Microsystem Technologies
https://doi.org/10.1007/s00542-018-3936-9

TECHNICAL PAPER

Unified framework for IoT and smartphone based different smart city related applications

Joy Dutta[1] · **Sarbani Roy**[1] · **Chandreyee Chowdhury**[1]

Received: 18 January 2018 / Accepted: 2 May 2018

Abstract

By embracing the potential of IoT and smartphones, traditional cities can be transformed to smart cities. The success of such smart city mission is firmly vested in populace and thus it should have a bottom-up nature, initiated by the citizens. This paper focuses on the design and development of a unified framework, which can provide a platform to empower all the applications across different dimensions of urban life in a smart city. The aim of this framework is to connect citizens, data, knowledge and services related to IoT as well as smartphone based applications. Here, we categorize all the applications for the smart city in three representative types, viz. *IoT* based, *IoT and smartphone* based and *smartphone as IoT* based applications. We have also developed and tested one prototype following this architecture for each of these three representative category type, i.e, *IoT* based smart classroom, *IoT and smartphone* based air quality monitoring system and only *smartphone* based noise monitoring system to demonstrate the effectiveness of the proposed framework for the smart city scenario.

1 Introduction

The process of developing a smart city involves the interaction of different stakeholders like city authorities, public and private sectors, as well as the citizens. For better urban governance, the need to plan and implement smart applications for citizens is becoming evident. Wireless sensor networks (WSNs) have already been applied in different applications related to smart city like environment monitoring, ambient assisted living, infrastructure monitoring, transport monitoring, etc. In this context, internet of things (IoT) will also play an active role by connecting and enabling devices to the internet (Jin et al. 2014). Moreover, using the smart-phones, smart wearables, citizens can provide data as well as receive information for better awareness of their surroundings. Aggregating information from these various sources with the goal of knowledge extraction is a major challenge of smart city applications. In this respect, citizens participation provides a key input for intelligent decision and policy making. As a matter of fact, the day is not very far that data will be costlier than the devices itself. Among the various data sources of the present world, the prime data sources that we need to mention is IoT, WSN, smartphones and crowd. Since all the devices generate different types of data, different format along with different rate and thus to handle this heterogeneity, we need a unified framework where these various data sources may fit in. There are many context dependencies between these various sources, which are often both logically and physically linked with one another. Moreover, there exist complex correlations between the different data sets generated from various applications related to smart city. To handle this an architectural framework is proposed here which not only fits this different type of applications, but can also seamlessly incorporate data and infrastructure resources available in the underlying environment. In reality Smart city related data is heterogeneous and can be structured, semi structured or unstructured in general (Nastic et al. 2014). Heterogeneity of data comes from the heterogeneity of sensors generating these data. Thus, sensor fusion is very important in terms of smart city realisation as well as for the proposed framework (Dutta et al. 2018). Sensor heterogeneity is also a

---

✉ Sarbani Roy
sarbani.roy@jadavpuruniversity.in

Joy Dutta
joydutta.rs@jadavpuruniversity.in

Chandreyee Chowdhury
chandreyee.chowdhury@gmail.com

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

reason that motivates to design an interoperable framework so that new sensors required by emerging smart city applications can also be added easily. Furthermore, cloud platform can be utilized to handle this smart city's data diversity and still provide a unified analytics layer such that meaningful information can be extracted by mining these data. Actually, cloud computing includes so many aspects of computing almost everything which is needed that hardly a non-cloud solution is able to provide (Datta et al. 2016). The motivation behind designing a framework to cater these different types of smart city applications and consequently our contribution in this work are discussed in the following subsections.

1.1 Motivation

Need for a basic unified architecture for smart city based applications is not new but a proper solution is not there which can map real world smart city applications to the standard architecture (Datta et al. 2016). High availability of smartphones, tablets, low cost sensors and anything as cloud services in terms of XAAS (X as a Service) speeding up the smart city initiatives taken by different government agencies. In reality, smart city is an overall scenario, not any single application rather a pool of applications, which is impossible without citizens. Actually, various applications for handheld devices are already available for making our everyday life simpler (e.g. Google map with live traffic, nearest petrol pump, cab availability, nearby parking spaces, etc.). These applications utilizes the power of hand held smart devices, e.g. smartphones, tablets etc. where we are utilizing smartphones built-in sensors and dumping that sensed data to the cloud infrastructure where meaningful insights can be extracted from mining gathered data. This is one aspect of the smartphone based applications.

However, another important aspect of a smart city is the citizen. Rather, citizen is the most important part of it as these applications use sensor data from smartphones of citizens and also uses the concept of human as sensor. To understand the problem, lets take an example, say the maintenance of a city. Local authorities daily deal with the poor condition of roads, misplaced garbage, non-functioning street lights, hanging electrical wires, waste water and so on. These are all urban problems that we daily encounter. The first step towards addressing the problem is the identification of the concern. In this situation, we, citizens can actively take part by reporting these issues to the authority by using their smart-phone apps. This is encouraging researchers for crowdsensing based applications (Ghosh et al. 2018).

The other class of applications includes IoT based applications in which variety of things (sensors) can be able to interact and cooperate with each other to achieve a common goal. Smartphone can also be a part of any IoT based system as an end user system's output display gadget or can even participate in the system as a part of it. Some relevant smart building related solutions based on sensors have been described in Ghayvat et al. (2015) and Lee et al. (2016). Most of the work relating to IoT architecture has been from the WSNs perspective (Castellani et al. 2016).

However, larger user base can generate huge amount of data, making its storage and timely retrieval a challenging task. Redundant data and energy starvation are other issues related to scalability of such applications. Cloud provides the storage infrastructure to manage this voluminous data collected from citizens along with a high performance computing environment to execute the logic, thus reduces the cost of managing and maintaining the data. Few frameworks can be found for such applications as in Ra (2012), Farkas et al. (2015), Jayaraman et al. (2012) and Gubbi et al. (2013). However, interoperability, privacy and sensor fusion still remain a challenge. In Gubbi et al. (2013), a cloud enabled framework for IoT is proposed that allows networking, computation, storage and visualization themes separate thereby allowing independent growth in every sector but complementing each other in a shared environment. In Geoffrey et al. (2012) and Antonic et al. (2014), IoT-cloud based framework is designed for monitoring tasks with wireless sensors as things and in Ganti et al. (2011), device selection is made at cloud for monitoring tasks. So, the motivation lies within these works, which try to suggest a proper architecture for any IoT enabled smart city based application considering the real-time response requirement for most of the applications.

Thus, both IoT and smartphone based smart city applications demand for an architectural framework to support pluggable smart devices (for both sensing and actuation), scalability and cloud based data analytics and feedback. However, the integration of these two separate application domains into a unified architecture could not be found in existing literature.

1.2 Contribution

In this work, a unified architectural framework is presented to cater the three broad categories of smart city applications, namely, (1) IoT based applications using sensing devices as Category I, (2) applications involving both IoT and smartphones as Category II and (3) smartphone based applications as Category III. To support these different category applications, the framework takes care of interoperability and utilizes sensor fusion to identify patterns by mining data collected by a heterogeneous group of sensors. This architecture ensures online monitoring as well as encouraging crowd participation by various related information delivery after data processing in the cloud. Use of fog computing for reducing data traffic overhead is also considered by the framework. Here, three case studies are presented corresponding to the three categories of applications supported by the framework. Example of first one is on smart classroom, which represents an IoT enabled application on cloud. The second one is crowdsensing based air pollution monitoring application which is a combined application of IoT and smartphone. Finally, last one is Smartphone based noise pollution monitoring system for the smart city. Integration of these three types of applications through the proposed framework is also shown in this paper.

1.3 Organization

Section 2 discusses existing architectural frameworks for developing smart city applications. Then Sect. 3 gives the overview of these representative applications. The architecture of the proposed framework is detailed in Sect. 4. Data management in the cloud is discussed in Sect. 5. Section 6 presents the implementation details of the said applications. Section 7 evaluates the proposed system and finally Sect. 8 concludes this paper.

