THE UNIVERSITY OF RHODE ISLAND

University of Rhode Island
DigitalCommons@URI

Department of Electrical, Computer, and Biomedical Engineering Faculty Publications

Department of Electrical, Computer, and Biomedical Engineering

2019

GeoFog4Health: a fog-based SDI framework for geospatial health big data analysis

Rabindra Kumar Barik

Harishchandra Dubey

Kunal Mankodiya
University of Rhode Island, kunalm@uri.edu

Sapana Ashok Sasane

Chinmaya Misra

Follow this and additional works at: https://digitalcommons.uri.edu/ele_facpubs

The University of Rhode Island Faculty have made this article openly available.
Please let us know how Open Access to this research benefits you.

This is a pre-publication author manuscript of the final, published article.

Terms of Use

This article is made available under the terms and conditions applicable towards Open Access Policy Articles, as set forth in our Terms of Use.

Citation/Publisher Attribution

Barik, R.K., Dubey, H., Mankodiya, K. et al. J Ambient Intell Human Comput (2019) 10: 551. https://doi.org/10.1007/s12652-018-0702-x

Available at: https://doi.org/10.1007/s12652-018-0702-x

This Article is brought to you for free and open access by the Department of Electrical, Computer, and Biomedical Engineering at DigitalCommons@URI. It has been accepted for inclusion in Department of Electrical, Computer, and Biomedical Engineering Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons@etal.uri.edu.

Manuscript
Click here to download Manuscript R_FogSDI_AIHC_RKB_HD_ver26.pdf
Click here to view linked References

Noname manuscript No.
(will be inserted by the editor)

GeoFog4Health: A Fog-based SDI Framework for Geospatial Health Big Data Analysis

Rabindra Kumar Barik[1] · Harishchandra Dubey[2] · Kunal Mankodiya[3] · Sapana Ashok Sasane[4] · Chinmaya Misra[1]

the date of receipt and acceptance should be inserted later

Abstract Spatial Data Infrastructure (SDI) is an important framework for sharing geospatial big data using the web. Integration of SDI with cloud computing lead to emergence of Cloud-SDI as a tool for transmission, processing and analysis of geospatial data. Fog computing is a paradigm where embedded devices are employed to increase the throughput and reduce latency at the edge of network. In this study, we developed and evaluated a Fog-based SDI framework named *GeoFog4Health* for mining analytics from geo-health big data. We built a prototype using both Intel Edison and Raspberry Pi for performing a comparative study. We performed a case study on Malaria vector borne disease positive maps of Maharastra state in India. The proposed framework had provision of lossless data compression. Also, overlay analysis of geospatial data could be performed. In addition, we discussed energy saving, cost analysis and scalability of proposed framework for efficient data processing. We compared the performance of proposed framework with state of the art Cloud-SDI in terms of analysis time. Results and discussions showed the efficacy of proposed system for enhanced analysis of geo-health big data generated from a variety of sensing frameworks.

Keywords Spatial Data Infrastructure (SDI) · Geospatial big data · Fog computing · Cloud computing · Geohealth Big Data · Malaria.

1 Introduction

Spatial Data Infrastructure (SDI) has been facilitated by sharing geospatial data by various stakeholders from local to global level. It has created an environment that enables users to retrieve, access and disseminate geospatial data and related meta-data in a secured way [28]. SDI has the capability for storage, decision making on raw geospatial data, thus bringing geospatial data and related maps to a common scale according to the need of the users. It performs querying, superimposition and analysis of geospatial data leading to generation of final reports that could be later used by planners [53]. The Cloud-SDI framework integrated the cloud computing technology with SDI. It was utilized for planning, environmental monitoring, natural resource

E-mail: {rabindra.mnnit@gmail.com (Tel.: +91-8763293589), harishchandra.dubey@utdallas.edu, kunalm@ele.uri.edu, sapanasasane@gmail.com, cmisra@yahoo.com}

[1]KIIT University, India; [2]University of Texas at Dallas, USA; [3]University of Rhode Island, USA; [4] Savitribai Phule Pune University, India.

management, healthcare, land use and urban planning, watershed management, marine and coastal management etc. [29,28,11, 10]. Cloud-SDI have became an emerging area that has the ability to integrate and analyze heterogeneous thematic layers along with corresponding attributes for creation and visualization of various planning scenarios useful for decision making. The user friendliness of Cloud-SDI framework has made it a preferred platform for planning at global, regional, national and local level along with various analysis and modeling functionalities. It integrates common geospatial database operations such as query formation, overlay analysis and statistical computations with unique visualization functionalities [29,19]. These characteristics distinguish Cloud-SDI framework from other geospatial decision support systems. Cloud-SDI is useful for explaining events, designing strategies and predicting outcomes for private and public enterprises [60]. The geospatial data plays an important role in Cloud-SDI as it contains rich temporal and geospatial information [38].

In traditional setup of Cloud-SDI architecture, we send the data to the cloud server where the data is processed and analyzed. This lead to longer processing time in addition to high bandwidth. So, to overcome this problem fog computing comes into the picture. Fog computing provides low-power gateway leading to increased throughput and reduced latency. Consequently, the overall cloud storage is reduced. In addition, reduction in the required transmission power results in overall efficiency. In this work, we process geo-health data at the network edge using proposed Intel Edison, fog computer. This study made the following contributions to health geographic information systems (GIS):

1. Proposed framework improved throughput and latency for efficient analysis and transmission of geo-health data using Intel Edison and Raspberry Pi as Fog devices;
2. Various compression techniques were used for reducing the data size in transmission;
3. Geo-spatial overlay analysis was performed on malaria vector borne disease positive maps of Maharastra state in India from 2011 to 2014 on thick, thin and mobile clients;
4. Analysis of energy saving and computational cost of proposed architecture is performed;
5. Comparison of computation time between the state-of-the-art Cloud-SDI and proposed framework shows improvements over existing works.

2 Related Works

2.1 Spatial Data Infrastructure

In early 80s, many national surveying and mapping agencies had planned the coordination of their activities. They felt the need for a strategy for providing unbound access to geographical information tools and it lead to development of Spatial Data Infrastructure (SDI) [28,37,65,13]. The fundamental components of SDI are people, data, policy, networking, and standards as depicted in Figure 1. These core components come together depending on the nature of interaction among them in SDI framework [48]. The people and data form one category while standards, policy and access network form the other. Policies,

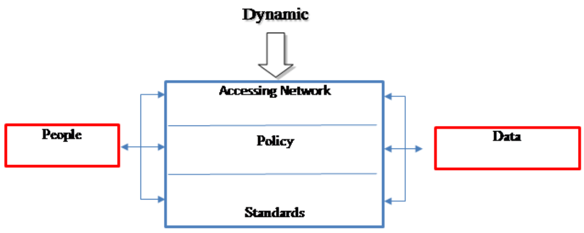

Fig. 1 The dynamic nature of SDI framework. People rely on data and policy, standards and networks have to facilitate the storage, exchange and analysis of data [28].

standards, and access network are dynamic in nature due to the rapid developments in related technologies [45]. This suggested that an integrated SDI has value-added services, end-users and geospatial data along with other important issues related to policies, interoperability and networks. The dynamic SDI framework shares geospatial data at global, national, state, regional and local levels. With the advancement of Service Oriented Architectures (SOA) and cloud computing technology, there is a rapid growth in usage of SDI for geospatial data processing and sharing [10,11]. Confluence of cloud computing technology with SDI led to emergence of Cloud-SDI discussed in next section.

2.2 Cloud SDI Framework

Cloud computing has provided ample storage and computational infrastructure for geospatial data analysis. It has facilitated a transition from desktop to cloud data servers. Cloud computing along with other related web services architectures have created an open environment with shared different variety of assets [60,59,7]. Cloud-SDI framework has delivered a robust platform in organizations that interrelate tools, technologies and expertise to nurture production, handling and use of geographical data. It deployed a unique instance, multi-tenant design that permitted more than one client to contribute assets without disrupting each other. This integrated hosted service method has helped in installing patches and variety of application advancements for the transparency of users. It features geospatial web services as an established architectural methodology [35,32,15]. Cloud platforms uncover the application functionalities through geospatial web services [27,62]. This permits clients to query and update different types of cloud services. It has provisions of a typical tool to assimilate different cloud applications in the software cloud with Service Oriented Architecture (SOA) infrastructure [46,36]. Figure 2 shows the systems' view of Cloud-SDI architecture [26]. In client-tier layer, there are three types of clients namely mobile, thick and thin. Clients could visualize and analyze the geospatial data. Mobile client operates though mobile devices whereas thin client works on standard web browsers. In thin client environments, the clients do not require additional software for operations or data processing. In thick clients environment, users process or visualize the geospatial data on desktop that requires installation of additional software for full-phase operations [4,3]. The application-tier layer is comprised of main geospatial services executed on the servers. It is intermediate between service providers and clients. There are different type of services such as Web Coverage Service (WCS),

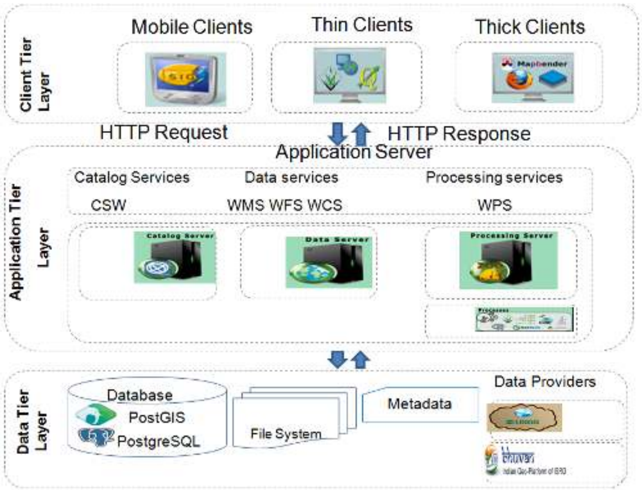

Fig. 2 Cloud-SDI architecture with three types of clients (thick, thin and mobile). It has client-tier layer, application-tier layer with different web services and data-tier layer with geospatial and meta-data storage [26].

