## LITERATURE SURVEY:

**Paper Title : A novel IOT–fog–cloud-based healthcare system for monitoring and predicting COVID-19 outspread.**

**Author Name : Ahanger, Tariq Ahamed, Usman Tariq, Muneer Nusir, Abdulazi Aldaej, Imdad Ullah, and Abdullah Sulman.**

**Publication : 2022**

**Methodology : Fuzzy C-mean classification**

The underlying layer is focused to gather patient's well-being-related information acquired progressively from the diferent sensors including health sensors, ecological sensors, meteorological sensors, and geographical sensors.The proposed framework includes a four-level architecture for the expectation and avoidance of COVID-19 contamination. The presented model comprises COVID-19 Data Collection (C-19DC) level, COVID-19 Information Classification (C-19IC) level, COVID-19-Mining and Extraction (C-19ME) level, and COVID-19 Prediction and Decision Modeling (C-19PDM) level.perceived from diferent gadgets is communicated to the intelligent device which is considered a fog hub and is responsible for information analysis. Moreover, it is sent for pre-preparing and investigation to the associate COVID-19 illness over the geographical location. Notwithstanding information collection, the protection and security of IOT devices are additionally targeted. For this reason, vital protocols are used for security to the information productively. All the data to and from IOT gadgets require information security, for example, secure socket layer (SSL) convention and elliptic curve cryptography (ECC) calculation. These conventions guarantee secure correspondence over HTTP and message query telemetry protocol (MQTP). The procured data perceived from different gadgets is communicated to the intelligent device which is considered a fog hub and is responsible for information analysis.

Moreover, it is sent for pre-preparing and investigation to the associate COVID-19 illness over the geographical location. Not with standing information collection, the protection and security of IOT devices are additionally targeted. For this reason, vital protocols are used for security to the information productively. All the data to and from IOT gadgets require information security, for example, secure socket layer (SSL) convention and elliptic curve cryptography (ECC) calculation. These conventions guarantee secure correspondence over HTTP and message query telemetry protocol (MQTP). Furthermore, with a user-friendly API, the IOT framework can be transformed to a product-level stage. Amazon Web Services (AWS) for the IOT model permits analyzing information with high precision and adequacy. Other than Amazon, Apple and Google additionally give security to unapproved access. For real-time handling, all information esteems are prepared at the fog layer for time-delicate alerts and caution about a contaminated user if any symptoms are distinguished.To recognize contaminated patients and locales using fuzzy C-mean classification to forecast the rising of COVID-19 for detecting the real-time hot spots for disease for precautionary measures by healthcare agencies.

**Paper Title : Development of An Android Application for Viewing Covid-19 Containment Zone and Monitoring Violators Who are the trespassing into it Using Firebase and Geofencing**

**Author Name : Ranajoy Mallik,Amlan Protim Hazarika, Sudarshana Ghosh Dastidar Dilip Sing,Rajib Bandyopadhyay**

**Publication : Year 2020**

**Methodology : Geofencing**

The Android application shows the location of the containment zones to the users. It also notifes the user when he or she trespasses the boundary of a containment zone or stays in the containment zones . There are mainly three activities in the application. The first activity consists of a welcome screen which is designed with images and information. Next activity is a screen displaying the instructions to operate the application and a disclaimer .The third activity is a maps activity which shows all the containment zones in a google map. This activity also has a bottom sheet which can be pulled up to show the real time Covid-19 statistic of West Bengal. The application uses Firestore which is a fexible and scalable database for mobile, web and server developments from Firebase and Google cloud platform (Cloud Firestore 2020). In Cloud Firestore, the mobile application supports serverless app architecture where the application connects to the Cloud Firestore database directly without any intermediate servers in between (Cloud Firestore SDKs and client libraries 2020). The application receives data from the database using WebSocket. The Web Socket transfers data at a higher speed than HTTP. The database transfers new data to the user as soon as it is updated. When changes are saved in the database then all connected clients receive the updated data almost instantly. The Cloud Firestore does not always fetch data from the database unless the data has been changed, it gathers previous data from the cache memory which also enables its ofine functioning.The Cloud Firestore features a NoSQL, document-oriented database (Cloud Firestore Data model 2020) and the data is stored in a JavaScript Object Notation (JSON) format. The location data are stored in documents, which are organized into collections. All the containment zones are stored in a collection in which each containment zone is represented as an individual document. Each document has four felds namely “Lat”, “Long”, “locationName” and “radius” for storing latitudes, longitudes, location names and radius respectively. It shows the documentoriented Cloud Firestore database with data of few containment zones. The “radius” feld in each document is used to indicate the radius of the containment zone. In the development stage, the billing plan for the database used here is the frebase spark plan which is the free plan provided by Firebase. This plan has few limitations like 50,000 and 20,000 reads and writes per day. These usage limits can be resolved by choosing a diferent paid plan from Firebase according to requirement. The application gets data from the Cloud Firestore database.

