SEMANTIC MULTIMEDIA FOG COMPUTING AND IOT ENVIROMENT: SUSTAINABILITY PERSPECTIVE

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

SIJO SAJI

Department of Computer Science & Engineering
Mar Athanasius College of Engineering
Kothamangalam

SEMANTIC MULTIMEDIA FOG COMPUTING AND IOT ENVIROMENT: SUSTAINABILITY PERSPECTIVE

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

SIJO SAJI

KNOWLEDGE IS POWER

Department of Computer Science & Engineering
Mar Athanasius College of Engineering
Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **Semantic Multimedia Fog Computing and IoT Environment: Sustainability Perspective** *submitted by* **Mr.SIJO SAJI**, *Reg.No.***MAC15CS055** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by him under our supervision and guidance.*

....	....	....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head Of Department***

Date: Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the 'God Almighty' for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

A fog-cloud hybrid architecture that supports a crowd composed of a massive social network and distributed IoT nodes around a smart city environment. The fog computing framework is introduced to support energy efficiency by incorporating IoT nodes. The fog nodes act as an interface to the crowd as they do real-time processing of user requests. They support a constrained amount of data offloading and pass the geo-tagged multimedia data available from the crowd to the big data repository in the cloud. The proposed sustainable framework takes the users' context and provides required service in the form of raw sensory data or in multimedia form when users subscribes to particular fog nodes. Thus avoiding bottlenecks and energy drain at cloud end.

Contents

Acknowledgement **i**

Abstract **ii**

List of Figures **iv**

List of Abbreviations **v**

1 Introduction **1**

2 Existing system **7**

3 Proposed system **10**

3.1 Background and challenges 11

3.2 Applicational example 13

3.3 System design 18

3.4 Adding semantics to the system 21

4 Conclusion **28**

References **29**

List of Figures

Figure No.	Name of Figures	Page No.
3.1	Context-aware framework of the proposed system	12
3.2	The three layers of system architecture: edge, fog and cloud. Measurements collected by wearable devices in the edge layer are processed in the fog layer while cloud layer provide information to caregivers	15
3.3	Acceleration changes in time during a fall.	17
3.4	High-level communication architecture among mobile clients in the edge, fog, and cloud environment.	19
3.5	Raw context data from diversified sources are transformed to a unified 3A model	20
3.6	Gas and air quality data assimilation to define a green zone	21
3.7	Defining different thresholds of of the green zone based on multiple types of raw sensory data	22
3.8	Visualizing semantic zones on the map	23
3.9	Interface where pilgrims can subscribe to any IoT node and sensory stream housed within the IoT node	23
3.10	Live sensory data available from each IoT node with semantic tagging based on the threshold.	24
3.11	Semantic visualization of IoT data to which user can subscribe.	25
3.12	Instance of real-life deployment scenario where IoT data has been used for semantic visualization	26

List of Abbreviations

IoT	Internet of Things
BSN	Body Sensor Networks
BLE	Bluetooth Low Energy
D2D	Device to Device

Introduction

The goal of fogging is to improve efficiency and reduce the amount of data transported to the cloud for processing, analysis and storage. This is often done to improve efficiency, though it may also be used for security and compliance reasons.Popular fog computing applications include smart grid, smart city, smart buildings, vehicle networks and software-defined networks. The metaphor fog comes from the meteorological term for a cloud close to the ground, just as fog concentrates on the edge of the network. While edge devices and sensors are where data is generated and collected, they don't have the compute and storage resources to perform advanced analytics and machine-learning tasks. Though cloud servers have the power to do these, they are often too far away to process the data and respond in a timely manner. In addition, having all endpoints connecting to and sending raw data to the cloud over the internet can have privacy, security and legal implications, especially when dealing with sensitive data subject to regulations in different countries.

In a fog environment, the processing takes place in a data hub on a smart device, or in a smart router or gateway, thus reducing the amount of data sent to the cloud. It is important to note that fog networking complements – not replaces – cloud computing; fogging allows for short-term analytics at the edge, and the cloud performs resource-intensive, longer-term analytics. Many use the terms fog computing and edge computing interchangeably, as both involve bringing intelligence and processing closer to where the data is created. However, the key difference between the two is where the intelligence and compute power is placed.