2 Related work

Smart city applications are mainly designed using IoT and smartphones (Jin et al. 2014). Though both these application categories basically follow a client server architecture having a data collection layer (client end) and use of computation layer (server end) for mining data, research in these two directions have mostly progressed separately with little overlaps. However, an interoperable framework is necessary that may handle the two application categories. In smartphone based smart city applications, the major underlying principle is crowdsourcing and crowdsensing. Mobile crowd-sensing is defined in Ganti et al. (2011) as a category of applications where individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest. Crowdsourcing, in addition, brings human (crowd) intelligence to decision making. These may link infrastructure to its operations in smart cities. Few works are done on designing frameworks for crowdsourcing applications in the smart cities. Many of them are based on publish-subscribe communication paradigm that is the smartphone users are publishing data as well as consuming the services. Few have extended the XMPP protocol in this context as in Farkas et al. (2015). Medusa (Ra 2012) is a programming framework for mobile crowdsensing, it is one of the initial attempts in this domain. It uses a high level XML-based domain-specific programming language, called MedScript. Here the crowd is the producer as well as the consumers of data also known as prosumers. Producers publish raw data to the event nodes. Service Providers intercept these data by subscribing to the raw event nodes, analyze it asynchronously and publish cleaned up information or value added service to the content nodes. Here-n-Now (Jayaraman et al. 2012) is an opportunistic crowdsensing framework that collects data from citizens and correlates mobile sensory information to information from social media in real-time. Localized analytics are a big advantage of this framework as only relevant data is uploaded to the crowd. In Hariri et al. (2013), silent mobile sensing framework is proposed where the cloud server after statistical analysis, publishes data on top of maps for better visualization. In this work, the authors also considered data compression before uploading to the cloud server. In Louta et al. (2016), a comprehensive survey of mobile crowdsensing architectural frameworks is presented in the smart city scenario. A closer look at state-of-the art reveals that smartphone based sensing applications like mobile crowd-sensing are still in its infancy as there is neither a unified data collection framework for collecting data from heterogeneous sources (different smartphones have different sensors with varying calibration) nor the data interpretation mechanism is uniform and most often not interoperable as well. Towards this, in Datta et al. (2016), a battery aware crowdsensing framework is proposed based on one M2M standard architecture.The framework takes sensory input from smartphone sensors as well as sensors built-in vehicles and stores it in a semistructure database to take care of heterogeneity. Further, sensor meta-data from heterogeneous domains are treated using semantic web technologies so as to incorporate a fusion of heterogeneous sensor data. IoT based smart city applications are also coming up where a number of heterogeneous sensors contribute data to the server side for analysis. In Zheng et al. (2013), the authors analyzed data from air quality sensors that are deployed at major road crossings and important areas of a city. They investigated the effect of meteorological conditions, location, human mobility on air pollution to find a spatial correlation of air quality in various places of a city. Artificial neural networks are used here to classify various areas. Many indoor applications of IoT can also be found in literature. In Ghayvat et al. (2015), a smart home management system is presented in which a community broker role is used for integrating community services, thereby reducing the workload of community management staff, providing electronic information services, and deepening the communitys integration with the surrounding environment.

Many of these applications do not utilize cloud, though in Gubbi et al. (2013), the importance of integration of IoT and Cloud is highlighted to provide computational and storage infrastructure and support the development of services and applications beyond the limits of conventional IoT. Thus, developers using IoT-Cloud can easily develop compute intensive services and applications for data analysis and decision reporting. IoT-Cloud could be a strong pillar to extend smart applications to people at affordable cost. Consequently, in Geoffrey et al. (2012), a cloud compatible open source messaging system is introduced along with APIs that allows developers to write efficient IoT applications. For monitoring applications, device information can be utilized to select deployed devices based on application purpose and device status (available energy, location etc.). In Antonic et al. (2014), a publish/subscribe middleware is proposed based on cloud that filters out redundant data by selecting minimum no. of devices for sensing/monitoring a particular area.

3 Representative applications for smart city

Smart city applications can be grouped into three broad categories, namely, (1) IoT based applications using sensing devices as Category I, (2) applications involving both IoT and smartphones as Category II and (3) smartphone based applications as Category III. In this section, three representative applications from each category are described where smart campus is chosen to be the underlying scenario of these applications. The applications are smart classroom, air quality monitoring of the classroom and noise level monitoring of the classroom. We have chosen the above mentioned applications from a smart campus point of view as a smart campus is a small representation of a smart city.

3.1 Smart classroom: IoT based application

In a smart campus scenario, building smart classroom is of foremost importance. It can also be considered as part of a smart building (Dutta and Roy 2017), that can significantly improve the institution's ambience as well as facilities available to students, teachers and administrative body. Thus, the main objective of this application is to utilize the classroom resources in a favorable way while maintaining the convenience of people (students and teachers). With the help of IoT, the management of a classroom related resources can be automated. Radio-Frequency IDentification (RFID) technology is one important building block for IoT based applications, which is used to automatically identify the people and objects (things). As mentioned earlier, another foundational technology of IoT is WSNs, where sensors are deployed to periodically sense the environment and send the information to the sink node. Along with these, many other technologies like Zigbee, Bluetooth, WiFi, cellular networks, fog and cloud computing support IoT to compose such applications. A prototype of simple and low cost, smart classroom using these technologies has been built. The integration of IoT, fog and cloud is done using open source hardware and software. In this application, electrical appliances like light and fan of the classroom are controlled using only a smartphone or any mobile devices (e.g., tablet, laptop etc.) or even using a PC also. Our envisioned smart classroom consists of smart RFID based entry. This application also has the inbuilt intelligence to respond back depending on environmental conditions. For example, automatically turn on the light based on student's presence in the classroom or automatically turn on fans based on room's present temperature.

3.2 Air quality monitoring (AirSense): IoT and smartphone based application

Most of the smartphone based applications either employ crowdsourcing or crowdsensing utilizing smartphone sensors of citizens. However, a few can also be found that takes input from other sensor devices (like wearable sensors) but collect data through the smartphone interface of citizens. Here, in this work, we have considered an opportunistic crowdsensing based application, AirSense (Dutta et al. 2016). The main objective of this application is to monitor the air quality in the vicinity for both indoor and outdoor scenarios. AirSense would encourage the citizens to participate in a crowdsensing initiative (Dutta et al. 2017), which could be a backbone of any smart city. The nature of the crowd-sensing used in this work is categorized as opportunistic sensing (Louta et al. 2016) where, user involvement is minimal to ensure reliable data at regular intervals. A portable Air Quality Monitoring Device (AQMD) is designed that interfaces with smart handheld to communicate data to the cloud. AirSense can be used for both personal and collective sensing, based on the circumstances being experienced. With the help of collective sensing paradigm, AirSense can provide air quality data with enhanced temporal and spatial resolution. As opposed to traditional, expensive, heavy but highly accurate air quality measurements (Zheng et al. 2013), the use of AirSense based on low-cost and light weight sensors is very appropriate in todays world. Most importantly, the issue of obtaining accurate and spatially resolved air quality data is achieved through the participation of people in AirSense. In the present version of the AirSense application, data is collected through registered user (volunteer) who carries the AQMD and a smartphone. However, any registered user who is not carrying AQMD can also see the Air Quality Index (AQI) of a nearby location and also the AQImap of the city on their smartphone.

