*Journal of Autonomous Intelligence (2024) Volume 7 Issue 1*
doi: 10.32629/jai.v7i1.845

**ORIGINAL RESEARCH ARTICLE**

# Enabling edge computing-based coverage hole detection framework for lossless data tracking

**Anitha Christy Angelin*, Salaja Silas**

*Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, India*

*** Corresponding author:** Anitha Christy Angelin, anithachristyangelin1p@gmail.com

## ABSTRACT

Countries throughout the world were experimenting with novel approaches to prevent the spreading of the Coronavirus pandemic illness (COVID-19). The use of IoT and cloud computing for data collection and analysis to avoid disease transmission via smartphone applications was a significant hurdle. Existing cloud services that are that retain the information of the victim to address severe challenges such as excessive latency and poor spectral performance. Furthermore, the likelihood of coverage gaps might result in the loss of genuine data, leading the technology to present inaccurate data. Motivated to solve these challenges, this paper addresses a new edge computing framework with a coverage hole detection module to detect and prevent the primary spread of pandemics like COVID-19 in an energy-efficient way. Experimental results exhibit excellent performance in terms of energy consumption under edge based and cloud based scenarios in the existence of coverage holes. The experimental findings demonstrate that the proposed structure has improved energy economy and reduced time to process while detecting coverage holes accurately.

***Keywords:*** edge computing; cloud computing; IoT; pandemic; wireless sensor networks; energy efficiency

**ARTICLE INFO**

Received: 30 June 2023
Accepted: 19 July 2023
Available online: 26 October 2023



## 1. Introduction

Coronavirus is a contagious type of virus, causing lesions of the nose, sinus, or upper throat[1,2]. A pneumonia cluster in Wuhan, Hubei province in China, revealed its source as a novel coronavirus in late 2019. The World Health Organization termed the virus COVID-19 in February 2020, which stands for 2019 coronavirus disease leading to its rapid spread and death toll in all countries as seen in **Figure 1**. In their battle against the pandemic, world nations have used a wide array of innovations[3,4]. The rising demand for high-speed networks for data transfer is accelerating at an alarming pace[5]. Aarogya Sethu application monitors the whereabouts of the people through the location of the information supplied by their phones was vitally important in evaluating where the infected person had gone before being quarantined[6,7]. The app collects and stores personal data such as name, phone number, age, sex, occupation, countries that have been on a visit in the last 30 days, or smoking and medical status in a cloud during registration entered by the person on the app when the app was first run to determine the existence of the coronavirus. The framework also permanently collects the location coordinates and checks where all authorized persons have been contacted. Many customers have incorporated this into their networks with the rise in cloud computing. Any businesses sell clients other than cloud computing services. Many programs incorporate broad algorithms such as the translation of natural languages, processing of images, and sound classification with these servers. This eliminates the bulk of computing pressures on consumer computers from the cloud. But that means the process is too centered. The data center then receives this responsibility. **Figure 1** illustrates the cases that were recovered are much lower than the overall cases, bolstering stress analysis towards discovering a successful end to the disease's spread.

The application and hardware design of the server now controls the shortcomings of operational analysis. Users' impact on the cloud has proven to be risky. The security of knowledge, effectiveness, and efficiency are some of the issues that cloud users might be at risk[4]. Too much reliance on the cloud may be risky for all data processes. Again, the rapid proliferation of the IoT and 5G wireless networks will have a major impact on the sheer amount of data, resulting in increased latency for conventional cloud computing technologies[5,7]. Despite the rapid rise in the amount of mobile devices, conventional centralized cloud computing is unable to provide QoS for a wide range of services. With the arrival of future 5G technologies, edge computing could be the critical answer to this problem[8]. One of the primary challenges related to 5G technologies is the Radio-Access Network (RAN). RAN data is usable in authentic mobile edge computing[9]. With the use of RAN data in real-time, network operators can increase their end-users quality of experience (QoE) as RAN will deliver real-time knowledge solutions. The modern MEC architecture has also been introduced to store, identify, and interpret IoT data sources.

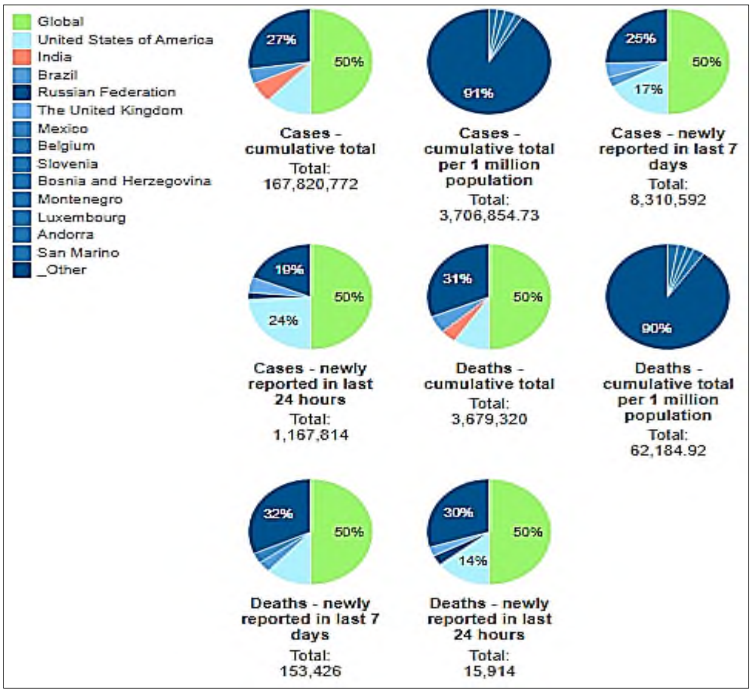


**Figure 1.** COVID impact during 2021.

The MEC architecture provides a modern supply chain and efficient mechanism that gives mobile operators and apps as well as service providers new opportunities. Making use of an edge computing system to overcome all the challenges of COVID detection in the proposed model is shown in **Figure 2**. This is the update of the COVID detection system as shown in **Figure 3**. The edge computing consists of MEC servers to provide wireless users with COVID knowledge and geographical data in real-time by utilizing the available edge near to end users. To provide mobile network providers with context-aware software and facilities, this actual network awareness can be utilized by meeting end-users quality of experience (QoE). By reducing latency and bandwidth usage, the MEC platform delivers excellent value but raises the edge's obligation to deliver multiple health services in real time to end-users. MEC allows data flow acceleration with low latency, even in real-time data analysis. Incorporating deep learning and predictive measures along with edge computing can enable the framework to extract valuable information and can take adaptive intelligent decisions[9].

