mf}$  | Data amount transmitted per unit time from smart phone to SCcNB/fog device/ RSU          |
| $R_{fc}$  | Data amount transmitted per unit time from fog device/SCcNB/RSU to cloud through Gateway |
| $S_{sj}$  | Data collection speed of sensor node $j$   |
| $S_m$     | Data accumulation speed of smart phone   |
| $S_{fi}$  | Data processing speed of fog device/ SCcNB   |
| $S_{fo}$  | Data processing speed of RSU   |
| $S_c$     | Data processing speed of cloud   |

## 4.1 | Latency in case of indoor region

The latency in data collection by the sensor nodes is given as,

$$L_{cs} = \max((D_{s1}/S_{s1}), (D_{s2}/S_{s2}), \dots, (D_{s_{N_s}}/S_{s_{N_s}})). \quad (5)$$

The latency in sending data from sensor nodes to smart phone is given as,

$$L_{sm} = \max(((D_{s1}/R_{sm}) \cdot (1 + F_s)), ((D_{s2}/R_{sm}) \cdot (1 + F_s)), \dots, ((D_{s_{N_s}}/R_{sm}) \cdot (1 + F_s))). \quad (6)$$

As there are multiple sensor nodes, the maximum latency is considered. The latency in data accumulation inside the smart phone is given as,

$$L_{ac} = (D_m/S_m). \quad (7)$$

The latency in sending data from smart phone to SCceNB/fog device is given as,

$$L_{mfi} = (D_m/R_{mf}) \cdot (1 + F_{mi}). \quad (8)$$

The latency in data processing inside the fog device/SCceNB is given as,

$$L_{fpi} = (D_m/S_{fi}). \quad (9)$$

The latency in sending data from SCceNB/fog device to the cloud through Gateway is given as,

$$L_{fci} = (D_c/R_{fc}) \cdot (1 + F_{fi}). \quad (10)$$

The latency in data analysis inside the cloud is given as,

$$L_{cp} = (D_c/S_c). \quad (11)$$

Therefore the total latency for health status detection while the user is at indoor region is given as,

$$L_{toti} = L_{cs} + L_{sm} + L_{ac} + L_{mfi} + L_{fpi} + L_{fci} + L_{cp} + L_{qpi}, \quad (12)$$

where  $L_{qpi}$  is query processing latency (including path prediction) if the user is at indoor region. If cloud only system is used, the data processing entirely happens inside the cloud. In that case, the latency for health status detection while the user is at indoor region is expressed as,

$$L_{totic} = L_{cs} + L_{sm} + L_{ac} + L_{mfi} + L_{fci} + L_{cpc} + L_{qpi}, \quad (13)$$

where  $L_{fci} = (D_m/R_{fc}) \cdot (1 + F_{fi})$ ,  $L_{cpc} = (D_m/S_c)$ . As in the cloud only system the data processing entire happens inside the cloud, the amount of data transmission to the cloud is higher in this case, which increases the network traffic as well as the total latency. The use of the intermediate device (SCceNB/fog device) in data processing reduces the amount of data transmission to the cloud, that consequently reduces the network traffic as well as the total latency. The reduction in latency using the proposed system than the cloud only system for the indoor region is then given as,

$$R_{Li} = (L_{totic} - L_{toti})/L_{totic}. \quad (14)$$

## 4.2 | Latency in case of outdoor region

The latency in data collection by the sensor nodes is determined using Equation (5). The latency in sending data from sensor nodes to a smartphone is determined using Equation (6). The latency in data accumulation inside the smartphone is determined using Equation (7). The latency in sending data from the smartphone to RSU is given as,

$$L_{mfo} = (D_m/R_{mf}) \cdot (1 + F_{mo}). \quad (15)$$

The latency in data processing inside the RSU is given as,

$$L_{fpo} = (D_m/S_{fo}). \quad (16)$$

The latency in sending data from RSU to the cloud through Gateway is given as,

$$L_{fco} = (D_c/R_{fc}) \cdot (1 + F_{fo}). \quad (17)$$

The latency in data analysis inside the cloud is determined using Equation (11). The total latency for health status detection while the user is at outdoor region is given as,

$$L_{toto} = L_{cs} + L_{sm} + L_{ac} + L_{mfo} + L_{fpo} + L_{fco} + L_{cp} + L_{qpo}, \quad (18)$$

where  $L_{qpo}$  is query processing latency (including path prediction) if the user is at outdoor region. If cloud only system is used, the data processing entirely happens inside the cloud. In that case, the latency for health status detection while the user is at outdoor region is expressed as,

$$L_{totoc} = L_{cs} + L_{sm} + L_{ac} + L_{mfo} + L_{fcoc} + L_{cpc} + L_{qpo}, \quad (19)$$

where  $L_{fcoc} = (D_m/R_{fc}) \cdot (1 + F_{fo})$ ,  $L_{cpc} = (D_m/S_c)$ . As in the cloud only system the data processing entire happens inside the cloud, the amount of data transmission to the cloud is higher in this case, which increases the network traffic as well as the total latency. The use of the intermediate device (RSU) in data processing reduces the amount of data transmission to the cloud, that consequently reduces the network traffic as well as the total latency. The reduction in latency using the proposed system than the cloud only system for the outdoor region is then expressed as,

$$R_{Lo} = (L_{totoc} - L_{toto})/L_{totoc}. \quad (20)$$

### 4.3 | Power consumption of user device in case of indoor region

The power consumption of the smartphone, that is, user device while data collection take place by sensor nodes is given as,

$$P_{cs} = P_i \cdot L_{cs}. \quad (21)$$

The power consumption of the user device while receiving data from sensor nodes is given as,

$$P_{sm} = P_r \cdot L_{sm}. \quad (22)$$

The power consumption of the user device during data accumulation is given as,

$$P_{ac} = P_a \cdot L_{ac}. \quad (23)$$

The power consumption of the user device while transmitting data to the SCcNB/fog device is given as,

$$P_{mfi} = P_t \cdot L_{mfi}. \quad (24)$$

The power consumption of the user device while data processing takes place inside the SCcNB/ fog device is given as,

$$P_{fpi} = P_i \cdot L_{fpi}. \quad (25)$$

The power consumption of the user device while fog device/SCcNB transmits data to the cloud through the Gateway given as,

$$P_{fci} = P_i \cdot L_{fci}. \quad (26)$$

The power consumption of the user device while data analysis takes place inside the cloud is given as,

$$P_{cp} = P_i \cdot L_{cp}. \quad (27)$$

The total power consumption of the user device during the entire process of health status detection while the user is at indoor region is therefore given as,

$$P_{toti} = P_{cs} + P_{sm} + P_{ac} + P_{mfi} + P_{fpi} + P_{fci} + P_{cp} + (L_{qpi} \cdot P_i). \quad (28)$$