3.3 Noise monitoring (NoiseSense): smartphone as IoT based application

This is a smartphone based crowd sourced application which basically exploits the power of smartphone's inbuilt sensors and dependent on the smartphone alone. In this application we have used device's microphone for noise monitoring and collected raw data is sent to cloud for further analysis (Dutta et al. 2017). There is inadequate stations in the city that can provide us city's noise pollution map and also can't provide us our individuals' pollution exposure. Our system actually gives individual's noise exposure which is very much important for the city perspective as well as to understand noise pollution's effect on individual's health. In the application we are using the smartphone which they are carrying automatically, i.e, no extra effort is needed for participating in this crowd sensing based application. The solution is also a context aware solution which basically cancels out wrong values by understanding the user's context. When we aggregate other user's data in the cloud from the city, we get a noise pollution map of the city which is very much important for the city authority.

In the next section, the unified architecture is proposed and mapped to above three different types of applications which shows that our proposed architecture is efficient enough to handle all three different dimensions of a smart city in the subsequent section.

4 Proposed architecture

The unified smart city architecture is depicted in Fig. 1, which is a descendant of client server architecture. Slicing concept has many applications in software engineering, one of them is architectural reuse (Zhao 1998). In this paper, architectural slicing is used to highlight knowledge about the high-level structure of the smart city related applications, rather than the low-level implementation details. The proposed architecture is a generic description which entails potentially different smart city related application behaviors. As discussed earlier, data sources are connected to the internet directly or via the smartphone or via fog and that push data on the cloud server. To offer a service, applications analyze this data in the cloud. This architecture maps all the smart city based applications to either IoT based applications or smartphone based applications or a combination of both. In this architecture smart citizen (crowd) is one of the most important portion as unless and until the citizens are smart, all other smart applications are of no use. Citizen, sensors, IoT, smart devices, smartphones are the key components of the data layer (as shown in Fig. 1). Basically, this layer is generating the data and is coined as data layer. This layer constitutes of crowd, IoT etc. which is generating data to the smartcity based applications and also responsible for consuming the services provided by the applications. This generalized architecture can be fine grained more according to the requirements of various applications associated to smart city. However, the main difference in the orchestration of this architecture for these three representative applications for each category is in the presence of fog layer. Fog layer can be utilized efficiently for the localized analytics (partial computing) and realtime control of IoT enabled devices (smart classroom application). However, in case of either *IoT and smartphone* based applications (AirSense) or *smartphone* based applications (NoiseSense), localized analytics can be done inside the smartphones used as edge computing node) as nowadays smartphones have high computational power as well as having comparatively higher battery life. Along with this, smartphone is now having sufficiently high memory to aggregate local data such that it can store data for some short duration of time before sending the data to the cloud using its internet connection. Power management can be done efficiently too in the smartphones. Thus, the fog layer is not necessary in case of applications where smartphone is generally involved. Then, we reach to the data analytics portion for the applications which generally takes place in the computation layer. This layer is generally common for different smart city based applications and it is taken care of in the cloud.

For any IoT based application, the thing you find common is the presence of various sensors which are actually sensing the environment. These sensors are sensing the environment and sending this data to the fog gateway for local computation which is actually a gateway to the internet but having computational and storage capabilities. Some part of the computation is done in the fog to make the system environment realtime and as well helps the system to reduce internet traffic as it can decide which data is redundant and which one is necessary. In the proposed architecture, data aggregation and power management are two key responsibilities of fog. Now to map the same thing in any classroom environment, we need to make those appliances IoT enabled. Electrical things of any standard room that we want to consider as things of IoT in the smart IoT enabled classroom application are actually not pluggable as it is. To make them IoT things, they have to give the power of sensor, processor (low-power), and a way of communication (usually wireless). In smart real life applications, these things may connect directly to the smart phones or nearby Internet gateway devices. The prototype is built using Arduino Uno, sensors, Bluetooth module,

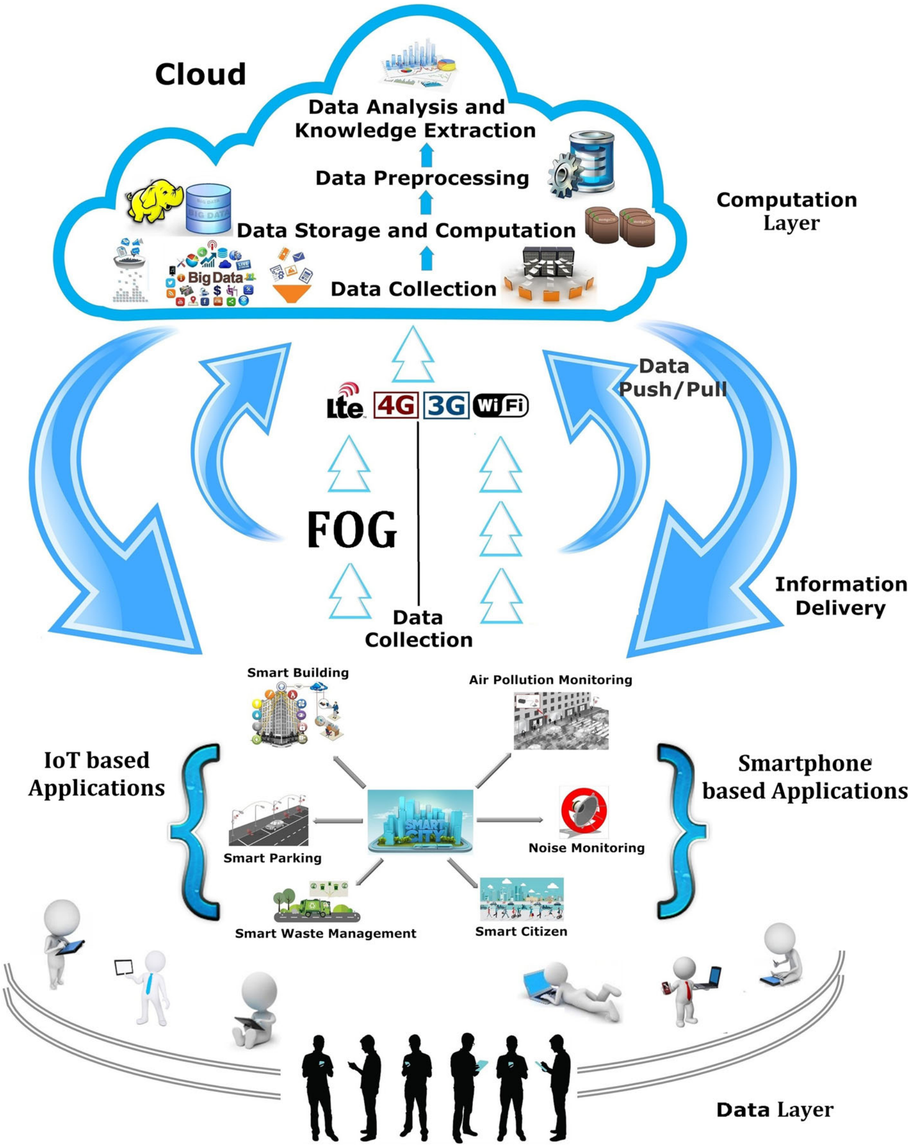


Fig. 1 Architecture for smart city applications

WiFi module, fog gateway server, the cloud platform (for online data storage and analysis) to control classroom electrical appliances from anywhere in this world. The proposed smart classroom system is designed as a three tier architecture (IoT slicing) as shown in Fig. 1. Description of each layer is given below.

1. *IoT layer* In tier 1, sensing and communication capabilities are incorporated within the physical things of environment to get them ready for IoT. A microcontroller is used for controlling things equipped with sensors and communication units. Now, these things can sense the environment and also can communicate with other devices using Bluetooth or WiFi. In smart classroom application, multiple such things can be present in close proximity. Thus to manage a group of high voltage electrical devices by Arduino, a relay module is used.
2. *Fog layer* In tier 2, the fog gateway server is maintained, which can take some local decision based on sensors data. This layer is responsible for collecting, filtering and preprocessing the data. It also works as an Internet gateway which actually connects IoT with Internet using VPN.
3. *Cloud layer* In tier 3, data from the fog layer is forwarded to cloud for storage and computation. Here, a database is maintained for cloud server, which is used for storing sensor data for controlling the IoT or triggering some action based on the analysis of sensed data or by user need from smart phones or computer. Apart from mining data, profiling residents of smart homes or usage pattern of smart classrooms can also be done in this layer.