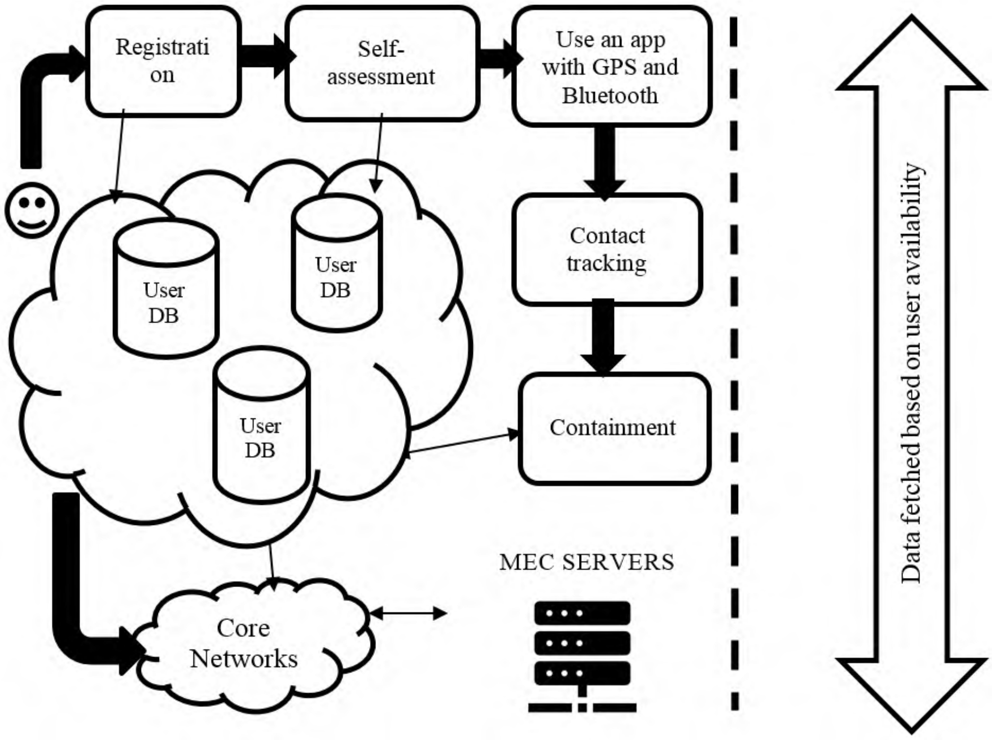


**Figure 2.** COVID-19 tracking and detection using the edge computer system.

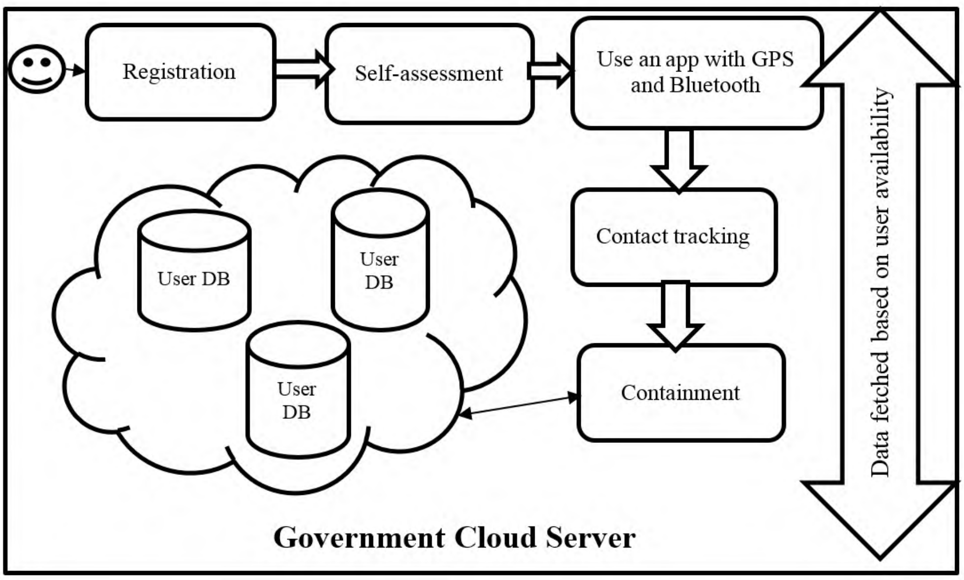


**Figure 3.** Cloud computing solution for COVID-19 observation.

The remaining part of the article is organized as follows. Section 2 presents many areas of edge computing uses in the healthcare industry. Additionally, the restoration rate in COVID detection utilizing cloud-based methods is analyzed by comparing predictive data analysis models. The behavior of edge computing apps with coverage holes is described in Section 3. In Section 4, an experimental setup is provided. In Section 5, metrics for performance are provided. The conclusion is discussed in Section 6.

## 2. Related works

All individuals in the private and public sectors have been drawn to new developments in edge computing for mobile devices, including the remote monitoring of patients using centralized. Therefore, the highest attention should be given to clear visualization, a high level of sensitivity, and improved service quality[8]. This new technique is used by numerous systems to understand data flow with higher QoS. While live tracking apps are employed for tracking patient whereabouts and recording the propagation of epidemics, stable tracking programs are used to maintain a patient's health information.