Web Feature Service (WFS), Web Catalog Service (CSW), Web Map Service (WMS), and Web Processing Service (WPS) that are operating on top of dedicated servers [46]. These services include three types of server applications, namely data, catalog and processing servers. Catalog server is an important component for data processing in cloud computing architectures. The catalog sever has used to search the meta-data information related to stored data. In catalog service, a standard unique publish-find-bind service model has implemented. This model is defined by the different OGC web service architectures. The data server is dealing with WFS, WCS and WMS [17,33]. The processing server is offering the geospatial processes that allowed different clients to smear the WPS standard geospatial data [57]. The detailed explanation of every process has done by the variety of client request, being forward the desire processing service with the different input of several factors and offers definite region in leaping box and feedbacks with composite standards. Data-tier layer is containing all types of geospatial data and stored in different data formats [8].

System utilizes data-tier layer to manipulate, store, update and recover the geospatial data for long-term analysis and storage. Data providers can store data in different open source database management packages, simple file systems or international organizations i.e. Open Street Map, Bhuvan, USGS, Google, [58,26]. System architecture of Cloud-SDI framework has shown the geospatial data as a key components for data analysis in data-tier layer [52,66,61,39]. It requires geospatial data from the various components. Increasing use of cloud SDI technology for management of geospatial data led to emergence of geospatial big data discussed in next section.

2.3 Geospatial Big Data

Generally, geospatial data has been categorized into raster, vector and graph data. Raster data include geospatial images that are obtained by satellites, security cameras and aerial vehicles. The raster data is provided by different government agencies for using in various analyses. A number of feature can be extracted from these raster data. Change detection and pattern mining are two examples in that data analysis pipeline. Vector data consist of points, lines and polygons features. For examples, in Google map, various landmarks such as temples, bus stops and churches are marked thorough points whereas lines and polygons correspond to the road networks. Geospatial correction pattern analysis and hot spot detection is performed on vector data. Graph data appear in the form of road networks. Here, an edge represents a road segment while a node represents an intersection.

Big data posses diversity, distribution, scale and timeliness that require the use of new technical architectures and analytics to enable business insights. Big data have included data sets with sizes beyond the ability of commonly used software tools to capture, manage and process data within an acceptable time frame [6]. Big data can come in multiple forms. Most of the big data are semi-structured, quasi-structured or unstructured, that requires numerous techniques and tools to prepare such raw data for further processing. Analysis of big data can discover the new correlations to spot business trends, combat crime and prevent diseases etc.. Big data sets are growing rapidly because they are increasingly gathered by the information sensing devices, mobiles, microphones, wireless sensor networks, cameras, aerial images and software logs. Geospatial data were always been big data with the combination of Geographic Information System (GIS), Remote Sensing (RS) and Global Positioning System (GPS) data. Now-a-days, big data analytics for geospatial data are getting considerable attention that allows users to analyze huge amounts of geospatial data [38, 43].

The reliability, manageability and cost are the key factors that made cloud computing attractive for geo-spatial data processing. However, there are some security and privacy concerns for processing of sensitive data using cloud technology. Particularly, in health GIS systems, the medical data is sensitive and demands secure methods for storage and analysis [32]. Thus, for minimizing the privacy and security risks, the data has to be released as per the user context so that limited data access occur within a specific model to prevent unauthorized use of data. Processing geo-health data near the clients using Fog computing adds a security benefit as now only clinical features and/or analysis reports are sent to the cloud. We describe the Fog computing in next section.

2.4 Fog Computing

Fog computing was coined by Cisco in 2012 [21]. It is a framework that complements the cloud computing for decentralizing the resources in data centers towards users for improving the quality of service and user experience [49]. However in fog computing framework, processing of different services are not only concentrated in cloud data centers [44] [51] [56]. Data computation and storage could be brought closer to the users that lead to reduced latencies and communication overheads with remote cloud servers [63, 14, 20]. It refers to a computing paradigm that uses interface kept close to the devices that acquire data. It introduces the facility of local processing leading to reduction in data size, lower latency, high throughput and power efficiency of the cloud-based systems. Fog computing technology has been successfully implemented in smart cities [31] and healthcare [23, 22, 42, 49, 50, 51]. Fog devices are embedded computers such as Intel Edison or Raspberry Pi that acts a gateway between cloud and mobile

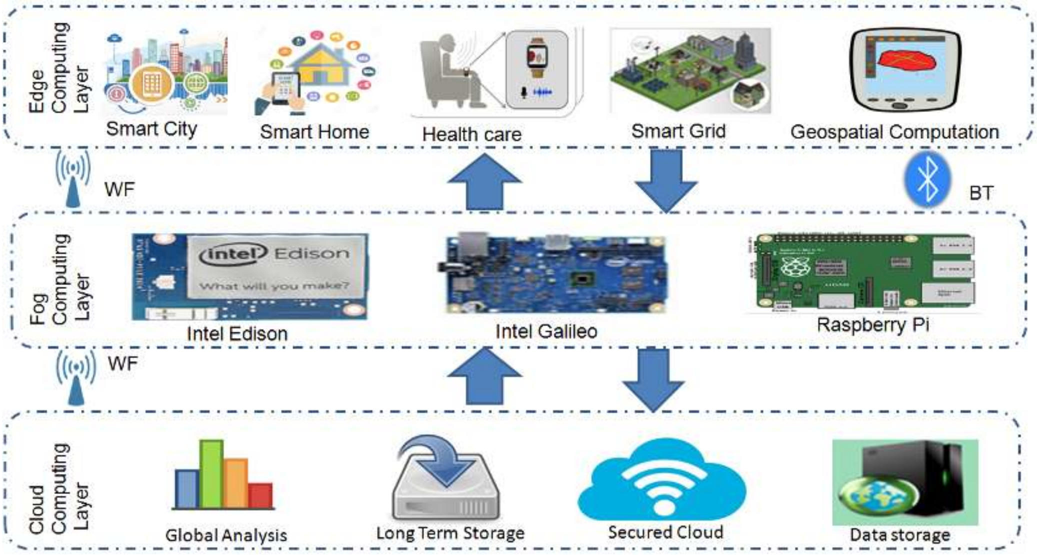

Fig. 3 Fog computing as an intermediate layer between edge and cloud. Fog layer enhanced the efficiency by providing computing near the edge devices. This framework is useful for geospatial application,healthcare, smart city, smart grid and smart home etc. [49, 9].

devices such as smart phones and mobile GIS interfaces [9, 67, 47, 41, 12, 54]. We consider this technology for integrating it with health GIS for better management of geospatial health data. Fog computing helps to reduce latency and increase throughput at the edge of different clients in cloud computing environment (See Fig. 3). The next section describes health SDI approaches for sharing geo-health data.

2.5 GeoHealth SDI

Particularly, for health sector, disease data sharing is a significant issues with respect to collaborative preparation, recovery and response stages of numerous disease control mechanism. Disease phenomena are strongly associated with geospatial and related temporal factors. For tackling these situation, Cloud-SDI framework provided dynamic and real-time approach to represent disease information through the maps on common browsers [55]. However, data integration, interoperability, data heterogeneities and cartographic representation are still major challenges in Cloud-SDI framework for health applications. Such barriers in extensively sharing geospatial health data restrain the effectiveness in understanding and responding to disease outbreaks. For overcoming such challenges in health SDI, sharing and mapping of geospatio-temporal disease information in an inter-operable framework based on OGC specifications under fog computing environment is the need of the hour [30].

From the above related work, it is clear that, it requires an efficient, reliable and scalable fog computing based SDI framework for sharing and analysis of geospatial big data across the web. Next section describes the proposed architecture based on fog SDI framework. Various approaches like compression techniques, overlay analysis, energy saving scheme, scalability issues, time analysis are discussed with respect to the geospatial data of malaria vector borne disease positive maps of Maharastra, India from 2011 to 2014.

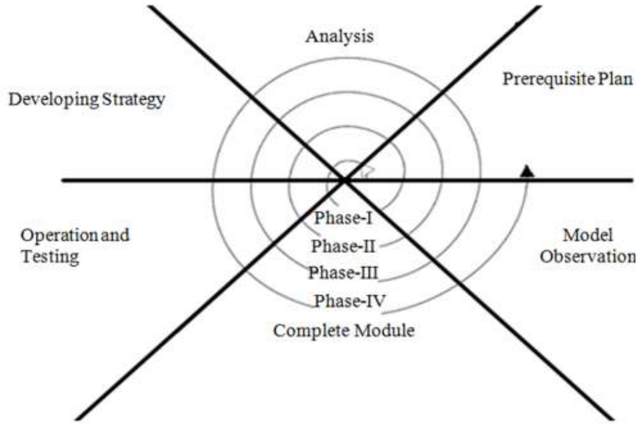

Fig. 4 Spiral process model for the development of *GeoFog4Health* starting from requirement stage, analysis, designing, coding, testing and complete framework observation.

3 Proposed Framework

For developing the prototype of *GeoFog4Health* i.e. Fog-based SDI framework, the primary emphasis is on Object Oriented Software Engineering (OOSE) method that involves the configuration of models that capture the real world actors of the system and their unique behavior for each of the design stages with conglomerate the time critical nature and strong user focus [53]. This is a usual way to explain the environment in that the system and its actors would be groomed in an evolving manner. This has helped in reducing the semantic gap between the developed framework and the real world applications. Figure 4 has shown the complete spiral model for development and implementation of *GeoFog4Health*. In OOSE approach, the software development process adopts a sequence of steps including requirements prerequisite plan, analysis, development strategy, operation and testing, complete module and framework observation. The process is incremental in nature and each of implementation phase was refined with the analysis and developing stages through evaluation and testing of a completed module. Further, the incremental development strategy of the proposed framework that has allowed the constructing the framework to be tackled in smaller parts that are more controllable portions with increasing complexity. So, there are different modules defined in *GeoFog4Health*. Table 1 presents the various prototype implementations with functionality description. In addition, it is expected that each phase would reveal a unique features related to the requirements of infrastructure and enable exploration of the interfaces between fog SDI framework components. The requirements stage of application design aims to specify the behavior of the framework from a user's perspective [53]. From the above defined four phases, next section describes the details of geo-health database for malaria vector borne disease positive maps of Maharastra, India from 2011-2014.