If cloud only system is used, the data processing entirely happens inside the cloud. In that case, the total power consumption of the user device during the entire process of health status detection while the user is at indoor region is given as,

$$P_{totic} = P_{cs} + P_{sm} + P_{ac} + P_{mfi} + P_{fcic} + P_{cpc} + (L_{qpi} \cdot P_i), \quad (29)$$

where  $P_{fcic} = P_i \cdot L_{fcic}$ ,  $P_{cpc} = P_i \cdot L_{cpc}$ ,  $L_{fcic} = (D_m/R_{fc}) \cdot (1 + F_{fi})$ ,  $L_{cpc} = (D_m/S_c)$ . As in the cloud only system the data processing entire happens inside the cloud, the amount of data transmission to the cloud is higher in this case, which increases the total latency and consequently, the power consumption of the user device during the period. The use of the intermediate device (SCcNB/fog device) in data processing reduces the amount of data transmission to the cloud, that consequently reduces the total latency and power consumption of the user device. The reduction in power consumption of the user device using the proposed system than the cloud only system for the outdoor region is then expressed as,

$$R_{Pi} = (P_{totic} - P_{toti})/P_{totic}. \quad (30)$$

#### 4.4 | Power consumption of user device in case of outdoor region

The power consumption of the smartphone, that is, user device while data collection takes place by sensor nodes, is determined using Equation (21). The power consumption of the user device while receiving data from sensor nodes is determined using Equation (22). The power consumption of the user device during data accumulation is determined using Equation (23). The power consumption of the user device while transmitting data to the RSU is given as,

$$P_{mfo} = P_t \cdot L_{mfo}. \quad (31)$$

The power consumption of the user device while data processing takes place inside the RSU is given as,

$$P_{fpo} = P_i \cdot L_{fpo}. \quad (32)$$

The power consumption of the user device while RSU transmits data to the cloud through the Gateway given as,

$$P_{fco} = P_i \cdot L_{fco}. \quad (33)$$

The power consumption of the user device, while data analysis takes place inside the cloud, is determined using Equation (27). The total power consumption of the user device during the entire process of health status detection, while the user is at the outdoor region, is therefore given as,

$$P_{toto} = P_{cs} + P_{sm} + P_{ac} + P_{mfo} + P_{fpo} + P_{fco} + P_{cp} + (L_{qpo} \cdot P_i). \quad (34)$$

For the outdoor region there is a higher probability of connection interruption with respect to the indoor region. Thus, the latency may be higher if the user is at the outdoor region and the power consumption during the period also becomes high consequently. If cloud only system is used, the data processing entirely happens inside the cloud. In that case, the total power consumption of the user device during the entire process of health status detection while the user is at outdoor region is expressed as,

$$P_{totoc} = P_{cs} + P_{sm} + P_{ac} + P_{mfo} + P_{fcoc} + P_{cpc} + (L_{qpo} \cdot P_i), \quad (35)$$

where  $P_{fcoc} = P_i \cdot L_{fcoc}$ ,  $P_{cpc} = P_i \cdot L_{cpc}$ ,  $L_{fcoc} = (D_m/R_{fc}) \cdot (1 + F_{fo})$ ,  $L_{cpc} = (D_m/S_c)$ . As in the cloud only system the data processing entire happens inside the cloud, the amount of data transmission to the cloud is higher in this case, which increases the total latency and consequently, the power consumption of the user device during the period. The use of the intermediate device (RSU) in data processing reduces the amount of data transmission to the cloud, that consequently reduces the total latency and power consumption of the user device. The reduction in power consumption of the user device using the proposed system than the cloud only system for the outdoor region is then expressed as,

$$R_{Po} = (P_{totoc} - P_{toto})/P_{totoc}. \quad (36)$$

The latency and power consumption of the user device, that is, smartphone during the entire process, will be graphically compared with the cloud-only framework in the next section.

In the latency and power calculation model, the latency in path prediction and alert message sending to the smartphone has not been yet considered. The total latency in path prediction, alert message, and path information transmission to the smartphone is given as: For indoor region:

$$L_{pathalin} = ((D_{al} + D_{path})/R_{cf}) \cdot (1 + F_{cf}) + ((D_{al} + D_{path})/R_{fm}) \cdot (1 + F_{fm}). \quad (37)$$

For outdoor region:

$$L_{pathalout} = ((D_{al} + D_{path})/R_{cr}) \cdot (1 + F_{cr}) + ((D_{al} + D_{path})/R_{rm}) \cdot (1 + F_{rm}). \quad (38)$$

The power consumption of the user device during this period will be:

For indoor region:

$$P_{pathalin} = (((D_{al} + D_{path})/R_{cf}) \cdot (1 + F_{cf})) \cdot P_i + (((D_{al} + D_{path})/R_{fm}) \cdot (1 + F_{fm})) \cdot P_r. \quad (39)$$

For outdoor region:

$$P_{pathalout} = (((D_{al} + D_{path})/R_{cr}) \cdot (1 + F_{cr})) \cdot P_i + (((D_{al} + D_{path})/R_{rm}) \cdot (1 + F_{rm})) \cdot P_r, \quad (40)$$

where  $D_{al}$ ,  $D_{path}$  are the data amount transmitted for alert message and path information respectively,  $R_{cr}$ ,  $R_{rm}$ ,  $R_{cf}$ ,  $R_{fm}$  are the data amount transmitted per unit time from cloud to RSU, RSU to user device, cloud to SCcNB/ fog device, SCcNB/ fog device to user device respectively,  $F_{cr}$ ,  $F_{rm}$ ,  $F_{cf}$ ,  $F_{fm}$  are the link failure rate from cloud to RSU, RSU to user device, cloud to SCcNB/fog device, SCcNB/fog device to user device respectively. In that case the total latency for indoor region will be  $(L_{toti} + L_{pathalin})$  and for outdoor region the total latency will be  $(L_{toto} + L_{pathalout})$ . In that case, the total power consumption of the user device for the indoor user will be  $(P_{toti} + P_{pathalin})$ , and for the outdoor user, the total power consumption of the user device will be  $(P_{toto} + P_{pathalout})$ .

## 5 | PERFORMANCE EVALUATION

In this section, we illustrate the efficacy of our system by developing a test-bed as well as using a simulation toolkit.

## 5.1 | Experimental test-bed

We have developed an experimental test-bed for evaluating the efficacy of the proposed system. We have used the compute engine and app engine of Google cloud platform (GCP) to carry out the spatio-temporal data analysis. In the test-bed, we have used Raspberry Pi 3 as the fog device. We have designed the android application using Android Studio 4.1 with *Firebase database* support. It is used to collect VGI data and personalized health data from the users. In Figure 5A, the user is asked to select the option between VGI and personalized health data. Personalized health data refers to the information collected from the BAN. When the user selects the personalized health data option, the BAN sensors are synced, and information is logged. On the other side, if the user selects the VGI option, then the next page of the app opens (as illustrated in Figure 5B). Here, the user is asked to select the type of event (such as accident or water contamination, etc.) he/she wants to log. In our case study, we have provided a list of event types, where the user may select one or more than one. Also, the user can add any other events not listed in the selection menu. Next, the user logs the severity of the event on a scale (0,5) and provides the location of the event occurrence.

As soon as the user submits the information, the data is sent to the nearby fog node and the information along with the timestamp is accumulated in the fog node. When more data points are accumulated, the fog node performs preliminary analysis and sends the report to the cloud server. Since the fog nodes cover a particular geographical region, the information of different regions are sent to the cloud by different fog nodes. Then, the cloud server can perform aggregate analysis as and when required. The app can also collect the contextual data (location, acceleration, proximity, temperature, and light sensor data) from the smartphone's in-built sensors using the *Android sensor framework*. The application can also communicate with wearable devices such as smart-watch (Fitbit), body temperature measuring module, and SPO2 tracking module. In the Raspberry Pi 3, we have installed the *Eddystone Bluetooth Beacon*, for sending data periodically. For evaluating the spatio-temporal analysis, we have implemented the methods in GCP and QGIS framework. For this prediction, we have considered a region, *Bankura* district of West Bengal, India. The experimental and simulation results are discussed in subsequent sections of the paper.