### 2.1. Static monitoring applications

Considering the network characteristics like peak-to-mean ratio, delay, and jitters from medical video footage is known as Uterine Horn Navigation. Sodhro et al.[10] proposed that the Window-based control rate Algorithm (w-RCA) in mobile devices edge-based services would increase medical facility efficiency. w-RCA findings that are more robust and successful in a small buffer and a window. The outcomes for a long data stream are unable to be improved[10]. In an integrated fog-to-cloud structure which incorporates data preprocessing, indoor position, and behavior detection processes via a residence gateway while a secure cloud acts for storing remote data, some researchers established an intelligent health surveillance atmosphere that is centered on the cloud[11–14]. Memory issues and high latency are present.

### 2.2. Dynamic monitoring applications

Dynamical monitoring applications in the diagnosis and management of epidemic diseases are becoming a sustainable approach. From this viewpoint, it is always a difficult job to set up real-time to transfer and aggravate intelligent data from various sensors. Such systems are successful at diagnosing initial infectious diseases so that adequate care can be administered in due course to enable rapid healing[15–18]. If a sensor's capacity to convey data is reached, particularly in continuous tracking and observation, a coverage hole results, which can be extremely problematic in medical applications. Approaches for preventing Zika virus infection are being described using fog, cloud computing, cell phones, and IoT sensor. According to Google Maps, each ZikaV transmission site indicates an individual who has contracted the virus and enables public health officials to appropriately monitor such risky locations. Thus, this technique does not reveal an exponential range of disorders[19,20]. Using sensing data provided by the many IoT devices, both processes are identified and evaluated in a centralized cloud that is presently hampered by latency and security issues. The majority of static and dynamic monitoring solutions struggle with latency and data safety concerns[21]. To address these difficulties, multiple investigations have been established and put into practice. The issue of the sensing devices' energy consumption, which contains vital medical data, is currently being investigated.

### 2.3. Paradigm of predictive analysis

The paradigm of predictive analysis is probably the most sought-after data analysis model. Predictive analysis models are supposed to interpret, determine patterns, classify trends, and use chronological case data to calculate the global pandemic spreading and rehabilitation rate. A statistical framework is one of the most well-known methods used in mathematical study. It controls metric value forecasts by assessing new values based on training from prior data. It is also used where the statistics found in historical documents are not available. There are different kinds of statistical models. One of the most commonly used methods of prediction research is classification models[21,22]. As shown in **Figure 4**, the model relies on historical facts to identify knowledge. The past data is fed into prediction algorithms that consider patterns and trends. The model is then used for current data to forecast the next happenings.

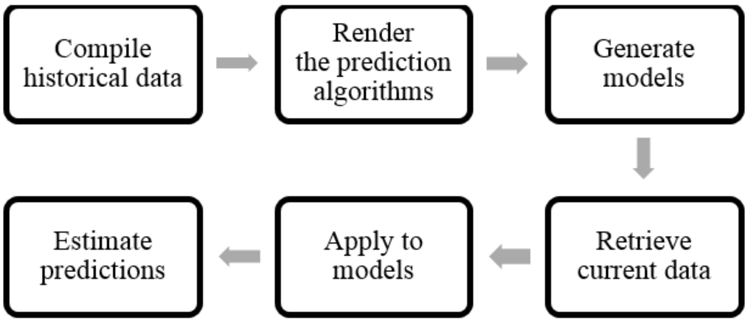


**Figure 4.** Paradigm of predictive analysis.

Global businesses use classification models so new data can be easily updated and problems can be carefully evaluated. Although historical data work in classification and prediction models, abnormal data items are utilized for modeling anomalies[23]. It works by identifying exceptional data, individually or concerning different categories and numbers. The time-series framework focuses on the data when the variables are given periodically. Several metrics that forecast trends in a certain timeframe using multiple data points collected from data from the year before are produced using the generalized linear approach[24]. The classification model separates data using identical features into distinct groups. In certain implementations, splitting data into distinct datasets based on specific characteristics is especially useful. Hazard analysis is carried out based on the generalized linear model[25], random forest model[26], decision tree model[27], and deep learning model[28]. Those models project how quickly the pandemic is going to recover soon.

### 2.3.1. Linear model

The generalized linear model (GLM) extrapolates linear regression, enabling a correlation between the linear model and the dependent variables and allowing the degree of variance to be determined by its expected value[29,30]. Abdellatief et al. explored in the IoMT system a confidence-based knowledge-sharing strategy to deduce the danger of consumer health failure from accessing a given position in compliance with its vulnerabilities[31]. The comparison between the total cases and total death is shown in **Figure 5**.

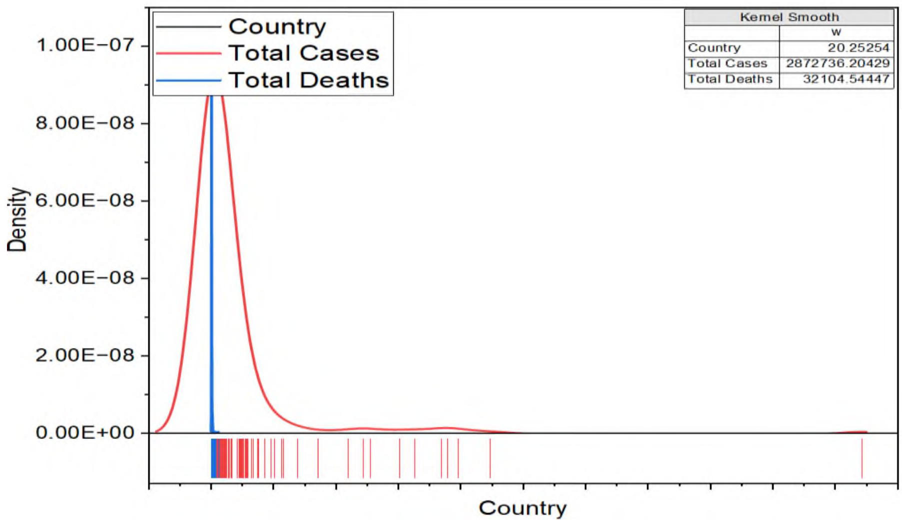


**Figure 5.** Descriptive statistics of active cases.

**Figure 6** shows the active case prediction through the generalized linear model. The predictions are within +/−16152.702 of the actual number with an accuracy of 95%. The accuracy of the forecasts increases with their proximity to the diagonal line. The locations of the points are 95% likely to be within the dotted lines.