A collection is created in Cloud Firestore with containment zones as documents. Each document has four felds: latitude, longitude, location name and radius. Accordingly, a Java

object is created which can get the data from the document. In the map's activity, the frebase Firestore instance and collection references are created to which a snapshot listener is attached. The snapshot listener retrieves the document snapshots which are then converted into the Java object mentioned earlier. With the help of getters each data from the document is retrieved and are converted to string. Markers and circles are set using the location coordinates and radius and tags are given by the location names. The google map gets populated with these markers surrounded by circles which represents the containment zones. A JSON request is made with the get method to the REST API URL which returns the West Bengal Covid-19 case data as a response. The response is converted to a JSON object and the information is extracted. The google map shows all the containment zones in West Bengal along with the location of the user using "set my location" enabled (Google map API). The Geofencing API can create up to 100 geofences per device and the number of containment zones are more than 1000 in West Bengal. 100 geofences on the 100 nearest containment zones. Once the map is loaded and populated with containment zones and user's location, the user can press a button to add the geofences on the closest 100 containment zones. The snapshot listener returns the documents containing the locations of the containment zones from Firestore which are not sorted according to the distance between the user and containment zone. The distance between the user and each containment zone is measured using distance between method of the location manager (Android developer-Locations 2020). This distance is then used as a key and is stored along with the document in a Tree map. Likewise, all the containment zones with their distance from the user get stored in the tree map and get sorted according to the distance or key. First 100 entries from the tree map are retrieved and geofences are created on these 100 containment zones. Once the geofences are set, the user can get notifcation on entering the containment zones . The geofences get triggered and notifcations are sent even when the app is not running in background.

**Paper Title : Development of an Android Application for COVID-19 using Firebase and Geo-fencing**

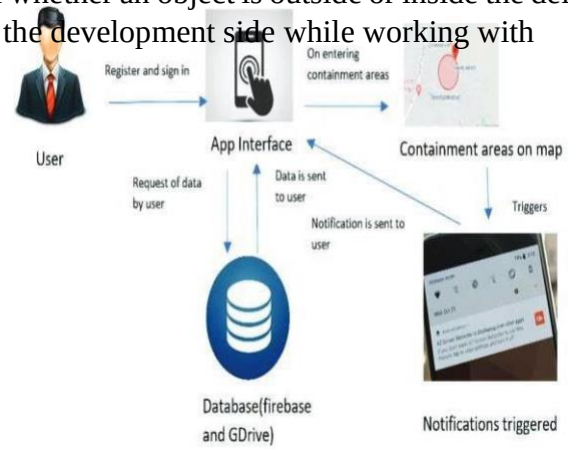
**Author Name : Jannaikode Yashwanth Kumar and S.V. Suryanarayana**