In a fog environment, intelligence is at the local area network. Data is transmitted from endpoints to a gateway where it is then transmitted to sources for processing and return transmission.In edge computing, intelligence and power of the edge gateway or appliance are in devices such as programmable automation controllers.Proponents of edge computing tout its reduction of points of failure, as each device independently operates and determines which data to store locally and which data to send to the cloud for further analysis. Proponents of fog computing over edge computing say it is more scalable and gives a better big-picture view of the network as multiple data points feed data into it. Because cloud computing is not viable for many internet-of-things applications, fog computing is often used. Its distributed approach addresses the needs of IoT and industrial IoT, as well as the immense amount of data smart sensors and IoT devices generate, which would be costly and time-consuming to send to the cloud for processing and analysis. Fog computing reduces the bandwidth needed and reduces the back-and-forth communication between sensors and the cloud, which can negatively affect IoT performance. Although latency may be annoying when sensors are part of a gaming application, delays in data transmission in many real-world IoT scenarios can be life-threatening – for example, in vehicle-to-vehicle communications systems, smart grid deployments or telemedicine and patient care environments, where milliseconds matter. Fog computing and IoT use cases also include smart rail, manufacturing and utilities.Hardware manufacturers, such as Cisco, Dell and Intel, are working with IoT analytics and machine-learning vendors to create IoT gateways and routers that support fogging .Hajj brings a huge number of pilgrims from all over the world together into a small area for a short period. The massive ad hoc crowd poses many challenges to the city in particular and the country as a whole to provide smart city services. Because millions of people have their individual contexts, researchers in the past have proposed the smartphone as a context manager and as an Internet of Things (IoT) node. This is because the modern smartphone comes with many features, such as a powerful processor, geo-tagged multimedia sharing capability, Internet connectivity through multi-modal means, built-in context providing sensors, and a large amount of local memory, to name a few. Once a smartphone with such sheer context-aware capability is complemented with human intelligence , it becomes a perfect framework to support many smart city applications. People carrying smartphones can participate in many crowdsensing and crowdsourcing applications, which require human intelligence to understand such things, for example, the user and ambience context, significant events, and emergency situations.

Mobile crowdsensing refers to the wide variety of sensing models in which the individuals collectively share data and extract information to measure and map phenomena of common interest . Mobile crowdsensing is emerging as a distributed paradigm, and it lies at the intersection between the IoT and the volunteer/crowd-based scheme. Mobile crowdsensing creates a new way of perceiving the world to greatly extend the service of IoT and explore a new generation of intelligent networks, interconnecting thingsthings, things-people and people-people. Usually, the mobile crowdsensing applications are deployed on contributing nodes, such as mobile, personal devices that can be used to sense the physical environment and provide sensor data to mobile application server. Recently, various kinds of applications have been developed to realize the potential of mobile crowdsensing throughout daily life, such as environmental quality monitoring noise pollution assessment and traffic monitoring . Mobile Crowdsensing requires a large number of participants (individuals) to sense the surrounding environment using the sensing devices (e.g., smartphone) with built-in sensors. It is well-known that in such a large-scale system, the sensing devices continuously generate a huge mounts of data (raw sensor data), which consumes much resource (e.g., bandwidth, energy, etc.). However, the sensing devices have limited resources. Due to the limited resource, the quality of the data collected can be even sacrificed in the scenario of bandwidth constrained networks because of the heavy traffic load . Therefore, the resource limitation imposes a key challenge . For example, images collected in the disaster area take an important role in disaster relief, the images collected may not be able to be uploaded in time due to the limited bandwidth, which can incur huge cost. Another example is that most sensing applications require location information, however GPS, as a widely used spacebased navigation system, has a high power consumption. This can reduce the quality of data (location information) collected. Thus, the resource limitation always hinders the necessary participation and widescale adaption of the targeting applications. Although mobile crowdsensing is a new emerging paradigm, it has been applied in real applications . The application of mobile crowdsensing attracts great attention from both academic and business communities, which started investigating the commercial exploitation of mobile crowdsensing. However, the adoption of mobile crowdsensing approach in business context requires the guarantee of the quality-ofservice (QoS). Hence, QoS is one of the most important arising issues. Therefore, QoS-driven policies are needed to deal with the application non-functional issues to guarantee QoS. In this paper, we review the mobile crowdsensing techniques and challenges. Our focus is to discuss the resource limitation and QoS (e.g., data quality) issues and solutions in mobile crowdsensing. Apparently, a better understanding of resource management and QoS estimation in mobile crowdsensing can help us design a cost-effective crowdsensing system that can reduce the cost by fully utilizing the resource and improve the QoS for users, which manifests the significance of our survey. Our objectives in reviewing the literature are three-fold: 1) to learn what are the problems existing in mobile crowdsensing and how the proposed techniques have helped to develop solutions in the past; 2) to learn the strengths and limitations of different mobile crowdsensing techniques for smartly managing the resource to achieve low cost and good QoS, and how can we use those techniques to better solve similar problems in the future in different paradigms such as the IoT; 3) to provide guidance on the future research directions of mobile crowdsensing for IoT. The remainder of this paper is organized as follows. Section II introduces the concepts of IoT and mobile crowdsensing. describes the strategies of mobile crowdsensing. Describes the challenges of mobile crowdsensing and the future research directions. Section V concludes this paper with remarks on our future work.

It can share geo-tagged multimedia information containing summarized context information with the inquirer or with one's social networks. Hence, an Internet connected smartphone can be considered as an IoT gateway or edge node. Mining this type of information provides great insight to the smart city framework. For example, when humans move from one spatial geo-location to another, they provide a rich set of information to the smart city framework. If a massive number of individuals in a crowd with smartphones participates or volunteers in the building process of such a smart city, a rich set of location-aware semantic services can be provided to the city residents.