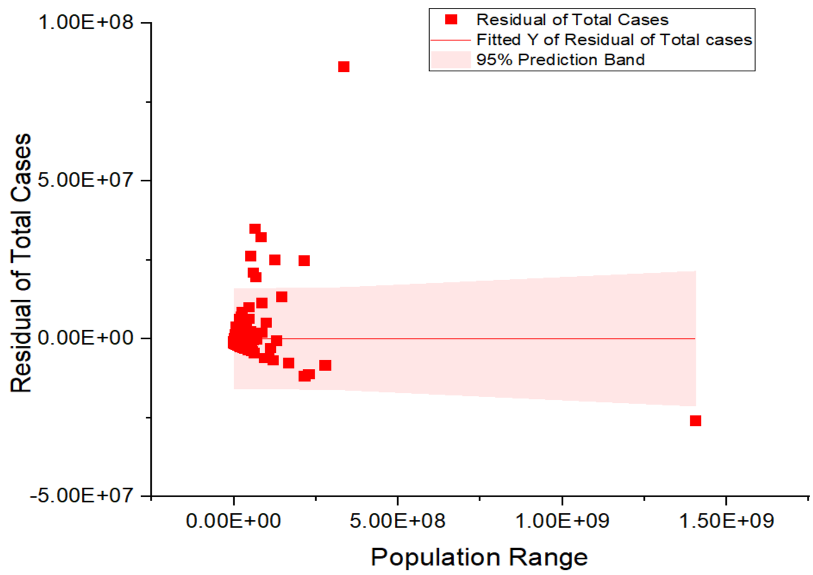


**Figure 6.** Active case prediction through the generalized linear model.

### 2.3.2. Deep learning

Deep learning models are commonly used to derive high-level spatial information to maximize efficiency compared with conventional models enhance estimation accuracy, and also to recognize and interpret biological information. **Figure 7** shows the predictions vs. the actual values of deep learning model. Ideally, all predictions equal the actual values and lie on a diagonal line. The closer they are to the diagonal line the better the predictions are. With a 95% probability, the points are between the dashed lines. With 95% probability, the predictions do not deviate more than +/−16127.184 from the actual value.

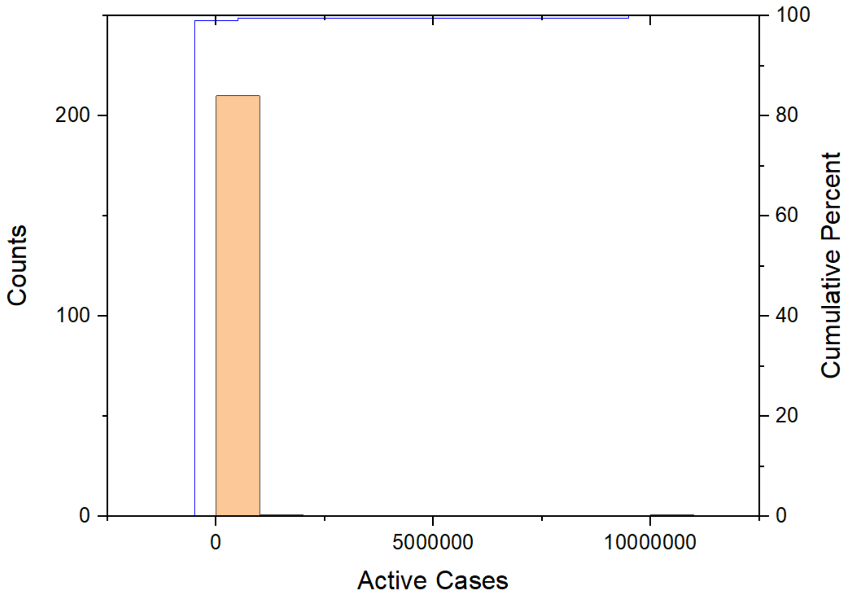


**Figure 7.** Deep learning model predictions for active instances.

### 2.3.3. Decision tree

An algorithmic approach segmenting the data set in various ways under particular criteria might produce decision trees. Probably the most analytical feature of the supervised technique is decision trees. The decision tree is a key technique for the analysis of predictions that may be utilized to both explicitly and effectively characterize beliefs. It is a graph that uses division techniques to display every possible result, notwithstanding limitations. The leaf depicts a choice made after the value of the feature is considered account if an internal decision tree node indicates a feature assessment.

**Figure 8** shows the decision tree visualization of total cases. Ideally, all predictions equal the actual values and lie on a diagonal line. The closer they are to the diagonal line the better the predictions are. With 95% probability, the points are between the dashed lines. With a 95% probability, the predictions do not deviate more than +/−16240.53 from the actual value.

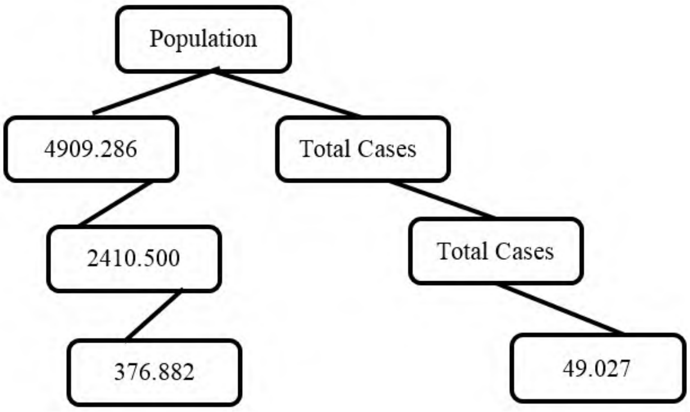


**Figure 8.** Decision tree visualization of total cases.

### 2.3.4. Random forest model

As the name suggests, the random forest contains several different decision-making entities, acting as a set. Each tree stretches a classification model in the random forest, and the group with more votes becomes the prediction of the model. All forecasts should ideally equal the actual numbers and fall on a diagonal line. The more close to the diagonal line and more accurate the forecasts. The boundaries are almost certainly within the lines with dashes. These variations influence both the peak timing and magnitude forecasts. **Figure 9** shows the probability distribution of active cases using random forest classifier.