**Publication : 2021**

**Methodology : Geo-fencing**

An efficient model is proposed to reach people's needs and facilitate the services conveniently. The model does not store the geographical location of the user in the database, rather it shows his/her location in the application itself. Privacy is given great importance in the proposed model, which is lacking in many other existing applications. Static and dynamically triggered notifications by the applicationhelp the user to be alert all the time. Containment zones are clearly marked on the maps to send the alerts on entering the regions. Many other features are included in this application and they are as follows:COVID-19 information, State-wise and overall COVID-19 cases statistics, Press-meets, Media Bulletins and Orders issued by the government (GO's) , Addresses and contacts of various COVID-19 health centers ,Google maps with containment regions highlighted ,One click to reach out to emergency servicesAndroid studio is used to develop the entire application. Third party services like Google firebase database, firebase cloud messaging service and firebase in-app messaging service are used to facilitate various services. Google drive is used to store various kinds of information. When a user needs the data, they are redirected to a particular file in the drive. The user login is made mandatory which is possible after the registration. User is authenticated by the authentication services of firebase. User gets static and dynamic notification once he/she registers in the application. Any information updated in the application is notified by static notifications. Most important information is displayed on the opening screen, which is done with the help of in-app messaging services provided by firebase. Various departments' contact numbers are displayed in the application, which helps the user to call the required department in one click. Dynamic notifications which are enabled using android notification manager are triggered based on a criterion set, i.e., on entering and exiting the radius defined in a containment region. B. Geo-FencingGeo-fencing feature enables it to detect when an object enters in and goes out of the area defined. This feature is used in the proposed model to notify the user on entering the containment areas. Whenever the user enters and exits the region, a notification is triggered by the application, provided the application runs in the background of the user's mobile. Notification stating "You are in the containment region" is popped up with the notification sound. The regions on map are defined by providing latitude and longitude in the database. These are fetched by android and the circles are marked on the map with latitude and longitude as center and pre-defined radius as size. Geo fencing is used in various fields depending on the requirement of the organization [18]. The present model is built completely from scratch by coding. Besides this, there are third party geo fencing API services which makes work simpler.

One such is TOMTOM API, which interprets virtual boundaries in real geographical areas. Also we can have knowledge of whether an object is outside or inside the defined areas. TOMTOM API provides various roles for the development side while working with




Block diagram of application functionality


Screenshots of application with different features

**Paper Title : Wearable IOTS and geo-fencing based framework for COVID-19 remote patient health monitoring and quarantine management to control the pandemic.**

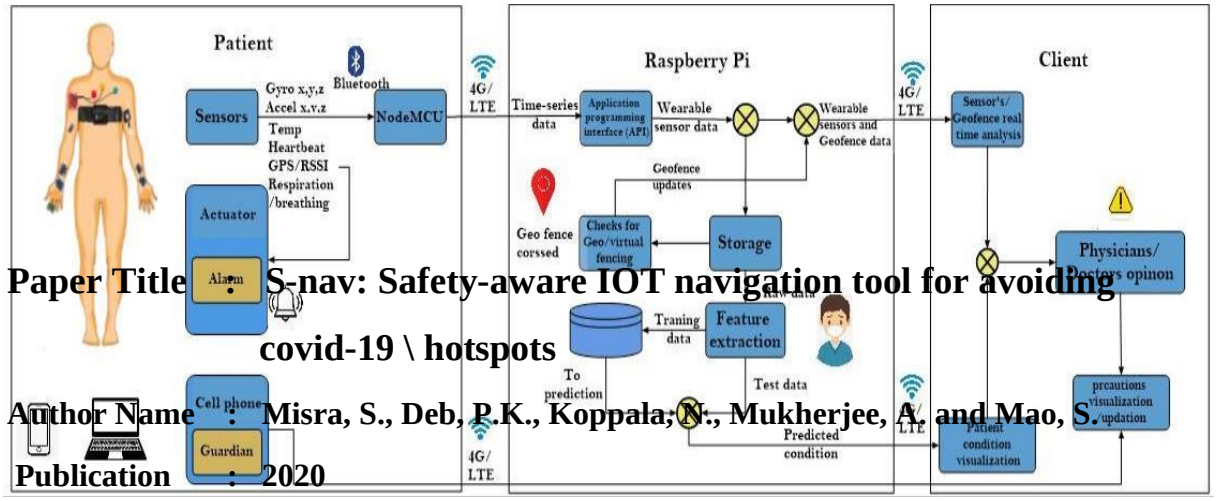
**Author Name : Ullah, F., Haq, H.U., Khan, J., Safeer, A.A., Asif, U. and Lee, S.,**