Table 1 Implementation of proposed architecture.

Phases	Function Description
I	Geospatial database for malaria vector borne disease positive maps of Maharastra, India from 2011-2014
II	Proposed *GeoFog4Health* framework and Lossless compression with overlay analysis of the geospatial data on thin and mobile client environment
III	Energy saving, cost analysis scheme and scalability issues for *GeoFog4Health* framework
IV	time analysis and comparison analysis for Cloud-SDI and *GeoFog4Health* framework

3.1 Malaria vector borne disease positive maps of Maharastra, India

Maharashtra is a state in the western region of India and the second most populous state and third largest state by area in India. Maharashtra is bordered by the Arabian Sea to the west, Karnataka to the south, Gujarat and the Union territory of Dadra and Nagar Haveli to the northwest, Telangana to the southeast, Madhya Pradesh to the north, Chhattisgarh to the east and Goa to the southwest. This state covers an area of 307,731 km2 that accounts for 9.84% of the total geographical area of India. There are 41000 villages and 378 urban centers in Maharashtra. Maharashtra has one of the highest levels of urbanization among all Indian states. The secondary health data positive cases of malaria and number of death due to malaria are collected from the National Vector Borne Disease Control Program (NVBDCP), New Delhi. Climatic data includes all the surface parameters like temperature, rainfall, humidity, wind speed etc. are collected from the National Data Centre (NDC) and India Meteorological Department (IMD), Pune, India.

The inputs of positive cases and the deaths (number of persons) were fed in Quantum GIS software and region wise maps with district boundaries were generated. It includes incidence of malaria with the interval of 2011-2014 to see the trends and patterns of the incidence of Malaria in Maharashtra. Finally find out the trend with help of linear regression equation: $y = a + bx$ where b value shows the rate of change per decade. In this way, the trend of malaria from 2011 2014 is generated in form of a Map. Death due to malaria from 2011-2014 is depicted in Figure 5. The creation of geospatial database are significant and tedious assignment with respect to efficacy of SDI development and implementation. Integrated geo-health database creation include stages such as input data such as geo-health and related attributes data, its authentication by connecting with same set of data. Geospatial database delivers a platform in that organizations interrelate with technologies to nurture actions for spending, handling and generating geo-health data. The development of geo-health database is supported in various administrative and political levels through decision-making functions. Quantum GIS 2.14.3 is the OS GIS software selected to examine the competences with respect to creation of geospatial database. The procedure model of geo-health database creation is frequent or recurring in nature. Each operation improved the study and strategy steps through assessment and testing of a complete module

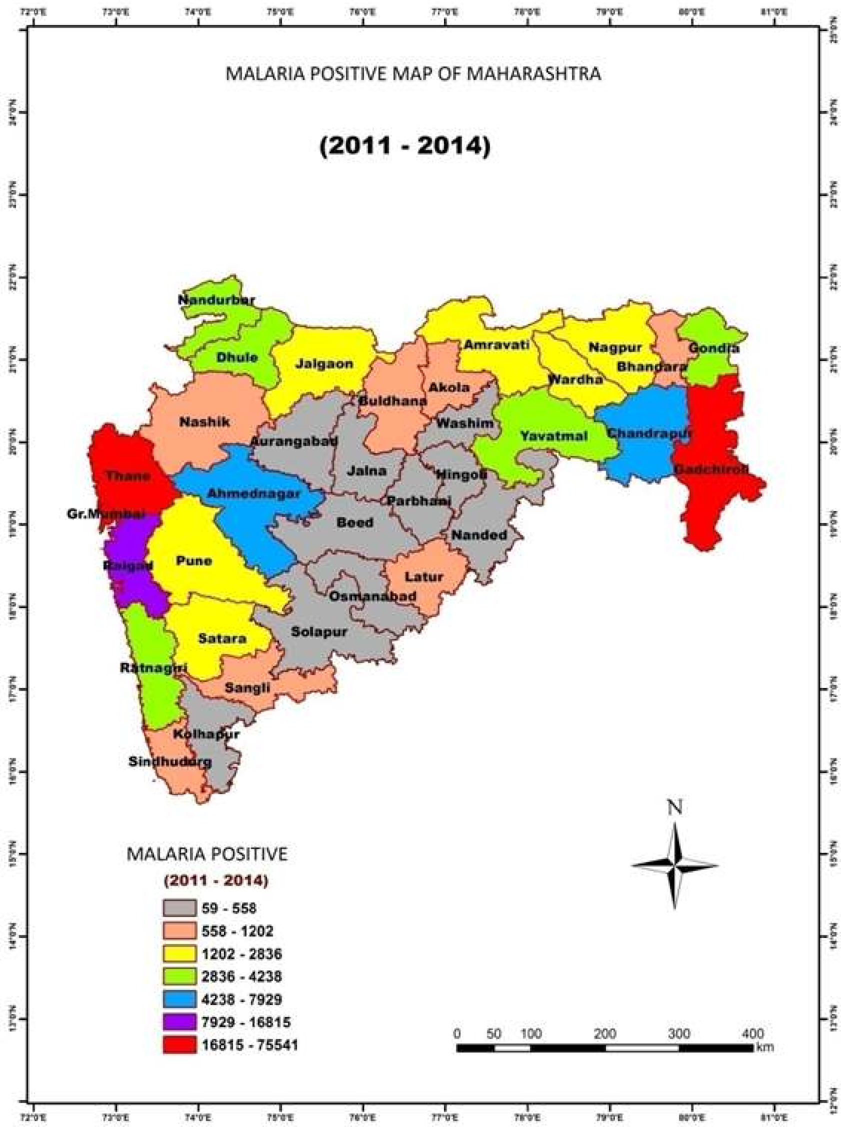

Fig. 5 Malaria positive maps of Maharastra, India from 2011-2014.

component. In module components, Open source Quantum GIS has set up malaria geo-health database using political map of India. Quantum GIS is also used for integrated geo-health database creation. After geo-health database is created, there is a need for process model that could perform accurate analysis of geo-health data.

3.2 Proposed Architecture

This section describes various components of the proposed *GeoFog4Health* framework and discusses the methods implemented in it. The main components are hardware, software and methods used for compression of geospatial big data. We employed Intel Edison and Raspberry Pi as fog computing device in proposed *GeoFog4Health* architecture [9]. Intel Edison is powered by a rechargeable lithium battery and contains dual-core, dual-threaded 500MHz Intel Atom CPU along with a 100MHz Intel Quark micro controller. It possess 1GB memory with 4GB flash storage and supports IEEE 802.11 a,b,g,n standards. It connects to WIFI

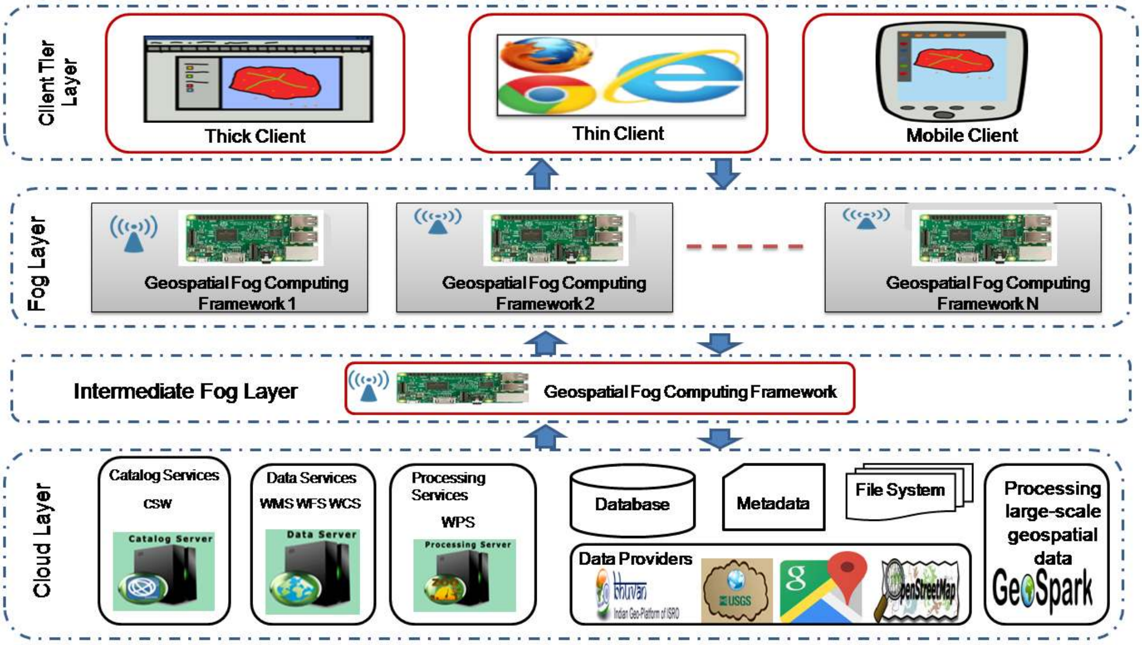

Fig. 6 Conceptual diagram of the proposed *GeoFog4Health* architecture with four layers (client-tier layer, fog layer, intermediate fog layer and cloud layer).

and used UbiLinux operating system for running compression utilities. Raspberry Pi B Platforms have been used. Raspberry Pi consists of a 900MHz 32-bit quad-core ARM Cortex-A7 CPU with 1GB RAM. For WiFI connectivity in Raspberry Pi, it has been used WIFI dongle of Realtek RTL8188CUS chip set. In the proposed framework, we used both Intel Edison and Raspberry Pi in every fog node [20] for better efficiency in time analysis discussed in later sections.