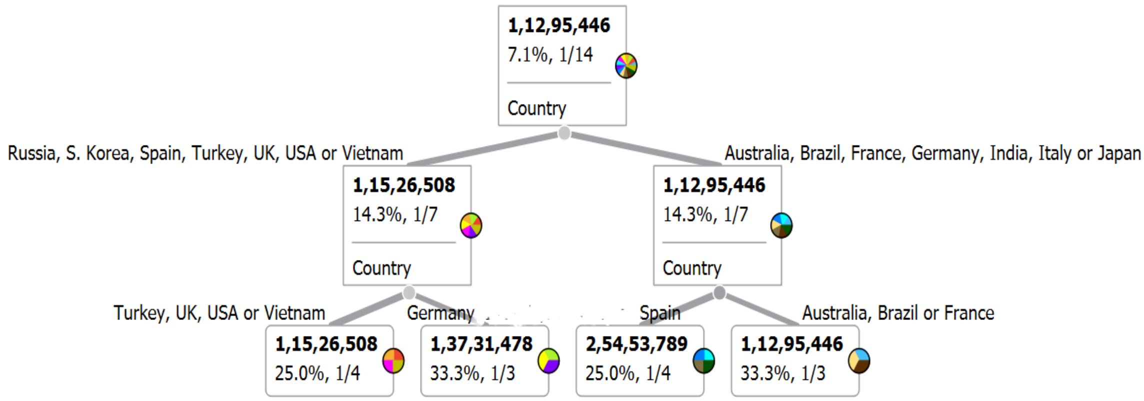


**Figure 9.** Probability distribution of active cases using random forest classifier.

The predictions have a 95% chance of being within +/−16681.614 of the actual number as well. The main obstacle in detecting the viruses in persons with COVID-19 symptoms was the lack of correct screening evidence and their capacity to screen. This has culminated in an unprecedented flux of information, which in turn affects the forecasts.

The proposed edge computing-based COVID detection for coverage hole detection seeks to identify and anticipate the position of a COVID-infected individual while consuming less power and delivering crucial information with a short latency. The structure includes the techniques that follow: i) visibility estimating to find neighbors; ii) polygon triangulating to iteratively divide segments; and iii) coverage hole detection utilizing a point location-based coverage hole detection technique.

## 3. Modeling a framework for energy-efficient edge computing-based COVID detection

The blood pressure, heart rate, non-invasive temperatures of the body, and oxygen level in the blood (SpO2) will be collected. The pulse rate of a user having a mobile device integrated with wearable sensors. The existing system makes use of a user survey to predict the medical condition whereas the proposed system provides medical accuracy using these wearable sensors.

Let us consider the following notations.

$\Phi$ = Minimum cost of triangulation;
$\varphi[v, u]$ = Minimum cost after recursive triangulation;
$D(v, u)$ = The perimeter of the distance between the edges;
$P < V_{(v-1)}\ V_{(v+1)}, \ldots, V_u \geq$ Polygon formed by connecting the vertices;
$\mu(v,u)$ = Effectiveness function to determine the cost of the triangle.

**Figure 10** shows the edge computing-based framework for COVID detection and prevention. The sensor-generated information collected by every user's smartphone or tablet could be processed privately or transferred to the MEC server for processing. MEC servers use the M/M/1 queuing model for the execution of data thereby reducing the latency. The processed data will be forwarded to the government cloud data center for prediction and tracking through core networks and the Internet. When a user searches for nearby corona diagnostic details, information is obtained from the data center in the cloud and sent to the individual via the Internet. If sensors ran out of battery or malfunction, incorrect information will be collected, presenting a major concern in the event of initial disease propagation. To address this risk, a coverage hole detection technique is utilized to locate failing sensors and refresh the data center to ensure users who are registered can get accurate data on the likelihood of interaction with a virus-infected person when their mobile device sensors are functional. The coverage hole detection technique employs a computational geometry-based hole identification methodology that can pinpoint the precise position of the failing sensor employing a point location approach. The point location-based hole detection algorithm technique employs a minimal cost triangulation solution for each zone. The following is the updated triangulation procedure for a convex polygonal structure generated by the distributed sensor nodes.