For smartphone based applications which includes applications related to IoT also where smartphone is a part of the system for computational purpose, there we do not need fog in the architecture as mentioned earlier, as those computation can be done in the smartphone. Even if fog is present in the structure, it will work as an Internet gateway. Here how AirSense or NoiseSense (described in Sect. 3) is mapped with the generalized architecture (smartphone slicing) is shown in Fig. 1. The main difference here is that, as smartphones have appreciable computation, communication and sensing ability, both data collection and data preprocessing can be done in the smartphone itself. Thus the two functionally predominant layers are data layer (Crowdsensing) and computation layer (Cloud) with fog layer only acting as an internet gateway.

5 Data management

Data is the most important part of any smart city application. Data from smartphone sensors or IoT sensing devices in a vast variety of geographical locations along with the data generated due to the participation of citizens increased significantly. Major steps of data management in the proposed system are discussed here. This is a common part for all the smart city related applications.

(a) *Data collection* In the proposed framework data sources (smartphones, tablets, things in IoT) comes in the lower tier and cloud is in the upper tier. So, functionalities are also divided in that fashion. The data generated and collected by mobile devices will be stored on the cloud. Android based mobile applications are developed for the proposed applications. Also, users can give individual experiences via his/her smartphone.

(b) *Data preprocessing* After getting raw data from the mobile devices, we need to pre process that data before further analysis or data mining. Since data will likely be imperfect, containing inconsistencies and redundancies and may not be directly applicable to the data mining process. Actually, data preprocessing is an important step of Knowledge Discovery from Data Process (KDD) that can enable the user to treat and process complex data. Here tasks like data cleaning, missing values imputation, etc., are applied in this work. Data cleansing is performed interactively through scripting.

(c) *Data processing and analytics* Now, the proposed system can efficiently process the data after preprocessing and give real time response to the users. Our requirement here is to provide the storage facility to all these real time and large volume of data. Here we have utilized Platform As A Service that provides an environment to store, aggregate and analyze stored sensor data in the cloud server. Actually, storing and protecting data, and making it accessible and available, requires some heavy-duty data management. Analytics include a variety of techniques for finding patterns and relationships in the data. Presently, our focus is mainly on predictive analytics in order to better understand and respond to citizens needs as well as making intelligent systems for the smart city. Predictive analytics is the practice of extracting information from historical and current data sets in order to determine patterns and trends, which can be used to anticipate future needs and accordingly reorganize service delivery. This type of analysis also enables individual recommendations and maximizing the value of every user interaction.

6 Implementation

(a) *Smart classroom* In this application, electrical device identification is a crucial part as there are many classrooms in a building, there are multiple floors, in each floor there are multiple classrooms and there are multiple electrical appliances. So, to solve this issue, we have used a single Arduino board for each room and given each board a unique ID. Now, to identify an electrical device uniquely we generate a Thing-ID which is composed of two parts, i.e, Board-ID and Pin. Here, the pin is the board pin of Arduino at which the appliance is attached via Relay. So, as a result, the user request is generated in tuple<Thing-ID, Request>. For practical realization of the smart classroom, general Circuit diagram of the smart classroom is shown in Fig. 2.

Authenticated users can control the classroom devices through smart phone using Bluetooth connection. We maintained our server in the cloud to store and update the status of the devices that we are using in the smart classroom application. The state of the switch is reflected in the application using an event trigger mechanism implemented on the server side. The status of each device is stored in the cloud database as well as in the fog gateway and updated whenever there is any change. For example, if the user changes the state of any device locally through the smart phone application, the information is pushed into the fog and then into the cloud with the single button click event. This change will also reflect on the Android application and PC interface. Whenever there is a request to change the status of any IoT device remotely, that request is sent to the cloud server, the server forwards this request to the fog

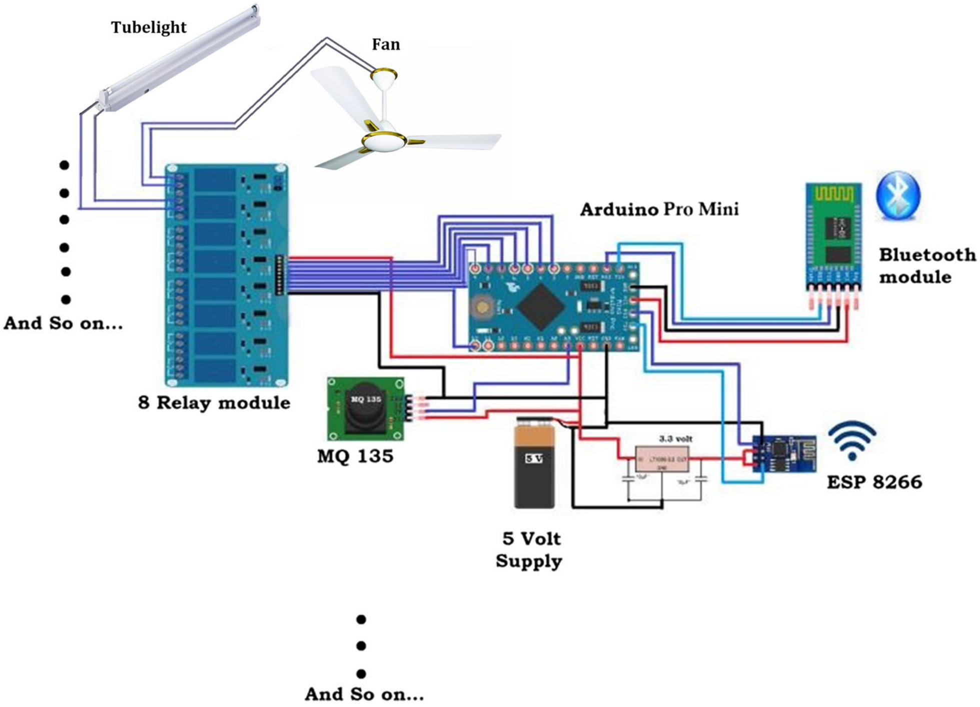


Fig. 2 General circuit diagram of the smart classroom

server, which generates IoT status change request for the respective IoT enabled devices. Similarly, when the status of the IoT enabled device is changed in reality, then its changed status is forwarded to the fog, as well as in the cloud and also updated in the database and after that this change is reflected in the Android application. In the proposed architecture the presence of fog has a very significant role. In this implementation, we have used a gateway node as a fog server. There are some scenarios where we can avoid pushing data to the cloud and decrease the systems overall response time. Fog server can reduce the internet traffic by handling many aspects locally using its computation and storage power. Thus, it makes the system more agile and responsive (real time).

(b) *AirSense* As mentioned earlier, AQMD is the sensing device, which is used for monitoring the air quality in the vicinity. Though it is evident that more sensors (e.g., PM10, PM2.5, O3, NO2, SO2, etc.) we will be using, the more accurate data we will get but the cost of the device will be increased. So, to build the device at low cost, we have used important primary air quality sensors like MQ-135 for hazardous gas detection and MQ-7 for CO, which are connected to the Arduino Pro Mini board. The MQ series of gas sensors uses a small heater inside with an electro-chemical sensor. The output is an analog signal and can be read with an analog input of the Arduino. To add the Bluetooth capability on AQMD, Bluetooth module HC-05 is used. Arduino Pro Mini has an interface between the sensor and the Bluetooth module. The circuit of the device is shown in Fig. 3 and it can be attached to the key ring or bag as shown in Fig. 4. It weighs only around 69 g.