Figure 6 shows the proposed *GeoFog4Health* framework that consists of four layers namely, cloud, fog, intermediate fog and client-tier layer. Cloud layer is mainly focused on overall storage and analysis of geo-health data. In cloud layer, we implemented GeoSpark [64] for real time geospatial big data processing on the top of Hadoop Ecosystem.

Fog SDI layer works as middle tier between client-tier layer and intermediate fog SDI Layer. It has been experimentally validated that the fog SDI layer is characterized by low power consumption, reduced storage requirement and overlay analysis capabilities. In fog SDI layer, all fog nodes were developed with Intel Edison and Raspberry Pi processor for geo-health data analysis. Additional intermediate fog SDI layer were added between fog SDI layer and Cloud-SDI layer for reducing load overhead in fog SDI layer. Thus, intermediate fog SDI layer were used for refinement of processing and temporary storage of geo-health data. In client-tier, there are three categories of users namely, thick, thin and mobile clients respectively. Processing and analysis of geo-health data can be done using any of these clients types. In the proposed framework, the processing at each fog node consume energy. We realized that energy should be properly managed. We experimented different overlay analysis and lossless compression techniques within proposed *GeoFog4Health* framework.

Table 2 Result of compression in proposed framework using malaria geo-health data.

Geo-health Data	Original Data Size (MB)	.rar Compressed Size (MB)	.gzip Compressed Size (MB)	.zip Compressed Size (MB)
India boundary	2.96	2.5	2.2	1.2
Death Mapping	.98	.55	.41	.32

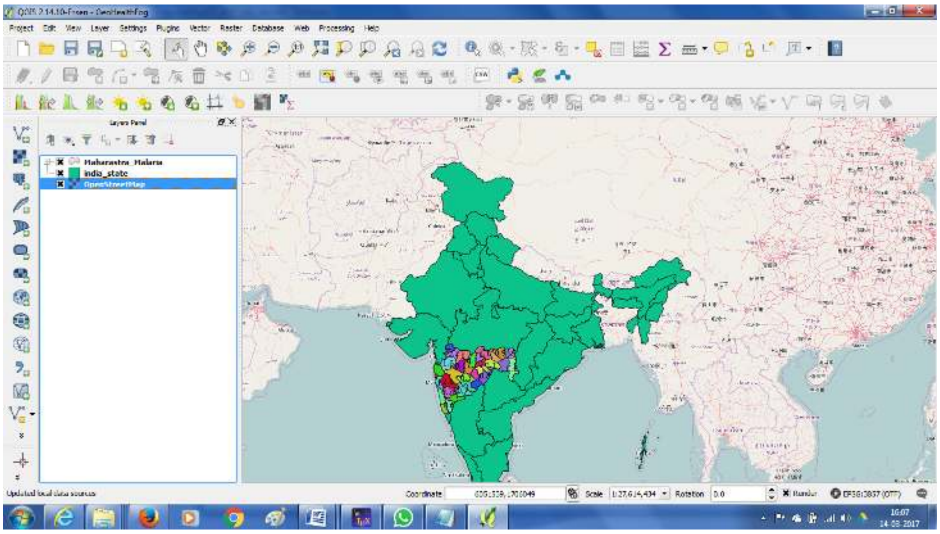

Fig. 7 Integrated geo-health database of Malaria.

3.3 Data Compression and Overlay Analysis

In proposed framework, we used well known popular compression algorithms for reduction of data size. The concept of data compression was used in several areas such as network and mobile SDI framework [68,16,34]. In present study, we translated various compression techniques from mobile platform to the proposed framework [44]. After completion of compression module, resultant compressed data at fog layer is transmitted to the cloud. Cloud layer could store the compressed data or decompress the data before processing, analysis and visualization. We used only lossless compression techniques such as .rar, .gzip, .zip *etc.*. Various lossless compression techniques applied at fog Layer lead to results summarized in Table 2. Overlay analysis was performed for malaria vector borne disease positive maps of Maharastra, India. Overlay analysis is a data analysis technique that superimposed various geospatial data in a unique platform for better analysis of vector and raster geospatial data. In the present sound, we found two shape files related to malaria information. Again, these two shape files were overlapped with Google satellite layer. We used the malaria death mapping data of Maharastra from 2011-2014 was processed in *GeoFog4Health*. Overlay analysis of various raster and vector data of particular areas were performed. Initially, the developed geospatial datasets are opened with Quantum GIS and performed some join-operations with the help of these datasets [7]. Desired overlay operation was done with standalone application and refereed as thick client operation as shown in figure 7. Figure 7 visualizes the Open Street maps with other two shape files. These files are opened in Quantum GIS desktop environment. QGISCloud plugin was

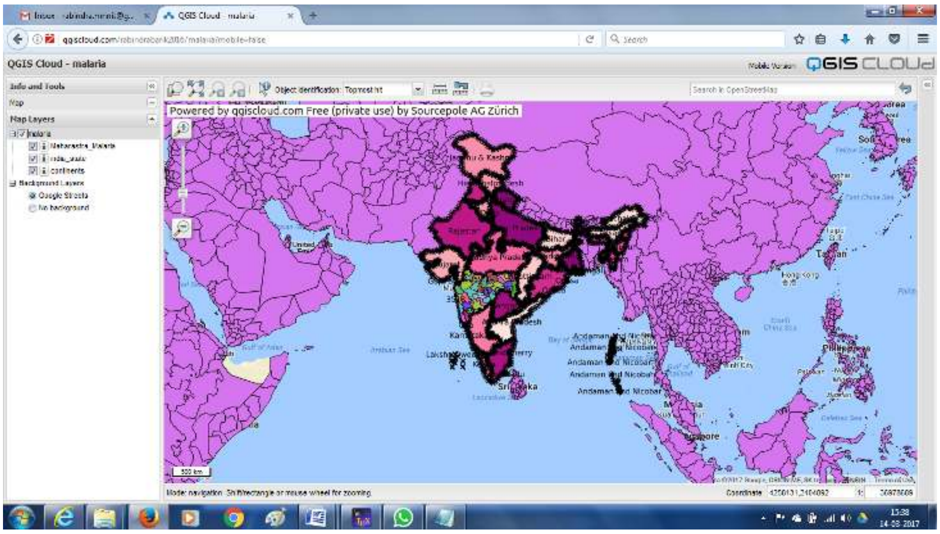

Fig. 8 Overlay operation on thin client environment in QGIS cloud [5].

installed and added in Quantum GIS desktop environment. This QGISCloud plugin has the unique capability of storing various vector and raster data set. This plugin was linked with the cloud database for storing and performing overlay analysis. After storing in desired cloud database, it generated the thin and mobile client link for visualization of both raster and vector data. Figure 8 and Figure 9 shows the overlay analysis on thin and mobile clients respectively. In this way, the overlay analysis is an useful and simple technique for geo-health data visualization. Next section describes better strategy for energy efficiency and management in *GeoFog4Health* framework.

3.4 Energy Efficiency

In this section, an analytical model was introduced for the energy saving management of intermediate fog layer in *GeoFog4Health*. Proposed framework investigated the energy saving management using finite buffer batch service buffering system that can change over time and multiple vacations. We studied that the overall message delay in the uplink channel and performance of mean number of data packets in the buffer, buffering delay and probability of blocking in the fog layer. Lots of energy is required for handling heavy traffic of fog node data from fog and intermediate fog layer. With vacation mode operation, intermediate fog layer node does not listen to the node of fog layer continuously but it alternates the active state and the vacation state. It has considered a finite buffer batch service buffering system with multiple vacation and changeover time.

Let, it has assumed a and b as the threshold values of activating the intermediate fog layer service and service capacity, respectively. Whenever the intermediate fog layer node finished all its work, it goes to vacation, an internal timer that is exponentially distributed with parameter θ is then started and the intermediate fog layer node awakes to check the buffer content of the fog layer. When upon awaking the intermediate fog layer finds that there are still less than $j(0 \le j \le a-2)$ data frames, it goes to vacation again. If the number of data frames in the buffer of the fog layer is $a-1$ either at a service completion epoch or at a vacation termination point, the intermediate fog layer service will wait for some more time that is called changeover