**Publication : 2022**

**Methodology : Geo-fencing**

They propose an intelligent health monitoring framework using wearable Internet of Things (IOT) and Geo-fencing for COVID-19 susceptible and patient monitoring, and isolation and quarantine management to control the pandemic. The proposed system consists of four layers, and each layer has different functionality: a wearable sensors layer, IoT gateway layer, cloud server layer, and client application layer for visualization and analysis. The wearable sensors layer consists of wearable biomedical and GPS sensors for physiological parameters, and GPS and Wi-Fi Received Signal Strength Indicator acquisition for health monitoring and user Geo-fencing. The IOT gateway layer provides a Bluetooth and Wi-Fi based wireless body area network and IOT environment for data transmission anytime and anywhere. Cloud servers use Raspberry Pi and ThingSpeak cloud for data analysis and web-based application layers for remote monitoring based on user consent. The susceptible and patient conditions, real-time sensor's data, and Geo-fencing enables minimizing the spread through close interaction. The results show the effectiveness of the proposed framework.An intelligent health monitoring framework using wearable Internet of Things (IOT) and Geo-fencing for COVID-19 susceptible and patient monitoring, and isolation and quarantine management to control the pandemic. To enclose a patient within a Geo-fence, many types of techniques are used. The GPSworks well for the outside environment, but, in indoor isolation, the GPS does not performwell [11,27]. The GPS National Marine Electronics Association (NMEA) gives information about a position in terms of strings that have coordinates i.e., latitude, longitude,and altitude [28,29]. The Geo-fence for in-door purposes can be implemented by the RSSIlevel of the communication interface such as Bluetooth [27] and RSSI level of Wi-Fi [11].They integrate both the GPS and RSSI-based Geo-fencing to monitor whetherthe susceptible and patients is inside the isolation and quarantine place or not. The distancefrom the access point (Wi-Fi) and the patient can be calculated using RSSI power receivedElectronics by the controller [11]. The drawback for Bluetooth RSSI level is the short-range coverage [27]. Thus, we have used the RSSI level of Wi-Fi for indoor environments with GPSintegration for outdoor coordinates. Related work for RSSI and Geo-fencing for COVID. the

overview of the proposed architecture to Geo-fence the COVID-19susceptible and patients and remotely monitoring their health conditions. The proposedsystem consists of four main modules: (1) Wearable Sensors for physiological data andlocation information; (2) Bluetooth and Wi-Fi-based wireless transmission of the data;(3) Flask and ThingSpeak cloud servers for data analytic and processing; (4) Web-basedApplication for remote data and Geo-fence visualization. This system can help to Electronics remotely monitor the COVID-19 susceptible and patients to overcome the transmissiondue to interaction. The human body is equipped with wearable biomedical sensors and sensors forsurveillance detection, which are placed on left, right, and chest positions of patients. The physiological biomedical parameters of patientare extracted from the sensors such as Electrocardiogram (ECG), Pulse Oximeter (SpO2),body temperature, respiratory sensor, accelerometer, and gyroscope. We interfaced theGoogle Map so that the administrator can select the Geo-fence area for the susceptible andpatients for quarantine allocation and management. The GPS sensor is installed to send theuser location to the server for checking and monitoring the isolation, movement, and Geofence of the susceptible and patients. We extract Latitude and Longitude from GPS data.The SpO2, gyroscopes, and accelerometer data are transmitted through Bluetooth to themain-controller installed at the belly position using master–slave Bluetooth configuration.The respiratory, GPS, body temperature, and ECG are directly interfaced with the controller.The NodeMCU (Wi-Fi) is used to enable the Web IoT environment. The NodeMCU sends allthe information to the Raspberry Pi and ThingSpeak server to store and remotely visualize. The overall flow chart of the proposed system. The sensors' initialization is checked to confirm that all sensors output are available. First, the Bluetooth dataare sent to the main controller, which sends the data through Wi-Fi to the Raspberry Pibased and ThingSpeak based cloud server. In the server, first the Geo-fence is checkedusing the GPS and Wi-Fi RSSI to find out whether or not the susceptible/patient crossedthe mentioned Geo-fence area. If the Geo-fence area is crossed, the server sends the alertmessage to the susceptible/patient and the caretaker organizations. After conversion toproper scales and units, the data are visualized in the user web-application.Since GPS has limitations in an indoor environment, we therefore integrate the Wi-FiRSSI level with GPS to check the Geo-fence area and actuate alarms in case the Geo-fence iscrossed. In an indoor environment, we measured the distance from the Wi-Fi Access Pointusing the RSSI level. The centralized database i.e., Raspberry Pi is connected to multiplenodemcu for multiple patients' monitoring. The application peripheral interfaces (API)of Raspberry Pi allows for extracting real-time wearable sensors' data. These data arestored corresponding to multiple patients ID, age, etc. Using machine learning algorithms,the patient condition is classified which is stored for maintaining historical records ofpatients' health. The classified patient condition and real-time sensors data are thensent to a centralized client-based application where doctors and physicians will analyzepatients' health and recommend precautionary measures according to his/her health.These precautionary measures can be accessed by patient and patients' guardians. In theElectronics following subsections, we explained about the breathing rate estimation, oxygen and heartrate estimation, and cough rate estimation.