The mobile application component of the AirSense is an Android application that is responsible for (1) data collection from AQMD, (2) forwarding data to the cloud service, (3) providing AQImap and (4) providing individual pollution footprint. The first task is to check the presence of a Bluetooth adapter. If the Bluetooth adapter is present and enabled, it checks for paired/bonded devices, as the device must be paired before the application can use it. Here, the device IDs are being compared for a match. After getting the AQMD, a socket has to be created to handle the outgoing connection. Since the data from AQMD can be received at any point of time, so a thread is running to listen for incoming data. Geocoding API is used to get the location of the smartphone. For indoor, predefined location points are used. Next, to forward the data to the cloud, WifiManager API of Android is used to manage all aspects of WiFi connectivity. However, 3G/4G connection may also be utilized instead. Mobile application provides local air quality data collected from the personal AQMD (or AQMDs in the vicinity). It also generates AQImap of the city from the collected data.

(c) *NoiseSense* Implementation of noise monitoring service involves implementing a mobile application and a server end. A mobile application using the Android platform was implemented. Android Studio

Fig. 3 Airsense circuit

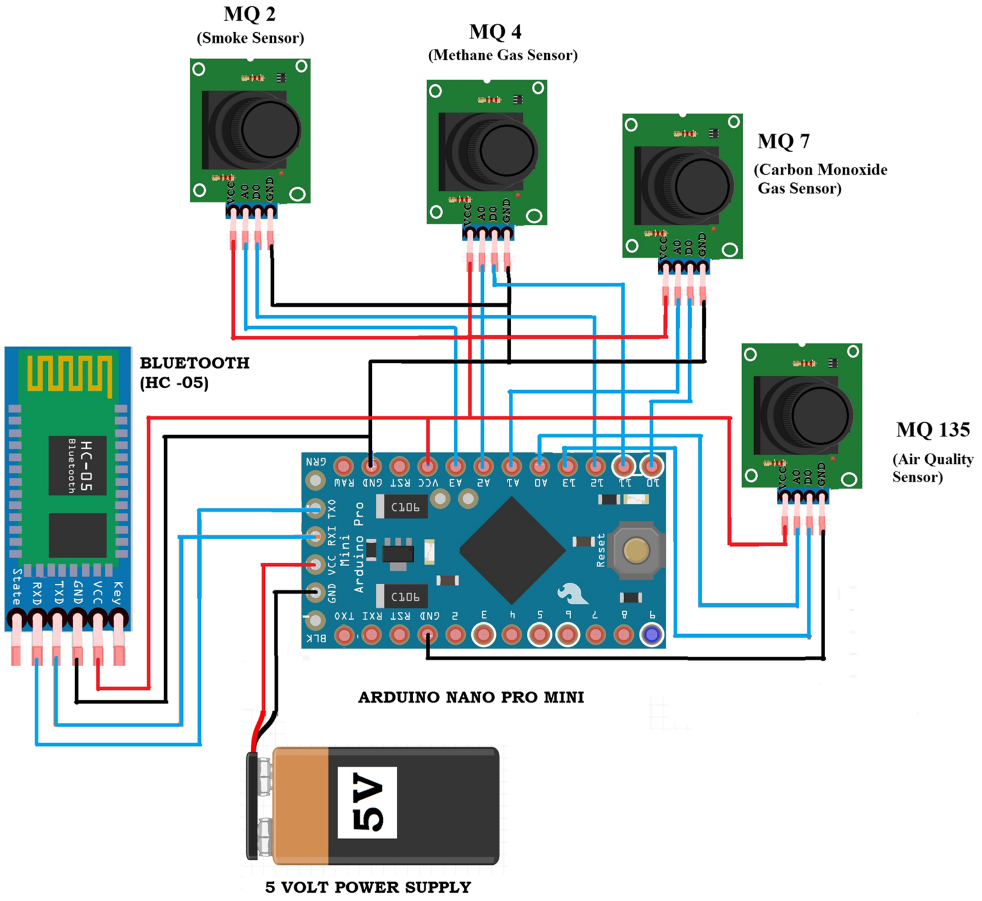
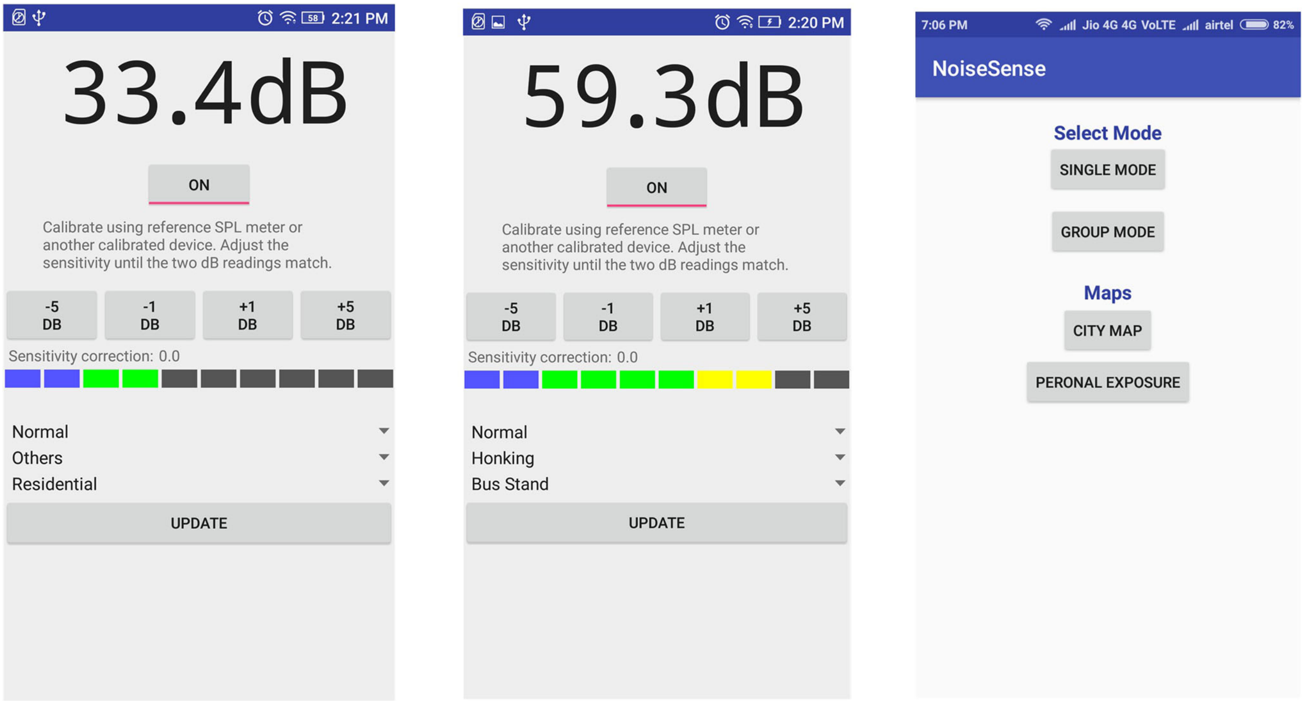


Fig. 4 Airsense in practical world

2.3.1 was used as IDE for both design and development of the application. In Android Studio user interface elements are generated using XML and Java is used as a programming language. We have used smartphone's microphone for the noise data collection purpose. The screenshot of the application is shown in Fig. 5.

Internal computation of the NoiseSense within smartphone is shown in Fig. 6. We have converted the collected data in the decibel format, used auto calibration technique for precise noise data collection, made the system context aware using proximity sensor and call monitoring while the application is active and used Google's SPL to get the noise value from the smartphone. Rest of the things (push collected data to cloud, data processing, pollution map generation etc.) are similar like AirSense application.