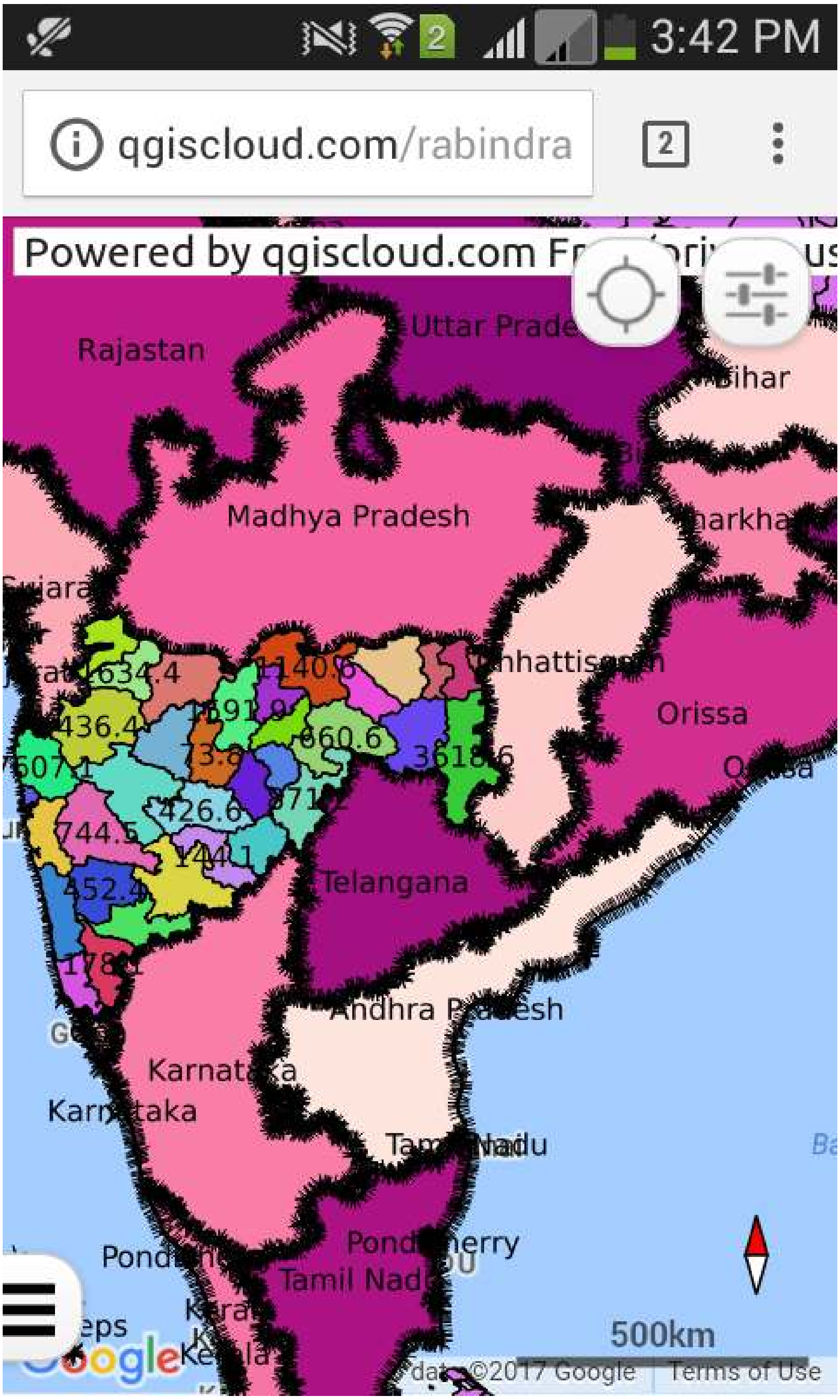

Fig. 9 Overlay operation on mobile client environment in QGIS cloud [2].

time. The changeover time is exponentially distributed with parameter γ. If there is an arrival during the changeover time, the intermediate fog layer service will start immediately, otherwise, it will go for a vacation period. If after a vacation period, the intermediate fog layer finds a non-empty buffer, it serves all data frames present at that point and also all new data frames that arrive while the intermediate fog layer service is working, until the buffer becomes empty again at the fog layer end and the whole procedure is repeated.

3.4.1 Analytical Model

It has considered a Markov chain with the state space $\{(i,j)|0 \le i \le N, j = 0,1 \bigcup (a-1,2)\}$ where i gives the buffer size and j represents the state of the server. The process is in the state $(i,0)$ if there are i data frames waiting in the buffer and the server is in sleep mode. It is in state $(i,1)$ if there are i data frames waiting in the base station buffer and the mobile station service unit is busy and it is in state $(a-1,2)$ if there are $a-1$ data frames in the buffer and the server is waiting in the system. Using probabilistic argument at steady state, we obtain the following system of equations

$$\beta P_{0,0} = \alpha P_{0,1}, \tag{1}$$

$$\beta P_{i,0} = \beta P_{i-1,0} + \alpha P_{i,1}, \; 1 \le i \le a-2, \tag{2}$$

$$(\beta+\theta) P_{a-1,0} = \beta P_{a-2,0} + \gamma P_{a-1,2}, \tag{3}$$

$$(\beta+\theta) P_{i,0} = \beta P_{i-1,0}, \; a \le i \le N-1, \tag{4}$$

$$\theta P_{N,0} = \beta P_{N-1,0}, \tag{5}$$

$$(\beta+\alpha) P_{0,1} = \beta P_{a-1,2} + \alpha \sum_{s=a}^{b} P_{s,1} + \theta \sum_{s=a}^{b} P_{s,0}, \tag{6}$$

$$(\beta+\alpha) P_{i,1} = \beta P_{i-1,1} + \theta P_{i+b,0} + \alpha P_{i+b,1}, \; 1 \le i \le N-b, \tag{7}$$

Using normalization condition $\sum_{i=0}^{N} P_{i,0} + \sum_{i=0}^{N} P_{i,1} + P_{a-1,2} = 1$ we recursively solved the equations

3.4.2 Performance Measures

The state probabilities of the incoming job request at arrival times are known, we can find out various performance measuring parameters like average number of job requests in the buffer L_q , average time spending in the buffer W_q and the probability of blocking (PBL). They are given by $L_q = \sum_{i=1}^{N} iP_{i,0} + \sum_{i=1}^{N} iP_{i,1} + (a-1)P_{a-1,2}$. The probability of blocking is given by $PBL = P_{N,0} + P_{N,1}$. The average time spending in the buffer using Little's rule is $W_q = L_q/\beta'$, where$\beta' = \beta(1-PBL)$ is the effective arrival rate.

3.5 Cost Analysis

In this section, it has been determined that the expected cost per unit time and optimize the threshold values for activating the server (a), batch service capacity (b) and service rate (μ) for downloading the data frame, so that the expected cost function can be minimized. Here it has used the genetic algorithm to find out the minimized expected cost.

Let F be the total expected cost per unit slot. Using the definitions of each cost element and its corresponding system characteristics, it has

$F = C_1 L_q + C_2 P_b + C_3 \beta PBL + C_4 \alpha + C_5 \gamma + C_6 \theta$;

C_1= the cost of each slot for every incoming frame waiting in the base station buffer,

C_2=fixed cost per slot when the base station buffer is blocked,

C_3= fixed cost for each lost data frame when the base station is blocked,

C_4= the transmission cost per slot when the mobile station is busy,

C_5= fixed cost per slot when the mobile station is in change over time,

C_6= fixed cost per slot when the mobile station is on sleep.

Among different technique used for optimizing, the genetic algorithm (GA) is an efficient optimization technique for find the value based on process of natural selection process. This algorithm has implemented particular rule to minimize the parameter based on some fit value. This algorithm was implemented by Holland in 1975. Some of the advantages of a GA has defined below:

1. Provide efficient, effective techniques for optimization mainly in scientific and engineering application.
2. It is not using conventional derivative calculation for finding out the cost function.
3. There are many ways to speed up and improve a GA based application at the same time can search from large sampling space.
4. Inherently parallel; easily distributed and can accommodate many number of variables.
5. Optimizes the variables values with highly complicated manner. In this algorithm best is not always picked, and worst is not necessarily excluded.
6. Causes movement in the search space and and provides more than one optimum values. Restores lost information to the population.
7. The optimization can be done by the genetic algorithm on encoded variables.
8. Gives satisfactory performances for engineering, scientific research and machine learning application.
9. Obtain the fitness value to determine solutions and no complicated mathematical computations are used. Mixture of greedy exploitation and adventurous exploration.

In traditional methods have lots of disadvantages. Genetic algorithm overcome few of this and provide significantly improved performance. Here the strings are mentioned in binary values and the bit value 0 and 1 represent the gene. The fitness value is generated by the associated function and constraint checking has done.

From the above analysis, it has been found that suitable mathematical model has required for efficient energy management is the need of the hours. But whenever, it will talk about processing of huge amount of real time data processing in *GeoFog4Health*, it is required high batch processing infrastructure that has been discussed in the next section.

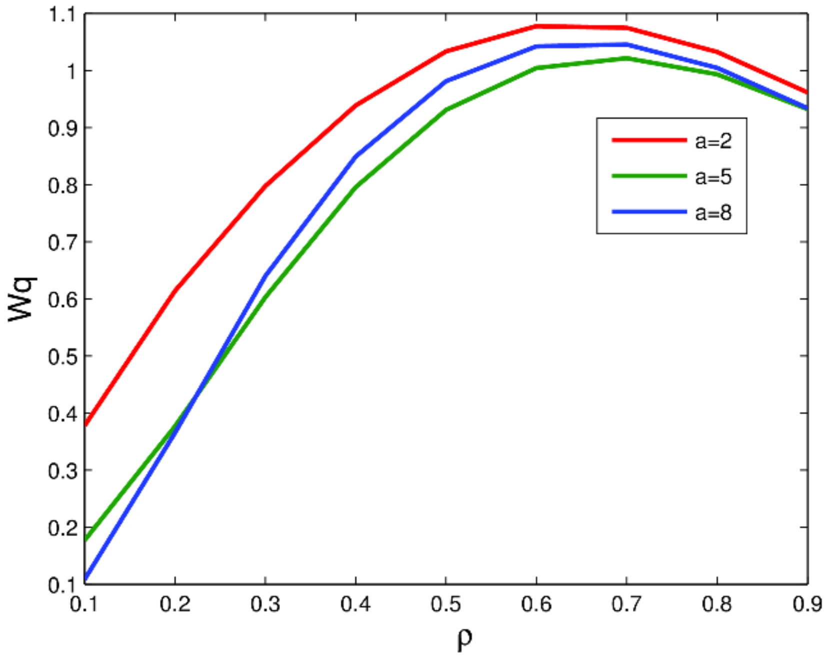

Fig. 10 Effect of (ρ) on W_q with varying a .