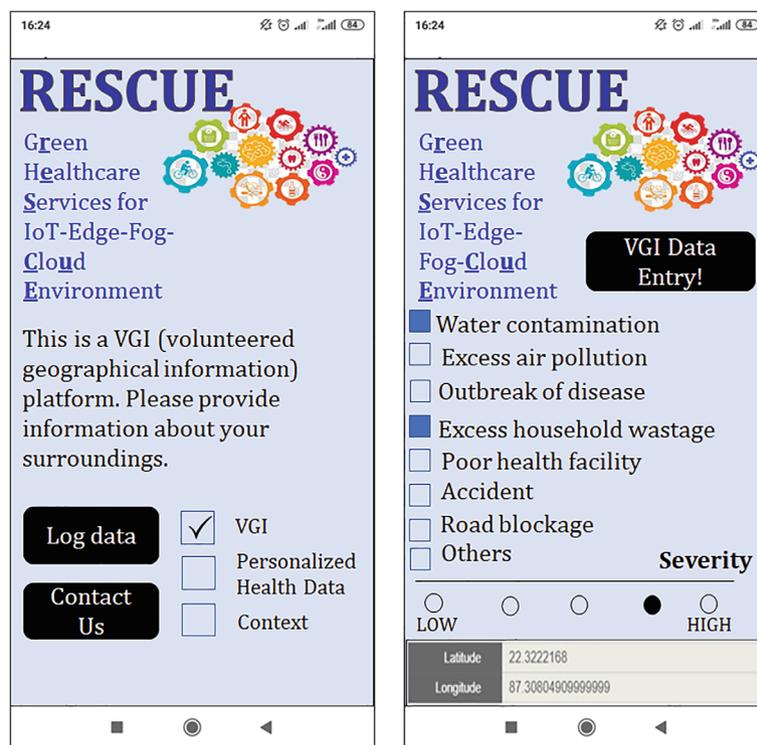


FIGURE 5 Frontend of RESCUE Android application: (A) home page; (B) data collection page

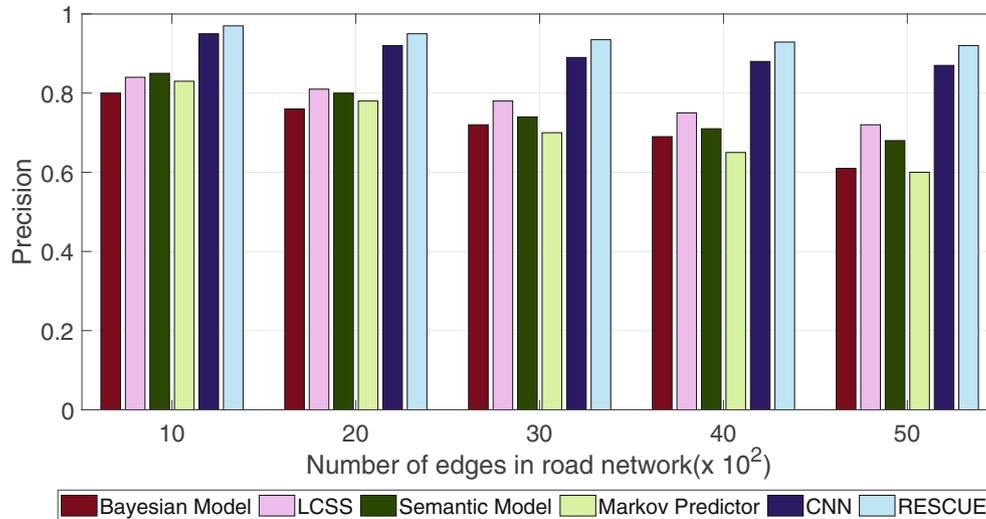


FIGURE 6 Precision value of path-finding algorithm

## 5.2 | Mobility analysis

In this section, we evaluate our proposed framework with five baseline methods to demonstrate the system's efficacy in terms of accuracy and delay of extracting path in the time of exigency. Here, we have used VM with 4 vCPU, 15GB memory of Google cloud platform (GCP), and TensorFlow platform for implementing the proposed path-finding algorithm, and obtaining the results. To illustrate the effectiveness of RESCUE, we have compared the performance with five baseline methods, *Bayesian model*, *LCSS*, *Semantic model*, *Markov predictor*, and *CNN*. We have selected the baselines carefully to justify the fairness of the evaluation. For instance, the Bayesian model is crucial to characterize the influences of varied contexts in path prediction, while LCSS is beneficial for searching and finding common subsequences among a set of paths, and selecting the optimal one. Similarly, the semantic model considers various external contexts that might affect the prediction efficacy, and the Markov chain-based prediction method is one of the strongest modules towards probabilistic models for path prediction. Finally, a convolutional neural network (CNN) helps to incorporate several features like road condition, in-flow, out-flow, weather parameters, and so forth. for finding the optimal path. Therefore, these baselines are justified to showcase the efficacy of our proposed framework, RESCUE.

To illustrate the efficacy of the proposed method, we have designed and executed a large set of experiments on mobility datasets and road-network. We consider a region of 10.8 km<sup>2</sup> with  $16 \times 10^3$  nodes in the underlying road network. We simulated several scenarios where varied segments in the study region become the victim (or affected) region, and unreachable. Our framework finds out the path when these scenarios occur. Figure 6 represents the precision metric of the path-finding module. It is observed that our method achieves 0.98–0.90 precision value with  $10 \times 10^2$  and  $50 \times 10^2$  edges in the road-graph, respectively. The significance of this result is that with a large number of edges, our framework is capable to efficiently extract the path to reach the destination avoiding the blockage or affected regions.

Figure 7 illustrates the runtime of the path-finding algorithm compared to the baseline methods. It is observed that with more numbers of edges, our framework, significantly outperforms others. It shows high accuracy, precision as well as less execution time compared to other existing approaches. Table 3 summarizes the experimental results compared with baseline methods and CLAWER.<sup>27</sup> The accuracy value is measured by the extraction of the most optimal path in the region. Stability represents the robustness of the system. We have measured both the stability and accuracy metrics by simulating ten scenarios and report the average results for all baselines and RESCUE framework. The *learning cost* of the model estimates the time to learn the parameters of the models using the training dataset. Here, the area of the study region has been considered as the input cardinality of the model. The learning cost is categorized into three categories concerning model training time: Low (6–12 min), Medium (13–20 min), and High (above 20 min). It is observed that the neural network model has *high* learning cost. The modeling cost consists of the data preprocessing time, data segmentation, and generating the structure if required. The modeling cost is categorized into three categories: Low (time 0–5 min), Medium