(d) *Data management* As mentioned earlier, in all the applications, viz, smart classroom, AirSense and NoiseSense applications, OPENSHIFT (Openshift Platform http://www.openshift.com) is utilized as a cloud service provider that provides an environment to store, aggregate and analyze stored sensor data in the cloud. The main functions of the cloud layer are data storage, processing, data analysis and giving feedback to the user. Authentication check for the user is also done here for maintaining a personal profile. So, this layer basically stores all the user related data, including authentication and sensed data from the application along with location and time stamp. This data is sent using the POST method of the HTTP protocol. Each time such a http post



Calibrate using reference SPL meter or another calibrated device. Adjust the sensitivity until the two dB readings match.

Sensitivity correction: 0.0

Normal
Others
Residential

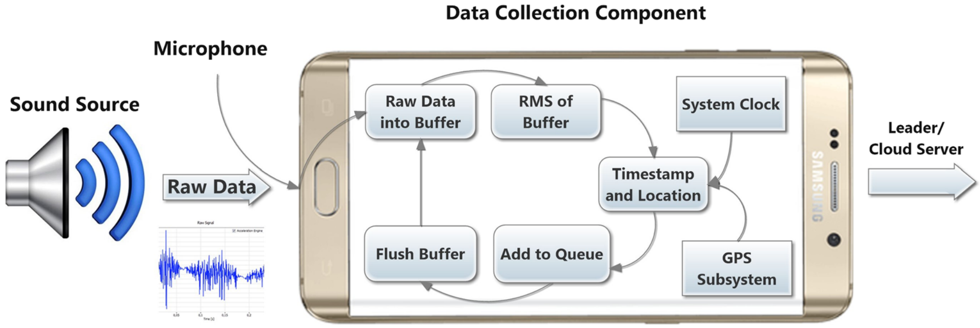
Calibrate using reference SPL meter or another calibrated device. Adjust the sensitivity until the two dB readings match.

Sensitivity correction: 0.0

Normal
Honking
Bus Stand

Fig. 5 NoiseSense application screenshot

Fig. 6 NoiseSense internal processing



request comes from our application, this layer is responsible for extracting information from the request message and insert the same into the database. Heterogeneous data are often generated from IoT and smartphone based systems. To handle the issue of heterogeneity we have used noSQL database MongoDB which is supported in OPENSHIFT. It supports both SQL and NoSQL. By the nature of NoSQL database, MongoDB is flexible and can handle heterogeneous nature of data. Now these heavy-duty data management can be handled well with Hadoop. Hadoop is an open-source framework to store and process big data in a distributed environment. Although serving analytics from Hadoop to online applications and users in real time requires the integration of a highly scalable, highly flexible operational database layer, it is very much essential when data becomes huge in size. In the backend, we have used PHP 5.4 as it has built in web server which handles requests sequentially and ideal for quick testing as well as unit testing of web services with the addition of new session handling class along with MySQL and MongoDB as our database. In the front end we have used Java Script along with HTML and CSS. In smartphone, both data cleaning and noise removal are done to get better accuracy of the sensed data.

7 Evaluation

In this case study, we have shown how a *IoT and smartphone* based application (AirSense) integrates with an *IoT* based application (smart classroom) along with a *smartphone* based application (NoiseSense) via the proposed architecture. To evaluate AirSense and NoiseSense along with the smart classroom, AQMD and smartphone is used in the classroom along with other sensors like smoke detection sensor, PIR sensor, LDR etc. AirSense collects rooms air quality which is a part of classroom monitoring and NoiseSense collects noise level of the classroom. Figure 7 shows the indoor AQI and noise level for smart classroom scenario. In this experiment, we have combined AirSense and NoiseSense applications with smart classroom application. In smart classroom application, we have used various types of sensors according to our need where some of the sensors (e.g. PIR, ultrasonic, temperature etc.) are used continuously where as some sensors (e.g. RFID, LDR etc.) are used occasionally which are generating different types of data (e.g. boolean, numeric, string, data and time etc.). On the other hand, in AirSense, we have used different air quality sensors in AQMD, where the output is an analog signal and then converted to AQI value. Smartphones microphone sensor is used to sense the noise level in NoiseSense application. Data collection rate of these applications are also different. If AQI or noise level is

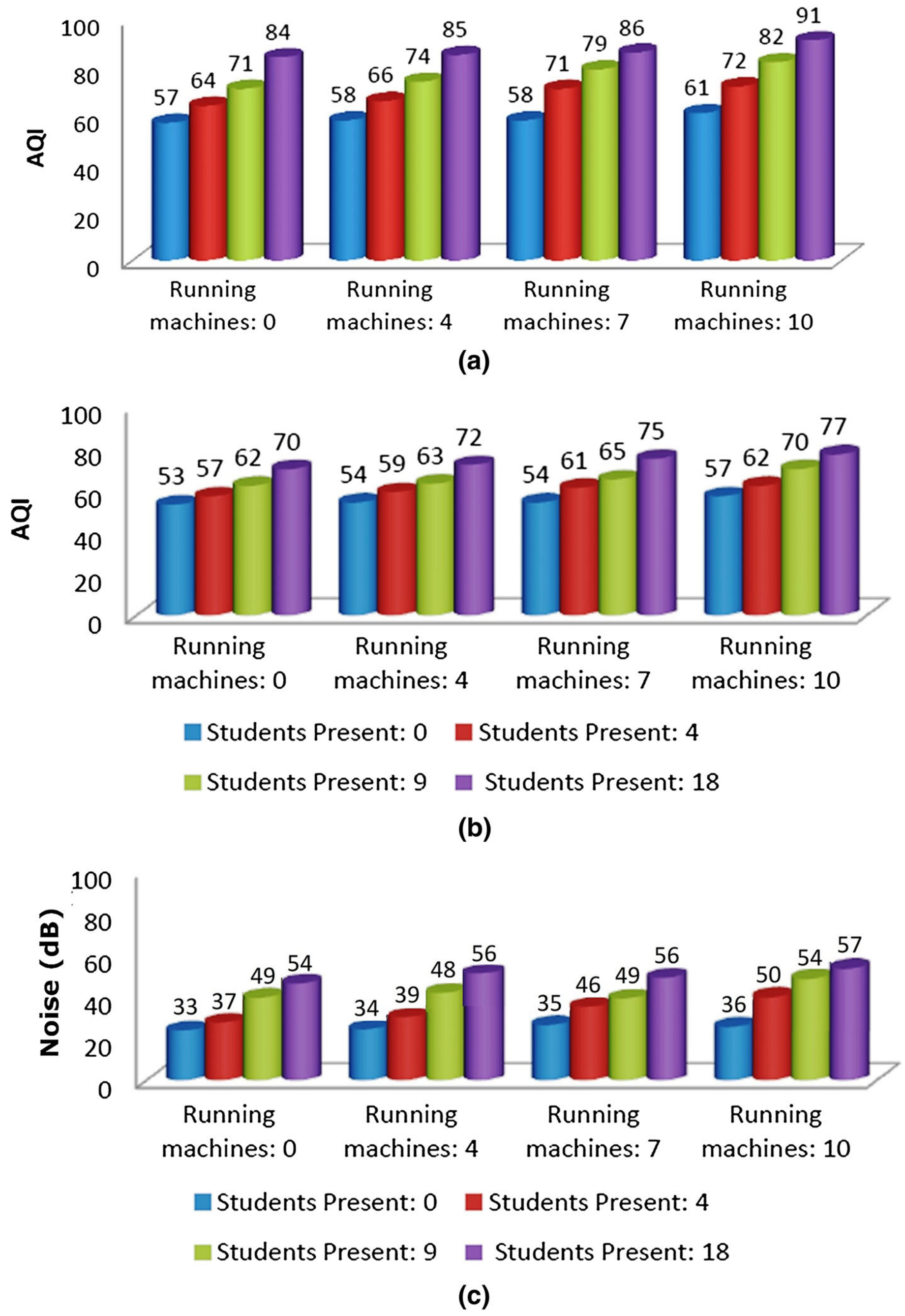


Fig. 7 AQI and noise level data obtained under different conditions in our laboratory setup for two scenarios. **a** Air purifier switched off; **b** air purifier switched on in fan mode and **c** corresponding noise level in the classroom

above threshold value then data will be collected continuously otherwise periodically. The average hourly value based on different data collection rate after the analysis in the cloud is shown in Fig. 7. For these applications, we have done various experiments with the test cases. The specific classroom that we have taken in our experiment, can hold up to 20 students at a time.