3.5.1 Numerical Results

The main objective of this section is to demonstrate the relation between the different system parameters. Figure 10 shows the effect of utilization factor$(/rho)$ on the average number of packets waiting in the buffer or queue length (W_q) for different values of a. From Figure 10, it can observe that for all values of a the W_q increases as utilization factor (ρ) increases and for higher utilization factor delay in the buffer have reduced. We have observed that for higher value of a more buffer delay in the system. The effect of buffer size(N) on loss probability for vacation and non-vacation is considered in Figure 11 . We observe that loss probability monotonically increases with the increase of buffer size. Further, the loss probability in case of vacation is slightly higher than the one obtained in case of non-vacation. It also illustrates dependence of the average waiting time on θ and γ. We observe that for fixed service rate the average waiting time decreases as the arrival rate θ increases. Further with fixed θ it increases when the service rateγ increases. Hence we can setup an admissible arrival rate and the sufficient service rate in the system in order to have lower average waiting time.

Table 3 and 4 establish the impact of a and b on the cost function for different values of ρ, respectively. From Table 3, it can visualize that the minimum expected cost first decreases, again it increases as ρ increases, for fixed value a. But for fixed ρ, the minimum expected cost increases as a increases. Similarly, in Table 4 the minimum expected cost decreases as batch size (b) increases, for fixed ρ. The minimum expected cost first decreases, again it increases as ρ increases, for fixed batch size b. Here it has implemented the experiment by considering the batch size (b) in the range of 8 to 17 and utilization factor (ρ) from 0.1 to 0.9. We have seen that the lowest optimum cost is 459.44 at $\rho = 0.5$ and batch size ($b = 17$).

Figure 12 presents the number of iteration effect on the cost function. We find that average cost is more than minimum value

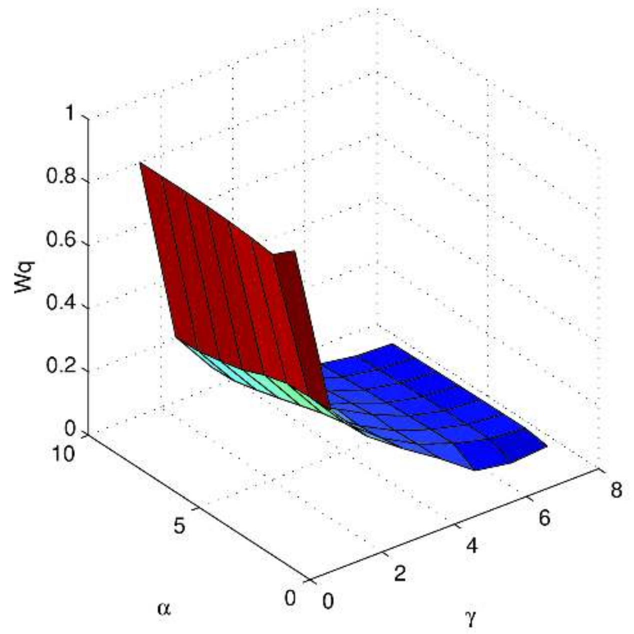

Fig. 11 Impact of γ and α on (W_q).

Table 3 The optimal values a^*, F^* for various values of ρ.

ρ	a							
	3	4	5	6	7	8	9	10
0.050000	**1873.872**	1886.680	1894.568	1897.380	1898.142	1898.263	1898.264	1898.271
0.100000	**1021.619**	1043.866	1071.289	1095.680	1112.885	1119.275	1119.339	1119.405
0.200000	**642.155**	682.434	728.687	780.731	837.031	874.026	875.247	876.578
0.300000	**539.406**	589.843	640.612	694.233	751.928	796.108	798.399	800.897
0.400000	**498.252**	549.769	599.219	649.409	702.090	746.139	748.970	752.037
0.500000	**485.287**	533.485	579.476	625.593	673.451	716.289	719.398	722.745
0.600000	**490.683**	533.916	575.690	617.734	661.351	702.648	705.899	709.383
0.700000	**509.445**	547.391	584.812	622.926	662.724	702.220	705.527	709.064
0.800000	**538.259**	571.186	604.427	638.827	675.128	712.605	715.910	719.444
0.900000	**574.741**	603.147	632.529	663.480	696.558	731.869	735.131	738.621
1.000000	**617.134**	641.575	667.474	695.265	725.380	758.452	761.640	765.059

for all iterations.

3.6 Scalability

Scalability is the ability of proposed *GeoFog4Health* architecture to handle a growing amount of geospatial big data for analysis and visualization. In fog layer, it gives the horizontal scalability for processing of large amount of geospatial data. It also keeps tracks of the proposed framework in cloud layer, it has implemented GeoSpark for scalability process within Hadoop Ecosys-

Table 4 The optimal values b^*, F^* for various values of ρ.

ρ	b									
	8	9	10	11	12	13	14	15	16	17
0.1	2086.34	1869.01	1695.75	1554.48	1437.16	1338.25	1253.82	1181.05	**1117.92**	1063.02
0.2	1117.44	1011.76	927.85	859.86	803.96	757.62	719.09	687.17	661.00	**639.91**
0.3	805.50	737.21	683.76	641.38	607.57	580.61	559.25	542.506	529.577	**519.78**
0.4	660.56	610.85	573.29	544.64	522.77	506.19	493.75	484.569	477.930	**473.26**
0.5	591.45	549.20	520.34	499.84	485.26	475.01	468.02	463.47	460.76	**459.44**
0.6	578.83	527.09	500.72	484.58	474.47	468.39	465.12	**463.89**	464.16	465.52
0.7	636.77	535.72	504.67	489.59	481.87	478.46	**477.84**	479.142	481.78	485.40
0.8	815.96	571.03	527.01	509.89	502.82	**500.93**	502.12	505.324	509.87	515.38
0.9	1218.08	629.03	563.98	542.13	534.25	**532.91**	535.23	539.82	545.87	552.92

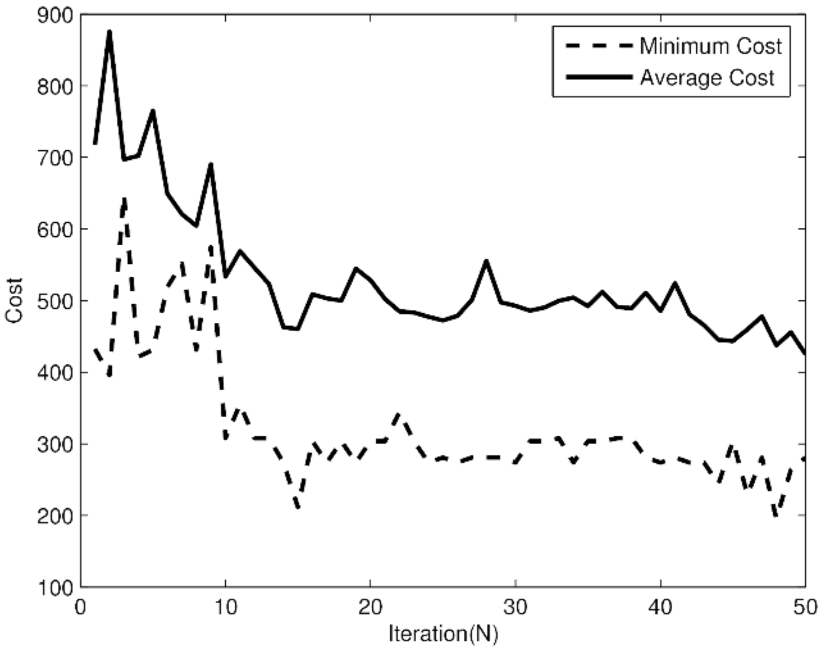

Fig. 12 Impact of cost on number of iteration.

tem [24,40] in cloud. To handle large volume of geospatial data, we used GeoSpark as it is an in-memory cluster computing system. It is an extension of Apache Spark that supports geospatial operations, indices and data types [25,64].

The architecture of GeoSpark consists of Geospatial Resilient Distributed Dataset Layer, Geospatial Query Processing Layer and Apache Spark layer. Geospatial Resilient Distributed Dataset Layer extends the Spark. There are three types of Resilient Distributed Dataset (RDD) in this layer i.e. Point, Rectangle and Polygon RDD. It contains geometrical operations library for every RDD. Geospatial Query Processing Layer is used to perform different types of geospatial queries. Geospark uses MapReduce framework derived from Apache Spark. Apache Spark consists of all the components present in Spark. It performs loading and querying data. It is much faster than SpatialHadoop that is run in MapReduce framework. It is investigated that

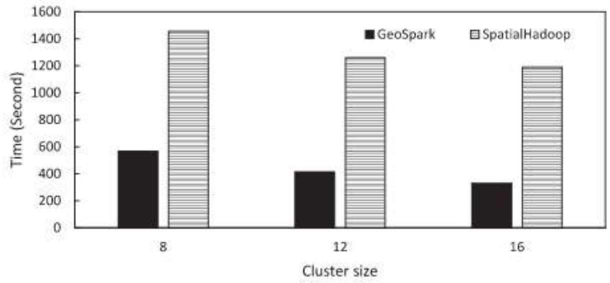

Fig. 13 Run time analysis comparison of SpatialHadoop and GeoSpark.

GeoSpark tool is powerful and handy to use and can efficiently handles geospatial big data analytics. It has the capability to add more functionalities and operations in each of these tools as per the requirements. Figure 13 shows the runtime analysis between SpatialHadoop and GeoSpark according to the cluster size. From the graph, it is clear that GeoSpark has the edge over SpatialHadoop for geospatial big data analytics when the cluster size compared with time span for geospatial big data processing. From the above analysis, we found that the addition of GeoSpark at Cloud-SDI layer shown greater processing power in terms of real time geo-health data size. In *GeoFog4Health*, we observed that the fog node can be replace with Raspberry Pi for better run time analysis of various data set. It reduces the analysis overhead to the cloud server as compare with Intel Edition as] discussed in the next section.