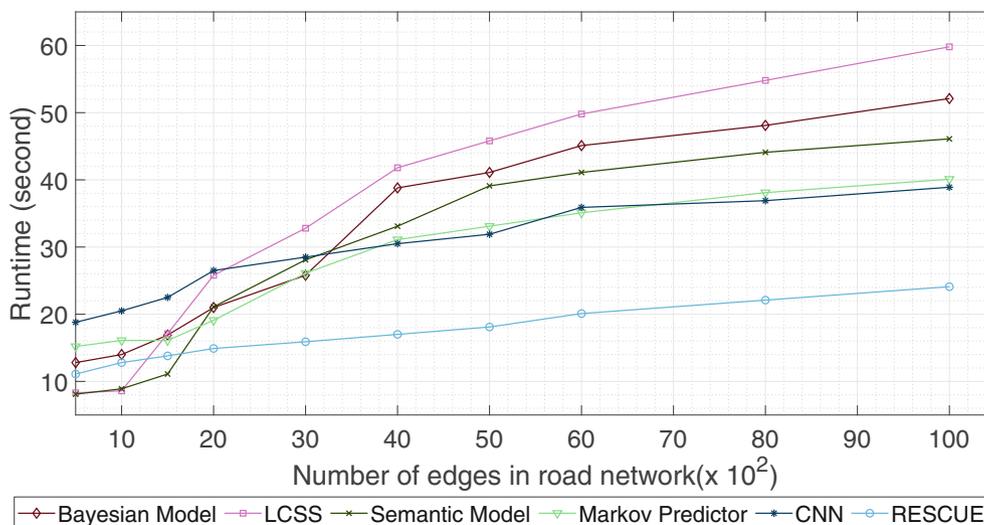


FIGURE 7 Comparison of runtime of path-finding algorithm

TABLE 3 Performance metrics of the proposed mobility analytics module with baseline methods

| Metric        | Bayesian model | LCSS   | Semantic model | Markov model | Neural network model | CLAWER | RESCUE |
|---------------|----------------|--------|----------------|--------------|----------------------|--------|--------|
| Accuracy      | 78.1%          | 74.06% | 79.8%          | 77.71%       | 89.48%               | 85.08% | 93.56% |
| Stability     | 35.8%          | 23.02% | 49.86%         | 30.06%       | 85.09%               | 87.24% | 91.87% |
| Learning cost | Low            | Medium | Low            | Medium       | High                 | Low    | Medium |
| Modeling cost | Medium         | Low    | Low            | Low          | High                 | Low    | Low    |

(6–8 min), and High (above 9–15 min). Here, we have used the categorical values, as different variations of the baseline models may provide different training time or modeling time. Hence, we have considered a range of values. In several aspects, RESCUE has outperformed other approaches to a significant margin. The key reason behind this result is that our framework models the study region in a graph-based structure in a hierarchical manner, and computes the path effectively. RESCUE segments the study area using geo-hash code and computes the risk of selecting a particular route and models it using Markov Decision Process. Further, the offline and online—two-phased process helps to improve the accuracy of the path prediction model by feeding real-time feedback—which is not incorporated in the existing studies and baseline methods. In all aspects, the mobility module of RESCUE outperforms the baseline methods and contributes to assisting users in an emergency.

### 5.3 | Simulation results using iFogSim

The proposed healthcare framework has been simulated in iFogSim.<sup>55</sup> Here, Eclipse IDE has been used for implementation, and JProfiler has been installed and integrated with Eclipse IDE. The proposed healthcare framework has been created in iFogSim, and the respective codes are written, compiled, and executed. The created topology for the proposed healthcare framework is shown in Figure 8.

As observed in Figure 8, there are data sensors (collecting contextual data, mobility-related data, blood pressure data, pulse rate data, body temperature data), and an ECG monitoring module under each edge device. There are six edge devices, which are connected to a fog device. The fog device is connected with the cloud. We have also created the cloud-only healthcare framework in iFogSim, and written, compiled, and executed the corresponding codes.

Four different configurations are considered, presented as follows.

- Config1:- Host storage: 1 GB, Cloud VM with CPU 3 GHz, RAM 4 GB, fog processor 3 GHz, RAM 4 GB
- Config2:- Host storage: 1.5 GB, Cloud VM with CPU 3 GHz, RAM 4 GB, fog processor 3 GHz, RAM 4 GB

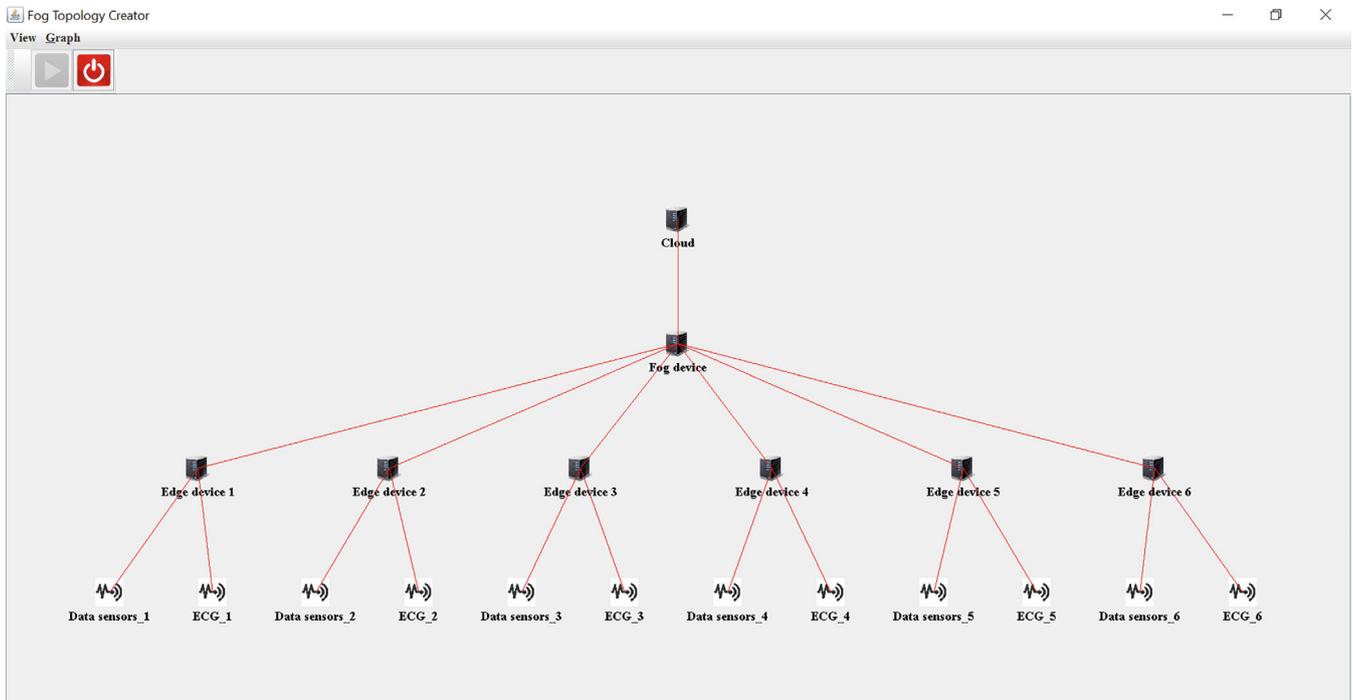


FIGURE 8 Created topology of proposed healthcare framework in iFogSim

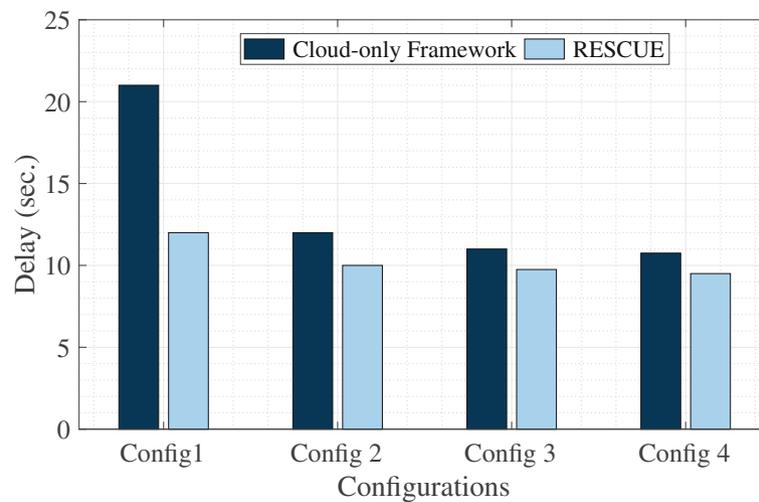


FIGURE 9 Delay in execution of fog based topology and cloud-only topology for e-healthcare

- Config3:- Host storage: 2 GB, Cloud VM with CPU 3 GHz, RAM 4 GB, fog processor 3 GHz, RAM 4 GB
- Config4:- Host storage: 2.5 GB, Cloud VM with CPU 3 GHz, RAM 4 GB, fog processor 3 GHz, RAM 4 GB

The execution delay (in second (s)) in the case of the proposed and cloud-only healthcare framework is monitored and compared. The results are presented in Figure 9. It is noted that the RESCUE has ~10%–40% less execution delay than the cloud-only framework.