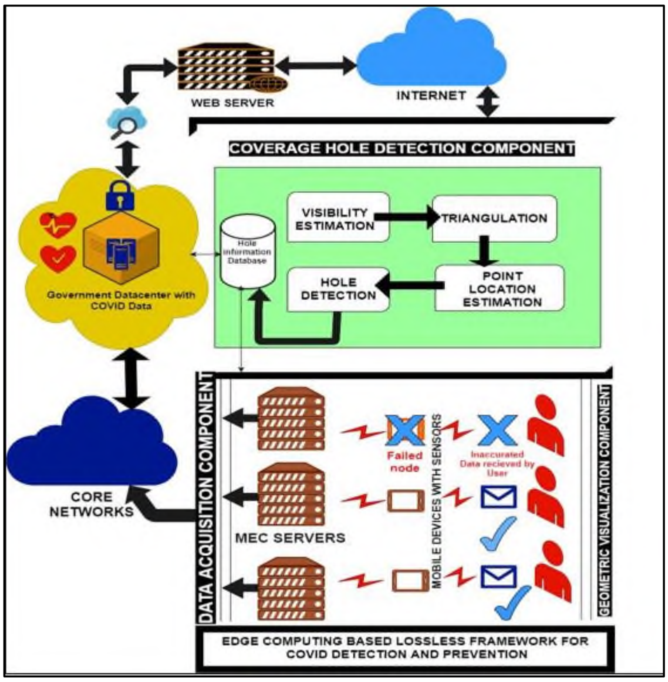


**Figure 10.** Edge computing-based framework for COVID detection and prevention.

Deliberate $\varphi$ ($v$, $u$) be the cost of an optimal triangulation of the polygon <$V_{v-1}$, $V_{v+1}$, ..., $V_u$>, then, $\varphi$ ($v$, $u$) is the least cost for point $v$-1 to $u$. Whenever there is a line segment connecting two points, then <$V_{v-1}$, $V_u$>, so $v = u$, then $\varphi$ ($v$, $u$) is 0. Otherwise, <$V_{v-1}$, $V_r$, $V_u$>)} > and all partitions <$V_{v-1} \ldots V_r$, > and <$V_{r+1}$, $V_u$, > and finding the minimum.

The cost of an edge is calculated by calculating the total length of the distance $D$ across two nodes that cause the edges, as shown below.

$$D(v,u) = \sqrt{((v.x - u.x) * (v.x - u.x) + (v.y - u.y) * (v.y - u.y))} \tag{1}$$

The effectiveness function to determine the cost of the triangles is,

$$\mu(v,u) = \sum_{v-1}^{u} D(v,u) \tag{2}$$

To triangulate $n$ possible triangles of a convex polygon of any size the algorithm on the edges is performed until all the edges are met. The minimum cost after the recursive triangulation is,

$$\varphi[v,u] = \begin{cases} 0, v = u \\ (\mu(\mathrm{v,u}) + D(v,u) + D(u,r)), otherwise \end{cases} \tag{3}$$

Then $\varphi$ (1, $n$) is the least cost of triangulation. The edges triangulation is determined twice a diagonal and once for the edge. So the time complexity for triangulation is $O(n^3)$. The monotone polygons can be divided into segments. The area between two subsequent segments in a platform corresponds to a single S-side. As a result, the point location problem can be reduced to two easier ones: because the plane is separated into vertical slabs, we can determine which slab holds a sensor node. The query processing time is $O(\log n)$ and the space complexity is $O(n)$. The hole areas are computed by picking sensor nodes that are nearest to segment parameters and identifying the exact location of the idle sensing node inside that segment. A hole detection technique is used to figure out the hole area, in which the energy level of the node corresponds to a threshold value obtained through a probabilistic methodology. Nodes having less energy than the acceptable level are detected as failure nodes that cause coverage holes.

### Experimental setup

The simulation uses a Cayenne Internet of Things (IoT) and Message Queuing Telemetry Transport (MQTT) protocol concurrently senses COVID. The application consists of a body temperature sensor unit with a microcontroller, storage unit, and transmitter that facilitate wireless communication. The application needed to obtain is delivered from the Google Play store if the cell phone is an Android phone. The Nokia developer J2ME development platform for a smartphone based on Java is used to build apps. The data sensed by the sensors can be communicated to the MEC servers using the MQTT protocol. The sensor data are uploaded to Cayenne when a link is formed. The data will be made available to users through the internet and core networks.

## 4. Results

To appraise the edge-based coverage hole detection framework, the proposed system compares it with the cloud-based detection framework using computational geometry. The proposed framework's performance is demonstrated by monitoring the average number of messages transferred in the network and the average amount of energy spent up to the completion of the coverage hole detection procedure for varied amounts of installed nodes.

### 4.1. Energy consumption rate

As demonstrated in **Figures 11** and **12**, an edge-based architecture minimizes the time needed for analyzing data utilizing the edge node, as well as the number of unwanted messages, leading to minimal power usage. Although a node must share and get coordinates from its *k* neighbors, a node's overall consumption of energy throughout data collection, delivery, and receiving could represent the entire energy. This research is being conducted in multiple sensor nodes of varied holes, and a greater amount of holes improves energy.

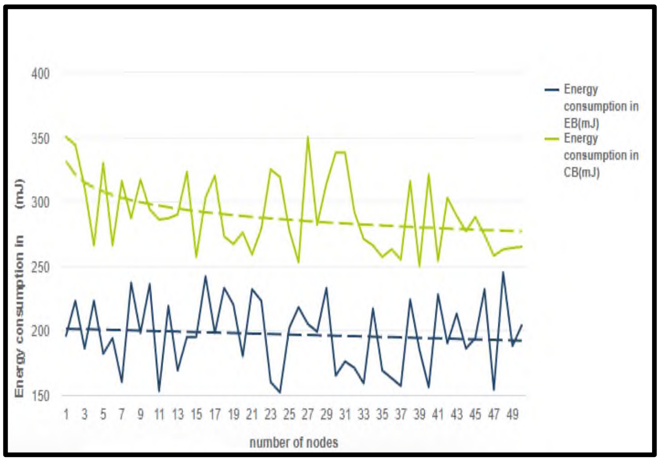


**Figure 11.** Comparison of average energy usage for 50 nodes.

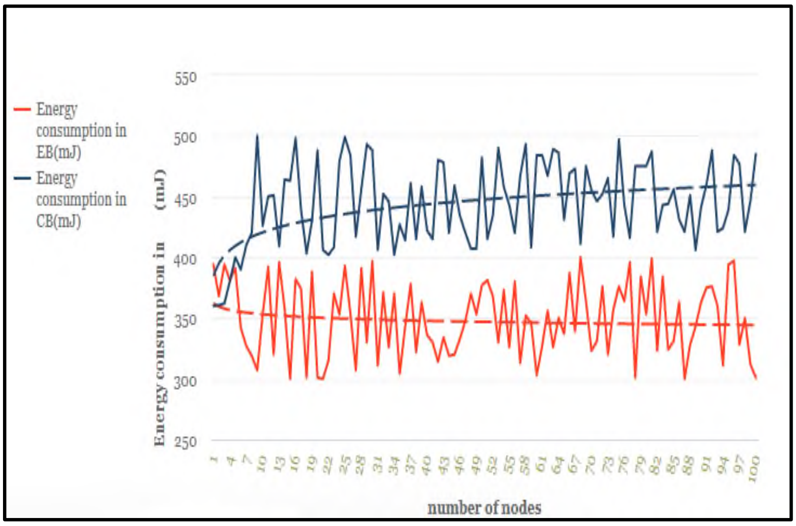


**Figure 12.** Comparison of average energy usage for 100 nodes.

The analysis of energy consumption values in **Table 1** shows that the *F* value of Cloud computing hole detection has the highest likelihood of node failure, whereas edge-based hole detection has the lowest likelihood of node failure.