4 Results & Discussions

4.1 Analysis of Computation Time

We used Intel Edition and Raspberry Pi as fog device in proposed *GeoFog4Health* architecture. Processing time of Intel Edison is greater time than Raspberry Pi [20]. Intel Edison has processing time of order NLog(N) where N defines the size of dataset. We found that Raspberry Pi completed the same process almost two times faster than Intel Edison. The main network was designed in framework between the client-tier layer and the cloud layer. It is assumed that the mean arrival rate of transmitted data would be once per minute assuming that the fog node is placed in the locations where only a small number of devices in that area exist. The average waiting time for each fog node was calculated using the Littles Law [1]. We used malaria positive geospatial data for the different bench-marking experiment. We calculated average memory load, CPU processing time in percentage and power consumption (in Watt). Figure 14 shows performance comparison between Cloud-SDI and *GeoFog4Health* framework using Intel Edition and Raspberry Pi processor. From the comparison analysis, it is clear that while running one set at a instant of time, the average waiting time for Cloud-SDI framework is 189:45 seconds, and the average waiting time for *GeoFog4Health* with Intel Edition processor is 73:57 seconds where as with Raspberry Pi has around 10:20 seconds. Also, the service rate with Raspberry Pi is one third of Intel Edition in *GeoFog4Health* framework. We found that the *GeoFog4Health* framework with

Table 5 Comparison of Cloud-SDI and proposed architecture.

Characteristics	**Cloud-SDI**	**Proposed architecture**
Bandwidth Requirements and Internet Connectivity	In this framework, it requires clients to have network connectivity to the cloud server for the entire duration of services and bandwidth requirements grow with the total amount of geospatial data generated by different varieties of clients.	In this framework, it operates autonomously to provide uninterrupted services even no or intermittent Internet connectivity and network bandwidth requirements grow with total the amount of data that need to be process and sent to the cloud server after being filtered by the fog layer and intermediate fog layer.
Size	At cloud layer, processing has done with large amount of geospatial data at a time and each typically contains tens of thousands of integrated servers	At fog layer, a fog node in each location can be small or as required to meet another fog node for customer or client demands.
Operation	In Cloud-SDI framework, it operates in facilities and environments selected by the specific domain with well trained technical experts.	In *GeoFog4Health* framework, it operates in environments that are primarily determined by customers or their requirements. The framework may not be controlled or managed by anyone and may not be operated by technical experts.
Deployment	It requires highly sophisticated and suitable strategically planning for deployment	It requires minimal planning for deployment but challenges is to connect with one fog node to other intermediate fog node.
Server Locations	It requires centralized server in a small number of big data centers distributed environment	It often requires distributed servers in many locations and over large geographical areas, closer to users along with fog-to-fog range or cloud-to-thing range. Distributed fog nodes and systems has been controlled either in centralized or distributed manners depending upon the clients/fog node.

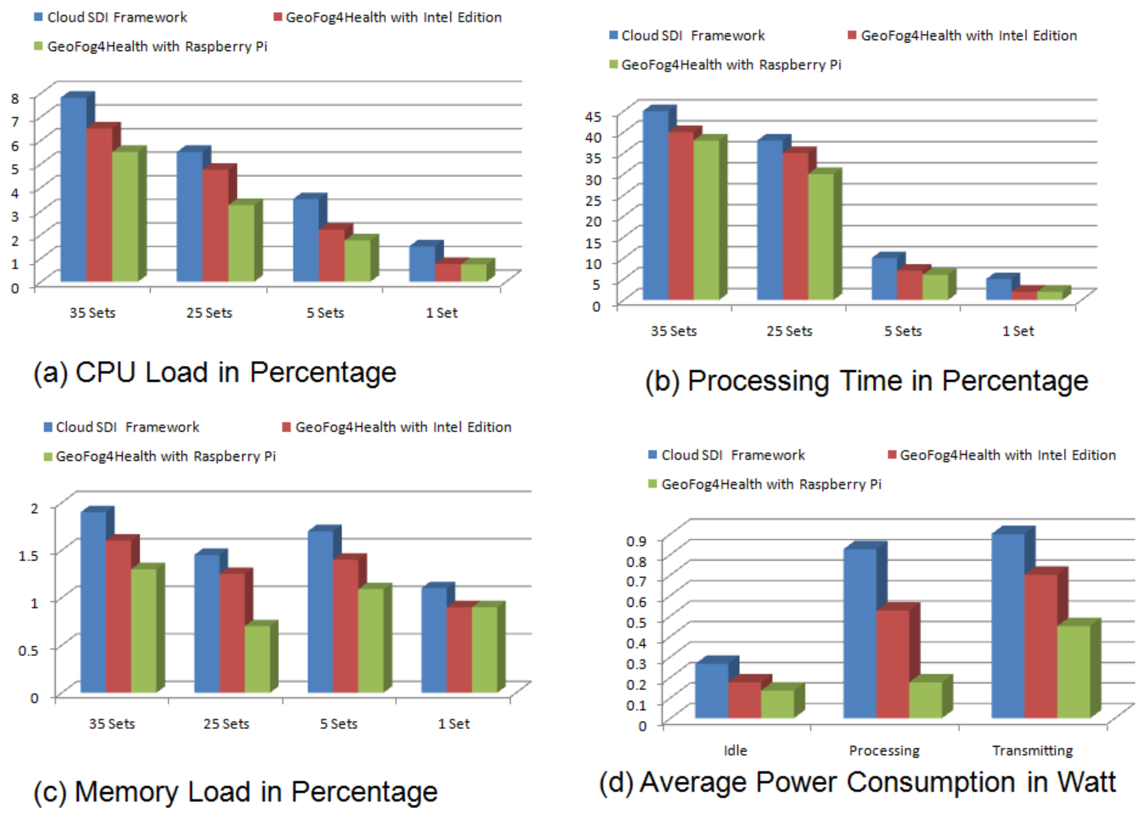

Fig. 14 Performance comparison between Cloud-SDI and *GeoFog4Health* framework using Intel Edition and Raspberry Pi.

Raspberry Pi has consumed 199mW/s where as *GeoFog4Health* framework with Intel Edition has 522mW/s when both these frameworks are in active states.

4.2 Comparison of Cloud-SDI and proposed architecture

Both Cloud-SDI and *GeoFog4Health* framework have specific meaning for a service range with in the cloud computing environment and client-tiers that provide the mutual benefit to each other and interdependent services that leads to the greater storage capacity, control and communication possible anyplace within the specified range [18]. Table 5 outlines the comparison characteristics of Cloud-SDI and *GeoFog4Health* framework.

5 Conclusions

In this study, we proposed and validated a Fog-based SDI framework for enhanced analysis of geo-spatial health data. Intel Edison and Raspberry Pi were used as fog computers in developed prototypes of proposed architecture. Fog devices reduced the storage requirements, transmission power leading to overall efficiency. Fog computing enhances the data analysis by increasing the throughput and reducing the latency. Geo-health data of malaria vector borne disease positive maps of Maharastra state in India was used for case study. We analyzed the energy saving and cost analysis for proposed *GeoFog4Health* architecture. Further, the comparison of computation time showed the efficacy of proposed fog architecture over Cloud-SDI for enhanced analysis of geo-health data. Thus, the fog devices add edge intelligence in geo-health data analysis by introducing local processing within cloud computing environments.

In future, we would like to add intelligent processing functions and feasibility aspects of fog Layer within SDI framework at national level in coastal, education, watershed, natural resource, energy and environmental monitoring sector. We plan to use mist computing in proposed framework for geospatial data analysis and management.

References

1. Available on : https://www.isixsigma.com/ dictionary/littles-law/ (Accessed on: 12th January 2017)
2. Available on : http://qgiscloud.com/rabindrabarik2016/malaria?mobile=true (Accessed on: 13th December 2016)
3. Available on : http://qgiscloud.com/ (Accessed on: 27th January 2017)
4. Available on :http://boundlessgeo.com/products/opengeo-suite/ (Accessed on: 27th January 2017)
5. Available on : http://qgiscloud.com/rabindrabarik2016/malaria?mobile=false (Accessed on: 6th December 2016)
6. Andreu-Perez, J., Poon, C.C., Merrifield, R.D., Wong, S.T., Yang, G.Z.: Big data for health. IEEE journal of biomedical and health informatics **19**(4), 1193–1208 (2015)
7. Barik, R., Samaddar, A., Gupta, R.: Investigations into the efficacy of open source gis software. Proceedings of the International Conference on Map World Forum (2009)
8. Barik, R.K.: Cloudganga: Cloud computing based sdi model for ganga river basin management in india. International Journal of Agricultural and Environmental Information Systems (IJAEIS) **8**(4), 54–71 (2017)
9. Barik, R.K., Dubey, H., Samaddar, A.B., Gupta, R.D., Ray, P.K.: Foggis: Fog computing for geospatial big data analytics. In: Electrical, Computer and Electronics Engineering (UPCON), 2016 IEEE Uttar Pradesh Section International Conference on, pp. 613–618. IEEE (2016)
10. Barik, R.K., Samaddar, A.B.: Service oriented architecture based sdi model for education sector in india. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013, pp. 555–562. Springer (2014)
11. Barik, R.K., Samaddar, A.B.: Service oriented architecture based sdi model for mineral resources management in india. Universal Journal of Geoscience **2**(1), 1–6 (2014)
12. Bhatia, M., Sood, S.K.: A comprehensive health assessment framework to facilitate iot-assisted smart workouts: A predictive healthcare perspective. Computers in Industry **92**, 50–66 (2017)
13. Bonham-Carter, G.F.: Geographic information systems for geoscientists: modelling with GIS, vol. 13. Elsevier (2014)
14. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13–16. ACM (2012)
15. Buyya, R., Yeo, C.S., Venugopal, S.: Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In: High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on, pp. 5–13. Ieee (2008)
16. Chen, F., Ren, H.: Comparison of vector data compression algorithms in mobile gis. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol. 1, pp. 613–617. IEEE (2010)
17. Chen, Z., Chen, N., Yang, C., Di, L.: Cloud computing enabled web processing service for earth observation data processing. IEEE journal of selected topics in applied earth observations and remote sensing **5**(6), 1637–1649 (2012)
18. Chiang, M., Zhang, T.: Fog and iot: An overview of research opportunities. IEEE Internet of Things Journal **3**(6), 854–864 (2016)