#### 5.4 | Visualization of use cases using real-life data

In our experiment, the location is Bankura district, West Bengal, India. We take two layers (health centers and road network) of Bankura district. The layers are presented in Figure 10.

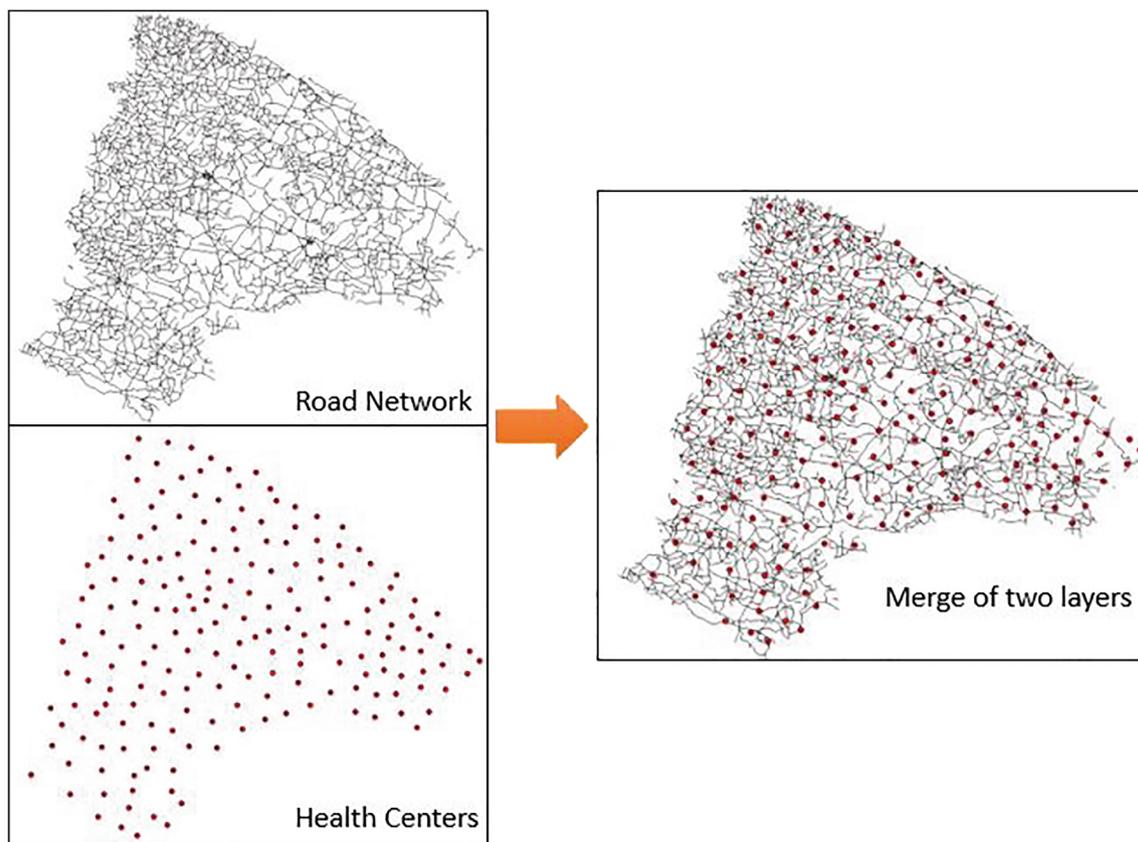


FIGURE 10 Road network and health centers of Bankura district, West Bengal, India

TABLE 4 MRI facilitate hospital details for GQ1

| H.ID | H.latitude | H.longitude | H.facility | H.time      | H.price    |
|------|------------|-------------|------------|-------------|------------|
| MCH  | 87.290     | 23.157      | MRI, ECG   | 11:00–16:00 | INR 12,000 |

Consider a user searching for MRI facilitate hospital details in Bankura district using geospatial query ( $GQ_1$ ) of Section 3.3, where District = “Bankura”. So, the query syntax for the user will be as follow:

```
SELECT H.ID, H.Latitude, H.Longitude, H.facility, H.Time, H.Price
FROM Hospital H
WHERE H.facility='MRI' AND District='Bankura';
```

The details of the geospatial query( $GQ_1$ ) result is presented in Table 4 and pictorially represented in Figure 11. The MRI facility is only available in the Medical College and Hospital(MCH)(87.290, 23.157). Suppose, two users are searching hospitals using geospatial query ( $GQ_2$ ) of Section 3.3, where radius “ $r$ ” = 10 km. Two users’ locations are (86.933, 22.896) and (87.250, 23.248). So, the query syntax for User-1 and User-2 will be as follows:

```
SELECT H.ID, H.Latitude, H.Longitude, H.distance
FROM Hospital H
WHERE Overlap (H.shape, Buffer((86.933, 22.896), 10))=1 ORDER BY H.distance ASC;
SELECT H.ID, H.Latitude, H.Longitude, H.distance
FROM Hospital H
WHERE Overlap (H.shape, Buffer((87.250, 23.248), 10))=1 ORDER BY H.distance ASC;
```

Users’ locations are shown in Figure 12 with navy blue points. Ten kilometers of buffer has been generated surroundings of both the users and shown in Figure 13 with sky blue circles. The details of the hospitals of each user is presented in the Table 5A,5B.

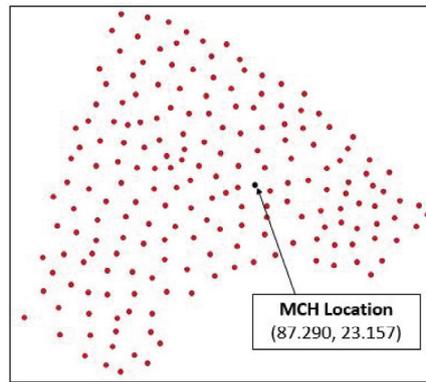


FIGURE 11 Hospital with MRI facility (Bankura district of West Bengal, India)