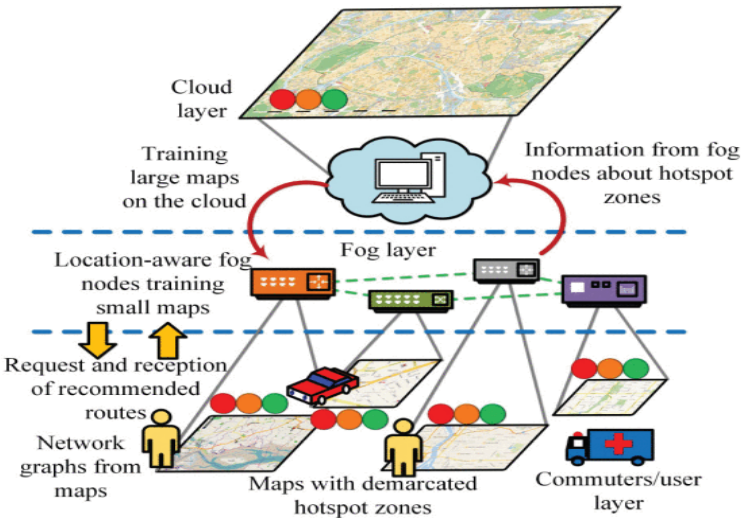

**Paper Title : S-nav: Safety-aware IOT navigation tool for avoiding covid-19 \ hotspots**

**Author Name : Misra, S., Deb, P.K., Koppala, N., Mukherjee, A. and Mao, S.**

**Publication : 2020**

**Methodology : S-Navigation**

In this article, They present a Q-learning-enabled safe navigation system-S-Nav-that recommends routes in a road network by minimizing traveling through categorically demarcated COVID-19 hotspots. S-Nav takes the source and destination as inputs from the commuters and recommends a safe path for traveling. The S-Nav system dodges hotspots and ensures minimal passage through them in unavoidable situations. This feature of S-Nav reduces the commuter's risk of getting exposed to these contaminated zones and contracting the virus. To achieve this, we formulate the reward function for the reinforcement learning model by imposing zone-based penalties and demonstrate that S-Nav achieves convergence under all conditions. To ensure real-time results, we propose an Internet of Things (IOT)-based architecture by incorporating the cloud and fog computing paradigms. While the cloud is responsible for training on large road networks, the geographically aware fog nodes take the results from the cloud and retrain them based on smaller road networks. Through extensive implementation and experiments, we observe that S-Nav recommends reliable paths in near real time. In contrast to state-of-the-art techniques, S-Nav limits passage through red/orange zones to almost 2% and close to 100% through green zones. However, we observe 18% additional travel distances compared to precarious shortest paths.




Network architecture for the S-Nav system.