**Table 1.** ANOVA analysis of energy consumption values.

| **Category** | | **DF** | **Sum of squares** | **Mean square** | ***F*** | **Prob > *F*** |
|---|---|---|---|---|---|---|
| Energy consumption in EB (mJ) | Model | 1 | 121.39 | 121.39 | 0.12 | 0.71 |
| | Error | 98 | 91817.19 | 936.91 | - | - |
| | Total | 99 | 91938.59 | - | - | - |
| Energy consumption in CB (mJ) | Model | 1 | 9948.59 | 9948.59 | 9.33 | 0.002 |
| | Error | 98 | 104414.90 | 1065.45 | - | - |
| | Total | 99 | 114363.50 | - | - | - |

### 4.2. Delay in processing critical data

Compared to the cloud computing-based coverage hole detection framework, the edge computing-based framework processes data in milliseconds.

Edge nodes consume a few minutes of calculation time, causing the computation latency to decrease. The time needed to analyze data lowers as the range of edge nodes grows. As a consequence, the delay is reduced in comparison to a cloud-based architecture, leading to a low *P*-value, shown in **Table 2** and **Figure 13**.

**Table 2.** 2-way Anova analysis of delay values.

| | DF | Sum of Squares | Mean sq | *F* value | *P*-value |
|---|---|---|---|---|---|
| EB | 9 | 8546.8996 | 949.6555 | 1.37608 | 0.21316 |
| CB | 11 | 8078.42945 | 734.4027 | 1.06417 | 0.40059 |
| Model | 20 | 28805.92582 | 1440.296 | 2.08704 | 0.01134 |
| Error | 79 | 54519.07418 | 690.1149 | - | - |
| Corrected total | 99 | 83325 | - | - | - |

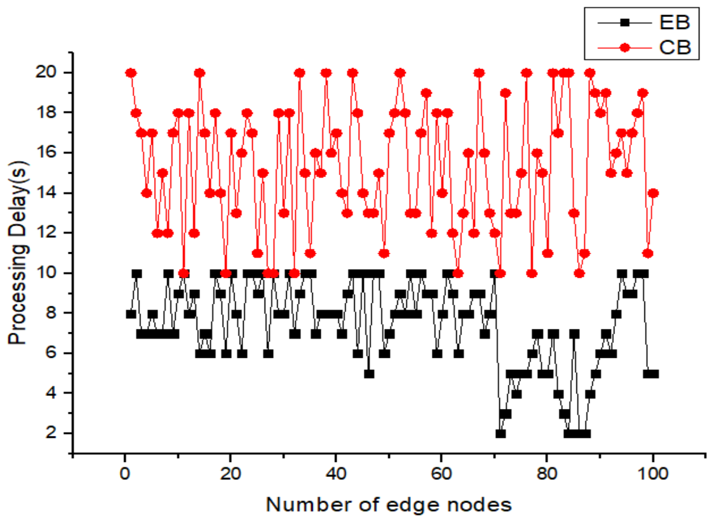


**Figure 13.** Processing delay for 100 nodes.

## 5. Conclusion

To prevent the COVID-19 pandemic, the COVID early detection and avoidance system headquartered at edge is supporting a broad, integrated healthcare infrastructure. Both medical instruments are wired to the Internet and instantly relay a message to the healthcare workers in any crucial condition. In a distant location, asymptotic situations of embedded IoT devices must be treated appropriately. To provide patient and emergency services with quality support, all urgent situations are intelligently managed. The monitoring and avoidance framework edge focused on COVID appears to be an outstanding way of tracking the compromised patient compared with cloud providers. This technology is helpful in current crucial times to ensure proactive maintenance with reliable real-time data. The edge-based COVID detection and prevention framework is successful through the use of a calculation-based geometry approach to forecasting the emerging situation with low latencies and high energy efficiencies. Researchers and health professionals may build better conditions to fight this epidemic by properly exercising this capability.

## Author contributions

Conceptualization, ACA; methodology, ACA; software, ACA; validation, SS; formal analysis, ACA;

investigation, ACA; resources, ACA; data curation, ACA; writing—original draft preparation, ACA; writing—review and editing, SS; visualization, ACA; supervision, SS; project administration, SS; funding acquisition, ACA. All authors have read and agreed to the published version of the manuscript.

## Conflict of interest

The authors declare no conflict of interest.