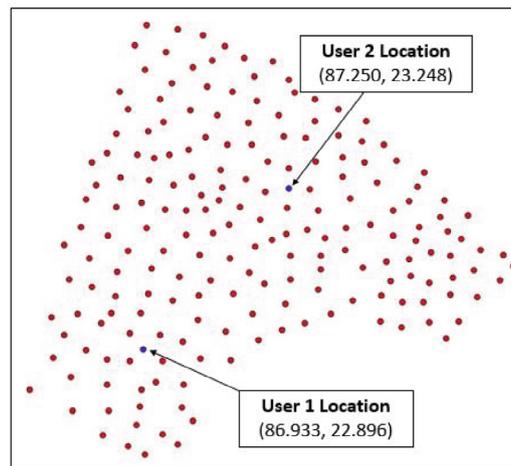


FIGURE 12 Locations of two users

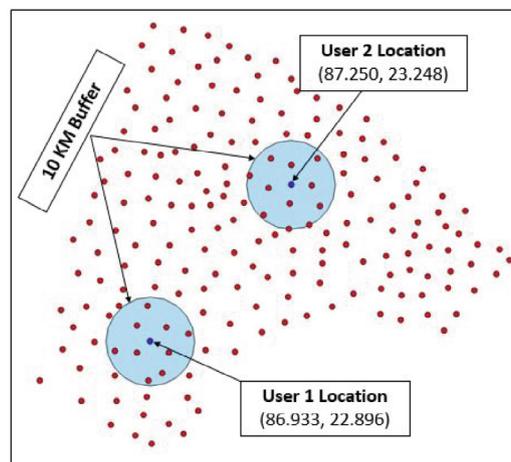


FIGURE 13 Ten kilometers buffer area of user locations

TABLE 5 Hospital details within 10 km radius of user

| H.ID   | H.latitude | H.longitude | H.distance (in km) |
|--|------------|-------------|--------------------|
| (A) Hospital details for User-1 (86.933, 22.896) |            |             |                    |
| BPHC12   | 86.930     | 22.808      | 0.6237             |
| HC06   | 86.928     | 22.977      | 0.7392             |
| HC10   | 86.906     | 22.931      | 3.0229             |
| PC16   | 86.904     | 22.872      | 3.2422             |
| PC17   | 86.902     | 22.933      | 3.4695             |
| SH2  | 86.968     | 22.929      | 3.9140             |
| PC13   | 86.976     | 22.872      | 4.8048             |
| RH09   | 86.979     | 22.826      | 5.1545             |
| PHC4   | 86.854     | 22.874      | 8.8245             |
| (B) Hospital details for User-2 (87.250, 23.248) |            |             |                    |
| RH21   | 87.252     | 23.295      | 0.3365             |
| PC31   | 87.246     | 23.207      | 0.4979             |
| PC24   | 87.244     | 23.148      | 0.8584             |
| MCH  | 87.290     | 23.157      | 4.4938             |
| PC35   | 87.206     | 23.308      | 4.9250             |
| RH25   | 87.298     | 23.248      | 5.3611             |
| HC   | 87.200     | 23.242      | 5.5846             |
| PHC25  | 87.189     | 23.181      | 6.8228             |
| PC33   | 87.311     | 23.306      | 6.8201             |
| BPHC28   | 87.316     | 23.202      | 7.3756             |
| PC40   | 87.163     | 23.270      | 9.7178             |

Abbreviations: BPHC, Block Level Primary Healthcare Center; HC, Health Center; MCH, Medical College and Hospital; PHC, Primary Healthcare Center; PC, Private Healthcare Center; RH, Rural Hospital; SH, Subdivisional Hospital.

## 5.5 | Result analysis of power consumption and latency

In Section 4, the theoretical model of calculating latency and power consumption of the user device has been presented. In this section, we will calculate the same to compare with the cloud-only scenario. The integrated health, movement, and environmental data amount are considered 2.2–3 GB. The data transmission speed of the network is considered as 100–200 Mbps.

Figures 14 and 15 present the latency for detecting health status in case of the indoor and outdoor scenarios, respectively, while using the proposed method and cloud-only scheme. The latency for detecting health status for indoor and outdoor users in the case of a cloud-only scheme<sup>21</sup> is determined to compare with the proposed method. Figures 16 and 17 present the power consumption of the user device, for example, smartphone in case of the indoor and outdoor scenarios during the health status detection period respectively while using the proposed method and cloud-only scheme. The power consumption of the user device during the health status detection period in the case of a cloud-only scheme<sup>21</sup> is determined for indoor and outdoor users to compare with the proposed method. In case of the outdoor region, there is a higher probability of connection interruption due to frequent movement. Thus the latency and power consumption are higher in case of the outdoor region with respect to the indoor region. From Figures 14 and 15, it is observed that the proposed method reduces latency up to 55% for the indoor user and up to 51% for the outdoor user than the cloud-only scheme. From Figures 16 and 17, it is observed that the proposed method reduces power consumption of the user device up to 81% for the indoor user and up to 80% for the outdoor user than the cloud only scheme. In the cloud-only paradigm, data processing, and health status detection take place entirely inside the cloud. But in the proposed framework, the

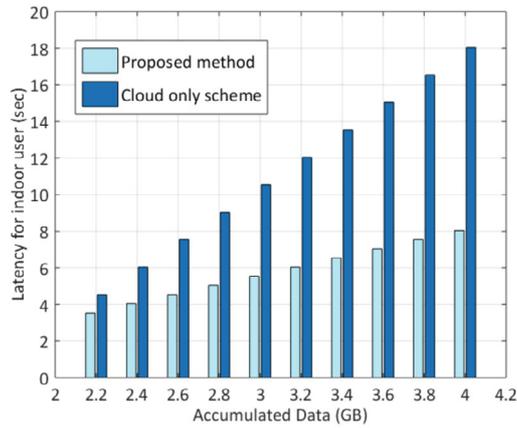


FIGURE 14 Health status detection latency (indoor scenario)

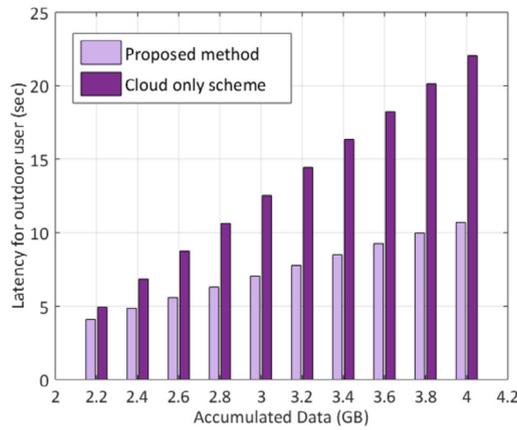


FIGURE 15 Health status detection latency (outdoor scenario)

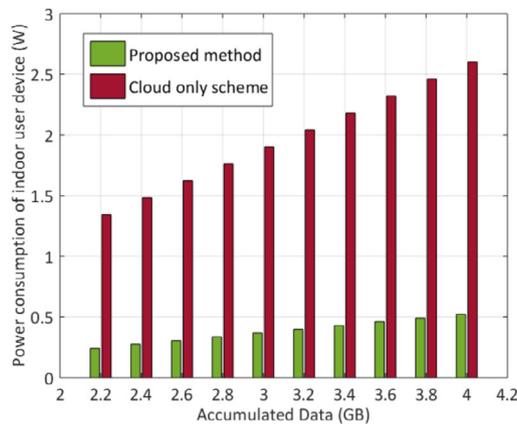


FIGURE 16 Power consumption of user device during health status detection period (indoor scenario)

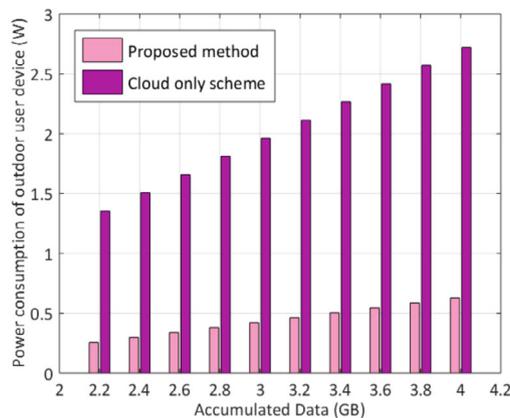


FIGURE 17 Power consumption of user device during health status detection period (outdoor scenario)

SCcNB/fog device/RSU processes the data to detect the health status before forwarding it to the cloud. Here we wish to mention that the mobility data analysis for path prediction is performed inside the cloud. As exhaustive data processing is required for path prediction, cloud is used. But in case of preliminary health data analysis, the amount of processing required is much less in our test case. However, in the proposed scheme after preliminary data analysis based on the result, the data transmission still takes place to the cloud for further storage. As the intermediate node performs preliminary data analysis, the result is obtained faster compared to the use of cloud for the preliminary data analysis. The theoretical analysis shows that the proposed framework provides faster and green health service provisioning to the user.