Since, our experiment space is a computer science laboratory, plenty of desktops and laptops are also there contributing to heating and noise. In this work, the values of AirSense and NoiseSense have been collected by varying number of students and running machines, which are stored in the cloud. Each data entry is an average of AQMD readings of a cycle of 5 min. If the AQI value is found to be above threshold (in this case 85) then it switches on the fan for better air circulation. We have also considered the empty classroom scenario. As shown in Fig. 7, the room becomes suffocated with more students and that is reflected in the AQI of the room. From Fig. 7b it can be observed that AQI is much better than the usual classroom when air purifier is switched on. Then, when we notice the noise level of the room, we notice that with more students, the room is becoming more noisy and it is reflected in Fig. 7c, because of gathering and whispering. As expected, machines are not contributing to AQI or noise increase but they do increase the room temperature, especially the high end machines.

Now, to make this whole classroom scenario IoT enabled and smart, we make a smart solution using event trigger mechanism where we are using above three mentioned system at a time to give it a proper shape. We take the classroom first, then we added AirSense in it and NoiseSense too. When, in the classroom, the air quality deteriorates (i.e. in the higher end of moderate section of AQI indexing (Air Pollution in the World http://aqicn.org/search/kolkata) and the AirSense system catches the same, we are switching on the air purifier and fan in the room. Similarly, when the noise threshold (set by the admin) is crossed in the NoiseSense system for more than 1 min, students get an automatic notification that the noise is high in the classroom and they need to keep quiet. Thus, both the AirSense and NoiseSense system automates the whole classroom system and makes the classroom smarter. Moreover, to make it even smarter further IoT enabled devices are incorporated with it, e.g., when the first student enters into the room, RFID sensor detects that event and switches on the light. This logic is running in the server end to monitor and act in real time without human intervention. However, a person may override the system and may switch on/off any of these appliances through smartphone application. Prototype built in our laboratory is shown in Fig. 8.

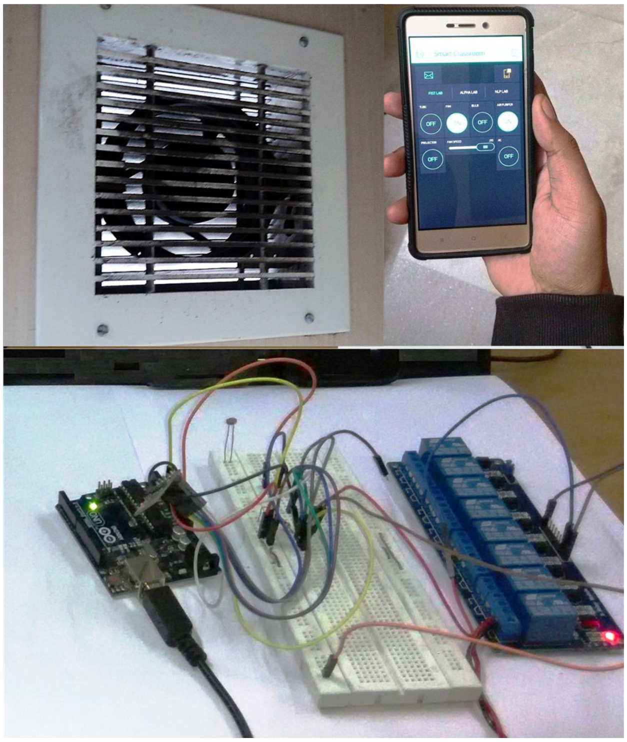


Fig. 8 Prototype of smart classroom: close look

Hence, the architecture followed in AirSense and NoiseSense is Fog independent and can transmit data directly to cloud without any hindrance. But when we take the smart classroom scenario, situation changes as there is a number of heterogeneous sensors. So, sensor generated data is also not negligible when we incorporate audio recording module, camera module etc. which is placed there according to necessity. Also, it is known to us, that huge data bandwidth is an issue and all time availability of that bandwidth is hard to find. So, we have to be very careful about data to send in our cloud server and which portion we can calculate locally. So, there is a trade off between these two. For our application, some immediate responses (e.g. emergency fire alarm or classroom appliances control) are handled locally using the fog concept and rest of the data are processed in the cloud. Although, some of the decision/responses are taken locally in the fog, still a record is always saved in the cloud too via periodic synchronization. Thus, the proposed IoT-fog- cloud is the ideal architecture for IoT based services based on application's requirement. This requirement includes real time response and quick decision which is based on moderate to low computation. We have tested the same, and from our lab set up, locally the user can control the appliances in real time (18 ms approx, using IoT-fog-cloud) where as remotely the response time is slightly higher (1230 ms approx, using IoT-cloud in setup). These values are averages calculated on 20 different instances.

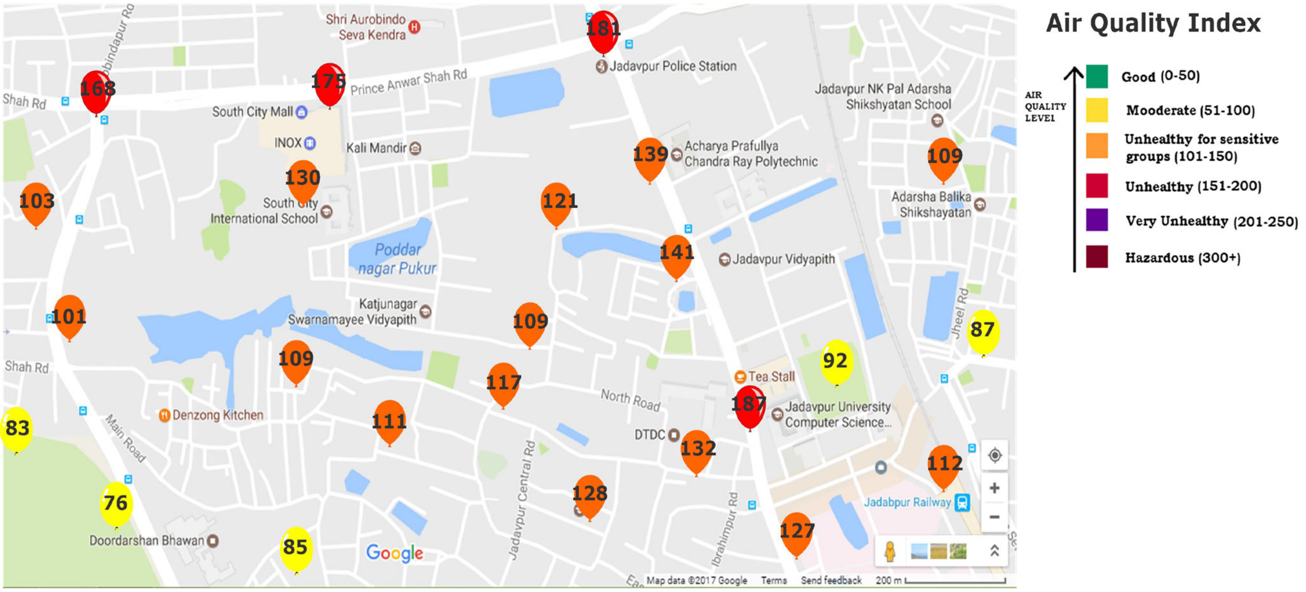


Fig. 9 Outdoor pollution map generated by Airsense for the area surrounding Jadavpur University main campus