19. Coleman, D.J., Rajabifard, A., Kolodziej, K.W.: Expanding the sdi environment: comparing current spatial data infrastructure with emerging indoor location-based services. International Journal of Digital Earth **9**(6), 629–647 (2016)

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

21. Dastjerdi, A.V., Gupta, H., Calheiros, R.N., Ghosh, S.K., Buyya, R.: Fog computing: Principles, architectures, and applications. arXiv preprint arXiv:1601.02752 (2016)

22. Dubey, H., Monteiro, A., Mahler, L., Akbar, U., Sun, Y., Yang, Q., Mankodiya, K.: Fogcare: fog-assisted internet of things for smart telemedicine. Future Gen Comput Syst (2016)

23. Dubey, H., Yang, J., Constant, N., Amiri, A.M., Yang, Q., Makodiya, K.: Fog data: Enhancing telehealth big data through fog computing. In: Proceedings of the ASE BigData & SocialInformatics 2015, p. 14. ACM (2015)

24. Eldawy, A.: Spatialhadoop: towards flexible and scalable spatial processing using mapreduce. In: Proceedings of the 2014 SIGMOD PhD symposium, pp. 46–50. ACM (2014)

25. Eldawy, A., Mokbel, M.F.: Spatialhadoop: A mapreduce framework for spatial data. In: Data Engineering (ICDE), 2015 IEEE 31st International Conference on, pp. 1352–1363. IEEE (2015)

26. Evangelidis, K., Ntouros, K., Makridis, S., Papatheodorou, C.: Geospatial services in the cloud. Computers & Geosciences **63**, 116–122 (2014)

27. Fang, S., Zhu, Y., Xu, L., Zhang, J., Zhou, P., Luo, K., Yang, J.: An integrated system for land resources supervision based on the iot and cloud computing. Enterprise Information Systems **11**(1), 105–121 (2017)

28. Georis-Creuseveau, J., Claramunt, C., Gourmelon, F.: A modelling framework for the study of spatial data infrastructures applied to coastal management and planning. International Journal of Geographical Information Science **31**(1), 122–138 (2017)

29. Giuliani, G., Lacroix, P., Guigoz, Y., Roncella, R., Bigagli, L., Santoro, M., Mazzetti, P., Nativi, S., Ray, N., Lehmann, A.: Bringing geoss services into practice: A capacity building resource on spatial data infrastructures (sdi). Transactions in GIS (2016)

30. Granell, C., Fernández, Ó.B., Díaz, L.: Geospatial information infrastructures to address spatial needs in health: collaboration, challenges and opportunities. Future Generation Computer Systems **31**, 213–222 (2014)

31. Hancke, G.P., Hancke Jr, G.P., et al.: The role of advanced sensing in smart cities. Sensors **13**(1), 393–425 (2012)

32. He, L., Yue, P., Di, L., Zhang, M., Hu, L.: Adding geospatial data provenance into sdia service-oriented approach. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **8**(2), 926–936 (2015)

33. Huang, Q., Yang, C., Liu, K., Xia, J., Xu, C., Li, J., Gui, Z., Sun, M., Li, Z.: Evaluating open-source cloud computing solutions for geosciences. Computers & Geosciences **59**, 41–52 (2013)

34. Ji, H., Wang, Y.: The research on the compression algorithms for vector data. In: Multimedia Technology (ICMT), 2010 International Conference on, pp. 1–4. IEEE (2010)

35. Khan, Z., Ludlow, D., McClatchey, R., Anjum, A.: An architecture for integrated intelligence in urban management using cloud computing. Journal of Cloud Computing: Advances, Systems and Applications **1**(1), 1 (2012)

36. Kharouf, R.A.A., Alzoubaidi, A.R., Jweihan, M.: An integrated architectural framework for geoprocessing in cloud environment. Spatial Information Research pp. 1–9 (2017)

37. Koswatte, S., McDougall, K., Liu, X.: Sdi and crowdsourced spatial information management automation for disaster management. Survey Review **47**(344), 307–315 (2015)

38. Lee, J.G., Kang, M.: Geospatial big data: challenges and opportunities. Big Data Research **2**(2), 74–81 (2015)

39. Leidig, M., Teeuw, R.: Free software: A review, in the context of disaster management. International Journal of Applied Earth Observation and Geoinformation **42**, 49–56 (2015)

40. Lenka, R.K., Barik, R.K., Gupta, N., Ali, S.M., Rath, A., Dubey, H.: Comparative analysis of spatialhadoop and geospark for geospatial big data analytics. arXiv preprint arXiv:1612.07433 (2016)

41. Li, B., Pei, Y., Wu, H., Shen, B.: Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds. The Journal of Supercomputing **71**(8), 3009–3036 (2015)
42. Li, W., Santos, I., Delicato, F.C., Pires, P.F., Pirmez, L., Wei, W., Song, H., Zomaya, A., Khan, S.: System modelling and performance evaluation of a three-tier cloud of things. Future Generation Computer Systems (2016)
43. Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., Jie, W.: Remote sensing big data computing: Challenges and opportunities. Future Generation Computer Systems **51**, 47–60 (2015)
44. Monteiro, A., Dubey, H., Mahler, L., Yang, Q., Mankodiya, K.: Fit: A fog computing device for speech tele-treatments. In: Smart Computing (SMARTCOMP), 2016 IEEE International Conference on, pp. 1–3. IEEE (2016)
45. Mwange, C., Mulaku, G.C., Siriba, D.N.: Reviewing the status of national spatial data infrastructures in africa. Survey Review pp. 1–10 (2016)
46. Patra, S.S., Barik, R.: Dynamic dedicated server allocation for service oriented multi-agent data intensive architecture in biomedical and geospatial cloud. In: Cloud Technology: Concepts, Methodologies, Tools, and Applications, pp. 2262–2273. IGI Global (2015)
47. Roy, D.G., De, D., Mukherjee, A., Buyya, R.: Application-aware cloudlet selection for computation offloading in multi-cloudlet environment. The Journal of Supercomputing pp. 1–19 (2016)
48. Salajegheh, J., Hakimpour, F., Esmaeily, A.: Developing a web-based system by integrating vgi and sdi for real estate management and marketing. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **40**(2), 231 (2014)
49. Sareen, S., Gupta, S.K., Sood, S.K.: An intelligent and secure system for predicting and preventing zika virus outbreak using fog computing. Enterprise Information Systems pp. 1–21 (2017)
50. Sareen, S., Sood, S.K., Gupta, S.K.: Iot-based cloud framework to control ebola virus outbreak. Journal of Ambient Intelligence and Humanized Computing pp. 1–18 (2016)
51. Sareen, S., Sood, S.K., Gupta, S.K.: Secure internet of things-based cloud framework to control zika virus outbreak. International Journal of Technology Assessment in Health Care pp. 1–8 (2017)
52. Schäffer, B., Baranski, B., Foerster, T.: Towards spatial data infrastructures in the clouds. In: Geospatial thinking, pp. 399–418. Springer (2010)
53. Smith, J., Mackaness, W., Kealy, A., Williamson, I.: Spatial data infrastructure requirements for mobile location based journey planning. Transactions in GIS **8**(1), 23–44 (2004)
54. Sood, S.K., Mahajan, I.: Wearable iot sensor based healthcare system for identifying and controlling chikungunya virus. Computers in Industry **91**, 33–44 (2017)
55. Vanmeulebrouk, B., Rivett, U., Ricketts, A., Loudon, M.: Open source gis for hiv/aids management. International Journal of Health Geographics **7**(1), 53 (2008)
56. Verma, P., Sood, S.K., Kalra, S.: Cloud-centric iot based student healthcare monitoring framework. Journal of Ambient Intelligence and Humanized Computing pp. 1–17 (2017)
57. Wang, X., Zhang, H., Zhao, J., Lin, Q., Zhou, Y., Li, J.: An interactive web-based analysis framework for remote sensing cloud computing. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **4**, W2 (2015)
58. Wu, B., Wu, X., Huang, J.: Geospatial data services within cloud computing environment. In: Audio Language and Image Processing (ICALIP), 2010 International Conference on, pp. 1577–1584. IEEE (2010)
59. Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., Fay, D.: Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? International Journal of Digital Earth **4**(4), 305–329 (2011)
60. Yang, C., Huang, Q., Li, Z., Liu, K., Hu, F.: Big data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth **10**(1), 13–53 (2017)
61. Yang, C., Raskin, R., Goodchild, M., Gahegan, M.: Geospatial cyberinfrastructure: past, present and future. Computers, Environment and Urban Systems **34**(4), 264–277 (2010)

62. Yang, C., Yu, M., Hu, F., Jiang, Y., Li, Y.: Utilizing cloud computing to address big geospatial data challenges. Computers, Environment and Urban Systems **61**, 120–128 (2017)

63. Yi, S., Li, C., Li, Q.: A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data, pp. 37–42. ACM (2015)

64. Yu, J., Wu, J., Sarwat, M.: Geospark: A cluster computing framework for processing large-scale spatial data. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 70. ACM (2015)

65. Yue, P., Guo, X., Zhang, M., Jiang, L., Zhai, X.: Linked data and sdi: The case on web geoprocessing workflows. ISPRS Journal of Photogrammetry and Remote Sensing **114**, 245–257 (2016)

66. Yue, P., Zhou, H., Gong, J., Hu, L.: Geoprocessing in cloud computing platforms–a comparative analysis. International Journal of Digital Earth **6**(4), 404–425 (2013)

67. Zhou, B., Dastjerdi, A.V., Calheiros, R., Srirama, S., Buyya, R.: mcloud: A context-aware offloading framework for heterogeneous mobile cloud. IEEE Transactions on Services Computing (2015)

68. Zhu, H., Yang, C.P.: Data compression for network gis. In: Encyclopedia of GIS, pp. 209–213. Springer (2008)