## 6 | CONCLUSIONS AND FUTURE WORK

This article presents an end-to-end framework, namely RESCUE, which is capable of assisting users in the time of emergency (say, natural disaster) by predicting the routes in the postdisaster scenario when several roads are likely to be affected. The major working modules are (a) accumulating and refining crowd-sourcing data and extracting the correlations among the event and the spatial location, (b) analyzing spatio-temporal traces such as mobility, health data, road information, and so forth and extracting optimal route to reach the destination in minimal time in an emergency, (c) developing hierarchical framework (cloud-fog-edge-IoT) and placement of modules and services for effective outcome, (d) latency and power-aware system design and implementing it for provisioning time-critical applications. RESCUE collects the BAN data consisting of health parameters' data of users and other environmental parameters' data from IoT devices. It analyses the data in the fog nodes, and in case any abnormality is found, the data is sent to the cloud server immediately. Further, RESCUE is conducive to predict routes to users in the time of exigency, avoiding the affected regions in minimal commute time.

While the IoT paradigm enables global connectivity worldwide, communicating with billions of interconnected devices, the whole process's energy consumption is rising steeply. In this regard, *Green IoT computing* has a pivotal role in reducing environmental problems, and subsequently creating a sustainable environment by emphasizing energy-efficient technologies. It reduces greenhouse gas emission and provides a *smarter* and *greener* view of varied applications. To this end, the major aim of this paper is also to reduce energy consumption and contributing towards sustainable urban development. It may be noted that the mobility analysis module is also capable of predicting routes from source to destination, which requires less fuel consumption. The autoencoder process of the mobility analysis module helps in facilitating this feature as well. In all sense, RESCUE provides a *green healthcare and user assistance* framework in the time of exigency.

In the future, we will extend the functionalities of RESCUE incorporating a more advanced energy preserving model like *dynamic voltage and frequency scaling (DVFS)* to maximize the power and energy savings of the computing devices when they are not required. We will also focus on the secure transmission of these health and other contextual data sources collected from the individuals using blockchain technology conserving the privacy of VGI, health, and mobility data.

## ACKNOWLEDGMENTS

The work is partially supported by the research grant from Department of Science and Technology, Government of India, under National Geospatial Chair Professorship scheme.

## AUTHOR CONTRIBUTIONS

**Jaydeep Das:** Conceptualization, Methodology, Investigation, Data curation, Visualization, Writing - original draft, Writing - review & editing. **Shreya Ghosh:** Conceptualization, Methodology, Investigation, Data curation, Software, Writing - original draft, Writing - review & editing. **Anwesha Mukherjee:** Conceptualization, Methodology, Investigation, Writing - original draft, Writing - review & editing. **Soumya K Ghosh:** Conceptualization, Investigation, Writing - review & editing, Supervision, Validation. **Rajkumar Buyya:** Conceptualization, Investigation, Writing - review & editing, Supervision, Validation.

## DATA AVAILABILITY STATEMENT

Data sharing not applicable as no new data generated.

## ORCID

Jaydeep Das  <https://orcid.org/0000-0002-7586-8205>

Shreya Ghosh  <https://orcid.org/0000-0002-6970-8889>

Anwesha Mukherjee  <https://orcid.org/0000-0001-9110-8591>

Soumya K. Ghosh  <https://orcid.org/0000-0001-8359-581X>

Rajkumar Buyya  <https://orcid.org/0000-0001-9754-6496>

## REFERENCES

1. Da Xu L, He W, Li S. Internet of things in industries: a survey. *IEEE Trans Ind Inform.* 2014;10(4):2233-2243.
2. Farooq MS, Riaz S, Abid A, Abid K, Naeem MA. A survey on the role of IoT in agriculture for the implementation of smart farming. *IEEE Access.* 2019;7:156237-156271.
3. Ghosh S, Ghosh SK. Traj-cloud: a trajectory cloud for enabling efficient mobility services. Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS); 2019:765-770.
4. Elazhary H. Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions. *J Netw Comput Appl.* 2019;128:105-140.
5. Rodrigues JJ, Segundo DBDR, Junqueira HA, et al. Enabling technologies for the internet of health things. *IEEE Access.* 2018;6:13129-13141.
6. Mukherjee A, Ghosh S, Behere A, Ghosh SK, Buyya R. Internet of Health Things (IoHT) for personalized health care using integrated edge-fog-cloud network. *J Ambient Intell Humaniz Comput.* 2021;12:943-959.
7. Eldrandaly KA, Abdel-Basset M, Shawky LA. Internet of spatial things: a new reference model with insight analysis. *IEEE Access.* 2019;7:19653-19669.
8. Gilchrist A. *Industry 4.0: The Industrial Internet of Things.* Springer; 2016.
9. Sisinni E, Saifullah A, Han S, Jennehag U, Gidlund M. Industrial internet of things: challenges, opportunities, and directions. *IEEE Trans Ind Inform.* 2018;14(11):4724-4734.
10. Yushi L, Fei J, Hui Y. Study on application modes of military Internet of Things (MIOT). Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE); Vol. 3, May 2012:630-634; IEEE..
11. Zheng DE, Carter WA. *Leveraging the Internet of Things for a More Efficient and Effective Military.* Rowman & Littlefield; 2015.
12. Puliafito C, Mingozzi E, Longo F, Puliafito A, Rana O. Fog computing for the Internet of Things: a survey. *ACM Trans Internet Technol.* 2019;19(2):1-41.
13. Ray PP, Dash D, De D. Edge computing for Internet of Things: a survey, e-healthcare case study and future direction. *J Netw Comput Appl.* 2019;140:1-22.
14. Santos J, Rodrigues JJ, Silva BM, Casal J, Saleem K, Denisov V. An IoT-based mobile gateway for intelligent personal assistants on mobile health environments. *J Netw Comput Appl.* 2016;71:194-204.
15. Solutions CFC. *Unleash the Power of the Internet of Things.* Cisco Systems Inc 2015. [http://www.cisco.com/c/dam/en\\_us/solutions/trends/iot/docs/computing-solutions.pdf](http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-solutions.pdf)
16. Das J, Ghosh S, Ghosh SK, Buyya R. LYRIC: deadline and budget aware spatio-temporal query processing in cloud. *IEEE Trans Serv Comput.* 2021.
17. Vinten-Johansen P, Brody H, Paneth N, et al. *Cholera, Chloroform, and the Science of Medicine: A Life of John Snow.* Oxford University Press; 2003.
18. Asghari M, Sierra-Sosa D, Elmaghraby AS. A topic modeling framework for spatio-temporal information management. *InfProcess Manag.* 2020;57(6):102340.