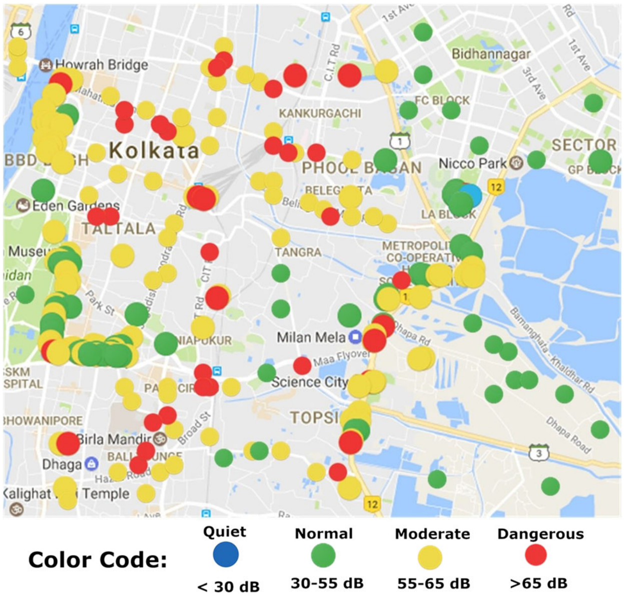


Fig. 10 Outdoor pollution map generated by NoiseSense for the Kolkata City area

However, both AirSense and NoiseSense are two separate and standalone system service which does not require any other system to complete them, but when these two separate services are mapped in one system, i.e., smart classroom, the system became smarter and followed the same proposed architecture which we proved earlier. To give an idea about these two applications standalone nature, we have done some experiment outside the classroom also. AirSense is used to collect AQI data from outdoor environment as shown in Fig. 9. Here, we have taken the result for Jadavpur University Main Campus surrounding area. It has been found that on major road crossings high values AQI are obtained indicating poor air quality. Students volunteered to get these data. The data gathered validates the design of our architecture. Similarly, noise data is collected from Kolkata city and plotted in the map as shown in Fig. 10.

8 Conclusion

Technologies like IoT, smartphone, fog and cloud offer unparalleled opportunities to enhance efficiency and improve urban life. The main purpose of this work is to develop a unified framework for IoT and smartphone based applications for smart cities. Case studies are presented to highlight the effectiveness of the proposed framework. The most important thing here is data and such applications will have a major impact on the amount of data created on a daily basis. In future, this huge amount of data will rule how cities are operated and managed. The proposed framework enables data integration from different sources to enhance the quality and performance of the services, though we have not yet experienced big data. In our future work, we plan to investigate the challenges associated with big data management in the proposed framework.

Acknowledgements The research work of the first author is supported by Visvesvaraya PhD Scheme, Ministry of Communications and IT, Government of India.

References

Air Pollution in the World, Realtime Air Quality Index. http://aqicn.org/search/kolkata. Accessed 30 Apr 2018

Antonic A, Roankovic K, Marjanovic M, Pripuic K, Zarko IP (2014) A mobile crowdsensing ecosystem enabled by a cloud-based publish/subscribe middleware. In: International conference on future internet of things and cloud (FiCloud), pp 107–114

Castellani AP, Bui N, Casari P, Rossi M, Shelby Z, Zorzi M (2010) Architecture and protocols for the Internet of Things: a case study. In: 8th annual IEEE international conference on pervasive computing and communications, pp 678–683

Datta SK, Ferreira da Costa RP, Bonnet C, Hrri J (2016) oneM2M architecture based IoT framework for mobile crowd sensing in smart cities. In: 2016 European conference on networks and communications (EuCNC), pp 168–173

Dutta J, Pramanick P, Roy S (2018) Energy-efficient GPS usage in location-based applications. In: Satapathy S, Tavares J, Bhateja V, Mohanty J (eds) Information and decision sciences. Advances in intelligent systems and computing, vol 701. Springer, Singapore. https://doi.org/10.1007/978-981-10-7563-6-36

Dutta J, Chowdhury C, Roy S, Middya AI, Gazi F (2017) Towards smart city: sensing air quality in city based on opportunistic crowd-sensing. In: Proceedings of the 18th international conference on distributed computing and networking (ICDCN '17). ACM, Article 42, pp 1–6

Dutta J, Gazi F, Roy S, Chowdhury C, AirSense: Opportunistic crowd-sensing based air quality monitoring system for smart city. In: 2016 IEEE sensors. Orlando, FL, USA, 2016, pp 1–3. https://doi.org/10.1109/ICSENS.2016.7808730.

Dutta J, Pramanick P, Roy S (2017) NoiseSense: crowdsourced context aware sensing for real time noise pollution monitoring of the city. In: Proceedings of the IEEE international conference on advanced networks and telecommunications systems (ANTS), Bhubaneswar, pp 1–6

Dutta J, Roy S (2017) IoT-fog-cloud based architecture for smart city: prototype of a smart building. In: Proceedings of the 7th international conference on cloud computing, data science and engineering (confluence), Noida, pp 237–242. https://doi.org/10.1109/CONFLUENCE.2017.7943156

Farkas K, Feher G, Benczur A, Sidlo C (2015) Crowdsending based public transport information service in smart cities. IEEE Commun Mag 53(8):158–165

Ganti RK, Fan Y, Hui L (2011) Mobile crowdsensing: current state and future challenges. IEEE Commun Mag 2011:32–39

Geoffrey CF, Kamburugamuve S, Hartman RD (2012) Architecture and measured characteristics of a cloud based internet of things API. in: International conference on collaboration technologies and systems (CTS), pp 6–12

Ghayvat H, Liu J, Mukhopadhyay SC, Gui X (2015) Wellness sensor networks: a proposal and implementation for smart home for assisted living. IEEE Sens J 15(12):7341–7348

Ghosh S, Dutta J, Roy S (2018) SenseDcity: a participatory sensing based approach. In: Proceedings of the workshop program of the 19th international conference on distributed computing and networking (Workshops ICDCN '18). ACM, New York, NY, USA, Article 16. https://doi.org/10.1145/3170521.3170537

Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (IoT): a vision, architectural elements, and future directions. Future Gen Comput Syst 29(7):1645–1660

Hariri F, Daher G, Sibai H, Frenn K, Doniguian S, Dawy Z (2013) Towards a silent mobile sensing framework for smart cities. In: Wireless World Research Forum (WWRF 30)

Jayaraman PP, Sinha A, Sherchan W, Krishnaswamy S, Zaslavsky A, Haghighi PD, Loke S, Do MT (2012) Here-n-now: a framework for context-aware mobile crowdsensing. In: Proceedings of the tenth international conference on pervasive computing, pp 1–4

Jin J, Jin J, Marusic Gubbi S, Palaniswami M (2014) An information framework for creating a smart city through internet of things. IEEE Internet Things J 1(2):112–121

Lee YT, Hsiao WH, Huang CM, Chou SCT (2016) An integrated cloud-based smart home management system with community hierarchy. IEEE Trans Consum Electron 62(1):1–9

Louta M, Mpanti K, Karetsos G, Lagkas T (2016) Mobile crowd sensing architectural frameworks: a comprehensive survey. In: 2016 7th international conference on information intelligence systems and applications (IISA), pp 1–7

Nastic S, Sehic S, Le DH, Truong HL, Dustdar S (2014) Provisioning software-defined IoT cloud systems. In: 2014 international conference on future internet of things and cloud, 2014, pp 288–295

Openshift Platform http://www.openshift.com. Accessed 30 Apr 2018

Ra MR (2012) Medusa: a programming framework for crowd-sensing applications. In: Proceedings of the 10th international conference on mobile systems, applications, and services, 2012, pp 337–350

Zhao J (1998) Applying slicing technique to software architectures. In: Proceedings of the 4th IEEE international conference on engineering of complex computer systems, pp 87–98

Zheng Y, Liu F, Hsieh HP (2013) U-Air: when urban air quality inference meets big data. In: Proceedings of ACM conference knowledge discovery and data mining, pp 1436–1444

Publisher's Note