## References


1. Tong A, Sorrell TC, Black AJ, et al. Research priorities for COVID-19 sensor technology. *Nature Biotechnology* 2021; 39(2): 144–147. doi: 10.1038/s41587-021-00816-8
2. Rothan HA, Byrareddy SN. The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak. *Journal of Autoimmunity* 2020; 109: 102433. doi: 10.1016/j.jaut.2020.102433
3. Zafari A, Zurita-Milla R, Izquierdo-Verdiguier E. Evaluating the performance of a random forest kernel for land cover classification. *Remote Sensing* 2019; 11(5): 575. doi: 10.3390/rs11050575
4. Schaid DJ, Tong X, Batzler A, et al. Multivariate generalized linear model for genetic pleiotropy. *Biostatistics* 2019; 20(1): 111–128. doi: 10.1093/biostatistics/kxx067
5. Alam MGR, Munir MS, Uddin MZ, et al. Edge-of-things computing framework for cost-effective provisioning of healthcare data. *Journal of Parallel and Distributed Computing* 2019; 123: 54–60. doi: 10.1016/j.jpdc.2018.08.011
6. Gupta R, Bedi M, Goyal P, et al. Analysis of COVID-19 tracking tool in India: Case study of Aarogya Setu mobile application. *Digital Government: Research and Practice* 2020; 1(4): 1–8. doi: 10.1145/3416088
7. Safavat S, Sapavath NN, Rawat DB. Recent advances in mobile edge computing and content caching. *Digital Communications and Networks* 2020; 6(2): 189–194. doi: 10.1016/j.dcan.2019.08.004
8. Mharsi N, Hadji M. A mathematical programming approach for full coverage hole optimization in Cloud Radio Access Networks. *Computer Networks* 2019; 150: 117–126. doi: 10.1016/j.comnet.2018.12.015
9. Deng X, Xu M, Yang LT, et al. Energy balanced dispatch of mobile edge nodes for confident information coverage hole repairing in IoT. *IEEE Internet of Things Journal* 2019; 6(3): 4782–4790. doi: 10.1109/JIOT.2018.2869110
10. Sodhro AH, Luo Z, Sangaiah AK, Baik SW. Mobile edge computing based QoS optimization in medical healthcare applications. *International Journal of Information Management* 2019; 45: 308–318. doi: 10.1016/j.ijinfomgt.2018.08.004
11. Wang J, Ju C, Kim H, et al. A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs. *Cluster Computing* 2019; 22(3): 1787–1795. doi: 10.1007/s10586-017-1586-9
12. Wang X, He X, Feng F, et al. TEM: Tree-enhanced embedding model for explainable recommendation. In: Proceedings of the 2018 World Wide Web Conference; 23–27 April 2018; Lyon, France.
13. Matasci G, Hermosilla T, Wulder MA, et al. Large-area mapping of Canadian boreal forest cover, height, biomass and other structural attributes using Landsat composites and lidar plots. *Remote Sensing of Environment* 2018; 209: 90–106. doi: 10.1016/j.rse.2017.12.020
14. Pham M, Mengistu Y, Do H, Sheng W. Delivering home healthcare through a cloud-based smart home environment (CoSHE). *Future Generation Computer Systems* 2018; 81: 129–140. doi: 10.1016/j.future.2017.10.040
15. Chen M, Li W, Hao Y, et al. Edge cognitive computing based smart healthcare system. *Future Generation Computer Systems* 2018; 86: 403–411. doi: 10.1016/j.future.2018.03.054
16. Ai Y, Peng M, Zhang K. Edge computing technologies for internet of things: A primer. *Digital Communications and Networks* 2018; 4(2): 77–86. doi: 10.1016/j.dcan.2017.07.001
17. Verma P, Sood SK. Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet of Things Journal* 2018; 5(3): 1789–1796. doi: 10.1109/jiot.2018.2803201
18. Koriem SM, Bayoumi MA. Detecting and measuring holes in Wireless Sensor Network. *Journal of King Saud University—Computer and Information Sciences* 2020; 32(8): 909–916. doi: 10.1016/j.jksuci.2018.08.001
19. Qiu C, Shen H, Chen K. An energy-efficient and distributed cooperation mechanism for k-coverage hole detection and healing in WSNs. In: Proceedings of 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems; 19–22 October 2015; Dallas, TX, USA.
20. Verma M, Sharma S. A greedy approach for coverage hole detection and restoration in wireless sensor networks. *Wireless Personal Communications* 2018; 101(4): 75–86. doi: 10.1007/s11277-018-5668-7
21. Khedr AM, Osamy W, Salim A. Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks. *Computer Communications* 2018; 124: 61–75. doi: 10.1016/j.comcom.2018.04.002
22. Pant Di, Verma S, Dhuliya P. A study on disaster detection and management using WSN in himalayan region of uttarakhand. In: Proceedings of 2017 3rd International Conference on Advances in Computing, Communication and Automation (Fall); 15–16 September 2017; Dehradun, India.

23. Isravel DP, Arulkumar D, Angelin AC. Cyber security threats and risk mitigation measures in internet of things. *International Journal of Civil Engineering and Technology* 2018; 9(10): 1619–1628.
24. Robert William J, Anitha Christy Angelin P. Investigations on graph-based boundary detection for hole estimation in wireless sensor networks. In: Proceedings of 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA); 29–31 March 2018.
25. Angelin ACP, Isravel DP, Arulkumar D. Throughput intensification for IoT based subversive drainage monitoring applications in wireless sensor networks. *International Journal of Civil Engineering and Technology* 2018; 9(11): 2186–2196.
26. Sareen S, Gupta SK, Sood SK. An intelligent and secure system for predicting and preventing Zika virus outbreak using Fog computing. *Enterprise Information Systems* 2017; 11(9): 1436–1456. doi: 10.1080/17517575.2016.1277558
27. Yu W, Liang F, He X, et al. A survey on the edge computing for the Internet of Things. *IEEE Access* 2017; 6: 6900–6919. doi: 10.1109/ACCESS.2017.2778504
28. Soundarya A, Santhi V. An efficient algorithm for coverage hole detection and healing in wireless sensor networks. In: Proceedings of 2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech); 28–29 April 2017; Kolkata, India.
29. Amgoth T, Jana PK. Coverage hole detection and restoration algorithm for wireless sensor networks. *Peer-to-Peer Networking and Applications* 2017; 10(1): 66–78. doi: 10.1007/s12083-015-0407-2
30. Khalifa B, Al Aghbari Z, Khedr AM, Abawajy JH. Coverage hole repair in WSNs using cascaded neighbor intervention. *IEEE Sensors Journal* 2017; 17(21): 7209–7216. doi: 10.1109/JSEN.2017.2755122
31. Abdellatief W, Abdelkader H, Hadhoud M. An energy-efficient coverage hole detection technique for randomly deployed wireless sensor networks. In: Proceedings of 2016 11th International Conference on Computer Engineering and Systems (ICCES); 20–21 December 2016; Cairo, Egypt.