19. Xiong X, Xiong F, Zhao J, Qiao S, Li Y, Zhao Y. Dynamic discovery of favorite locations in spatio-temporal social networks. *Inf Process Manag.* 2020;57(6):102337.
20. Eom S, Jin X, Lee KH. Efficient generation of spatiotemporal relationships from spatial data streams and static data. *Inf Process Manag.* 2020;57(3):102205.
21. Kaur PD, Chana I. Cloud based intelligent system for delivering health care as a service. *Comput Methods Prog Biomed.* 2014;113(1):346-359.
22. Barik RK, Dubey H, Mankodiya K, Sasane SA, Misra C. GeoFog4Health: a fog-based SDI framework for geospatial health big data analysis. *J Ambient Intell Humaniz Comput.* 2019;10(2):551-567.
23. Khalifa S, Lan G, Hassan M, Seneviratne A, Das SK. Harke: human activity recognition from kinetic energy harvesting data in wearable devices. *IEEE Trans Mob Comput.* 2017;17(6):1353-1368.
24. Goranson C, Thihalolipavan S, de Tada N. VGI and public health: possibilities and pitfalls. In: Daniel S, Sarah E, Michael G, eds. *Crowdsourcing Geographic Knowledge.* Springer; 2013:329-340.
25. Mooney P, Corcoran P, Ciepluch B. The potential for using volunteered geographic information in pervasive health computing applications. *J Ambient Intell Humaniz Comput.* 2013;4(6):731-745.
26. Das J. Geolocation-aware IoT and cloud-fog-based solutions for healthcare. In: Pardeep K, Yugal K, Mohamed AT, eds. *Machine Learning, Big Data, and IoT for Medical Informatics.* Elsevier; 2021:37-52.
27. Ghosh S, Das J, Ghosh SK, Buyya R. CLAWER: context-aware cloud-fog based workflow management framework for health emergency services. 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID); 2020:810-817; IEEE.
28. Das J, Mukherjee A, Ghosh SK, Buyya R. Spatio-Fog: a green and timeliness-oriented fog computing model for geospatial query resolution. *Simul Model Pract Theory.* 2020;100:102043.
29. Ghosh S, Mukherjee A, Ghosh SK, Buyya R. Mobi-iot: mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications. *IEEE Trans Netw Sci Eng.* 2019;7(4):2271-2285.
30. Ghosh S, Mukherjee A, Ghosh SK, Buyya R. STOPPAGE: spatio-temporal data driven cloud-fog-edge computing framework for pandemic monitoring and management; 2021. arXiv preprint arXiv:2104.01600.
31. El Kafhali S, Salah K, Alla SB. Performance evaluation of IoT-fog-cloud deployment for healthcare services. Proceedings of the 2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech); November 2018:1-6; IEEE.
32. Albreem MA, El-Saleh AA, Isa M, et al. Green Internet of Things (IoT): an overview. 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application; 2017:1-6; IEEE.
33. Namboodiri V, Gao L. Energy-aware tag anticollision protocols for RFID systems. *IEEE Trans Mob Comput.* 2009;9(1):44-59.
34. Li T, Wu SS, Chen S, Yang MC. Generalized energy-efficient algorithms for the RFID estimation problem. *IEEE/ACM Trans Network.* 2012;20(6):1978-1990.
35. Lee CS, Kim DH, Kim JD. An energy efficient active RFID protocol to avoid overhearing problem. *IEEE Sens J.* 2013;14(1):15-24.
36. Singh SK, Singh M, Singh D. A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks. *Int J Adv Netw Appl (IJANA).* 2010;2(02):570-580.
37. Chang RS, Kuo CJ. An energy efficient routing mechanism for wireless sensor networks. Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06); Vol. 1, April 2006:5; IEEE.
38. Schurgers C, Srivastava MB. Energy efficient routing in wireless sensor networks. Proceedings of the 2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No. 01CH37277); Vol. 1, October 28, 2001:357-361; IEEE.
39. Singh SK, Singh M, Singh DK. Energy-efficient homogeneous clustering algorithm for wireless sensor network. *Int J Wirel Mob Netw (IJWMN).* 2010;2(3):49-61.
40. Leu JS, Chiang TH, Yu MC, Su KW. Energy efficient clustering scheme for prolonging the lifetime of wireless sensor network with isolated nodes. *IEEE Commun Lett.* 2014;19(2):259-262.
41. Huang J, Meng Y, Gong X, Liu Y, Duan Q. A novel deployment scheme for green internet of things. *IEEE Internet Things J.* 2014;1(2):196-205.
42. Bharti P, De D, Chellappan S, Das SK. HuMAN: complex activity recognition with multi-modal multi-positional body sensing. *IEEE Trans Mob Comput.* 2018;18(4):857-870.
43. Riboni D, Bettini C. COSAR: hybrid reasoning for context-aware activity recognition. *Pers Ubiquit Comput.* 2011;15(3):271-289.
44. Goodchild MF. Citizens as sensors: the world of volunteered geography. *Geo J.* 2007;69(4):211-221.
45. Modaresnezhad M, Iyer L, Palvia P, Taras V. Information Technology (IT) enabled crowdsourcing: a conceptual framework. *Inf Process Manag.* 2020;57(2):102135.
46. Senaratne H, Mobasheri A, Ali AL, Capineri C, Haklay M. A review of volunteered geographic information quality assessment methods. *Int J Geogr Inf Sci.* 2017;31(1):139-167.
47. Martin JP, Kandasamy A, Chandrasekaran K. Mobility aware autonomic approach for the migration of application modules in fog computing environment. *J Ambient Intell Humaniz Comput.* 2020;1-20.
48. Mahmud R, Ramamohanarao K, Buyya R. Latency-aware application module management for fog computing environments. *ACM Trans Internet Technol.* 2018;19(1):1-21.
49. Das J, Dasgupta A, Ghosh SK, Buyya R. A geospatial orchestration framework on cloud for processing user queries. 2016 IEEE International Conference on Cloud Computing in Emerging Markets; 2016:1-8; IEEE.

50. Vales R, Moura J, Marinheiro R. Energy-aware and adaptive fog storage mechanism with data replication ruled by spatio-temporal content popularity. *J Netw Comput Appl.* 2019;135:84-96.
51. Ghosh S, Das J, Ghosh SK. Locator: a cloud-fog-enabled framework for facilitating efficient location based services. Proceedings of the 2020 International Conference on COMMunication Systems & NETWORKS (COMSNETS); 2020:87-92.
52. Rajavel R, Ravichandran SK, Harimoorthy K, Nagappan P, Gobichettipalayam KR. IoT-based smart healthcare video surveillance system using edge computing. *J Ambient Intell Humaniz Comput.* 2021;1-13.
53. Chelladurai U, Pandian S. A novel blockchain based electronic health record automation system for healthcare. *J Ambient Intell Humaniz Comput.* 2021;1-11.
54. Ghosh S, Ghosh SK, Buyya R. MARIO: a spatio-temporal data mining framework on Google cloud to explore mobility dynamics from taxi trajectories. *J Netw Comput Appl.* 2020;164:1-17.
55. Gupta H, Vahid Dastjerdi A, Ghosh SK, Buyya R. iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments. *Softw Pract Exper.* 2017;47(9):1275-1296.

**How to cite this article:** Das J, Ghosh S, Mukherjee A, Ghosh SK, Buyya R. RESCUE: Enabling green healthcare services using integrated IoT-edge-fog-cloud computing environments. *Softw Pract Exper.* 2022;52(7):1615-1642. doi: 10.1002/spe.3078