However, there remains a challenge of using people's smartphones for smart city crowd sourcing applications as people usually use smartphones for personal usage. For example, due to security and privacy reasons, and power drainage concerns, many crowdsourcing applications cannot assume that the smartphone is used as a fixed, always online IoT node . Nevertheless, thanks to the recent advancements in IoT devices, these devices can complement the mobile smartphones by allowing the creation of a new type of smart city computing paradigm, called fog computing. A smartphone can act as a fog node or edge node by connecting both stationary and mobile IoT nodes, which will allow the collection of even rich user contexts and ambient contexts at the city level.

The above mentioned advances in fog computing along with the fifth generation (5G) network show promising results in solving the issues with the cloud computing paradigm (i.e., the cloud is slow). For example, outsourcing the load of the cloud to the spatially divided fog nodes will allow energy efficiency and sustainability. In scenarios such as Hajj, where millions of people gather for a short duration, real-time communication involving a cloud back-end becomes a bottleneck. This is due to the massive amount of geo-tagged multimedia data, which needs to be sent to and processed by the cloud, and finally shared with the cloud to be visualized by the edge devices. More importantly, due to the mobility of the edge devices, the communication pattern poses even more challenges on the cloud framework. IoT nodes and smartphones, together leveraging the fog computing paradigm, show a promising solution to this massive amount of real-time geo-tagged multimedia data payload processing, context extraction, and providing location-aware context information at the edge while keeping communication with the cloud-based big data architecture intact.

In this article, we propose an IoT-enabled smart city framework to support context-aware services to an ad hoc massive crowd. The framework integrates spatio-temporal multimedia data from IoT stationary and mobile sensory data, and mobile crowdsensing data augmented with human intelligence and social media to support rich types of smart city phenomena. This rich source of data can then be sent to a massive crowd using our proposed smart city model, where people can explore different query results in a semantic visualization environment. The framework intelligently tries to respond to each spatio-temporal user query such as “show me the green zone while I travel from point A to point B” by using the fog computing infrastructure and also updating the back-end cloud architecture using phenomena called distributed transactions. The framework is also able to provide context-aware services to the massive crowd by supporting crowdsourcing. The proposed framework can digest the massive amount of spatio-temporal multimedia data coming from IoT nodes, social media, and smartphones, process the data in real time at both the edge and cloud ends, deduce interesting phenomena, share the results with the massive crowd based on individual queries, and finally, store the query and payload in a big data environment. The proposed architecture is envisioned to support energy efficiency by relieving a cloud from being “always engaged” as much of the processing is performed at the client edge. Moreover, the addition of an incentive model, the deployment of IoT nodes within a smart city landscape, and supporting crowdsensing and crowdsourcing, the proposed framework is envisioned to support sustainability. We have deployed the framework within the city of Madinah, where millions of people gather for a very short period of time. We present details about the implementation and initial deployment scenarios.

The rest of the article is organized as follows. We present the design and modeling of the proposed framework in the next section. Following that, we describe the IoT-based smart city framework deployment experience. We finally conclude the article with the summary and our future directions in the last section.

Existing system

Researchers in the past have proposed the smartphone as a context manager and as an Internet of Things (IoT) node. This is because the modern smartphone comes with many features, such as a powerful processor, geo-tagged multimedia sharing capability, Internet connectivity through multi-modal means, built-in context providing sensors, and a large amount of local memory, to name a few. Once a smartphone with such sheer context-aware capability is complemented with human intelligence, it becomes a perfect framework to support many smart city applications. People carrying smartphones can participate in many crowdsensing and crowdsourcing applications, which require human intelligence to understand such things, for example, the user and ambience context, significant events, and emergency situations. It can share geo-tagged multimedia information containing summarized context information with the inquirer or with one's social networks. Hence, an Internet connected smartphone can be considered as an IoT gateway or edge node. Mining this type of information provides great insight to the smart city framework. For example, when humans move from one spatial geo-location to another, they provide a rich set of information to the smart city framework. If a massive number of individuals in a crowd with smartphones participates or volunteers in the building process of such a smart city, a rich set of location-aware semantic services can be provided to the city residents.

However, there remains a challenge of using people's smartphones for smart city crowd sourcing applications as people usually use smartphones for personal usage. For example, due to security and privacy reasons, and power drainage concerns, many crowdsourcing applications cannot assume that the smartphone is used as a fixed, always online IoT node. Nevertheless, thanks to the recent advancements in IoT devices, these devices can complement the mobile smartphones by allowing the creation of a new type of smart city computing paradigm, called fog computing. A smartphone can act as a fog node or edge node by connecting both stationary and mobile IoT nodes, which will allow the collection of even rich user contexts and ambient contexts at the city level.

With the trend towards ubiquitous computing, context awareness is becoming a key factor in multiple applications. Context-aware computing is a mobile computing paradigm in which applications can discover and take advantage of contextual information (such as user location, time of day, nearby people and devices, and user activity). While several context-aware applications demonstrate the usefulness of this technology, they have not been widely available to everyday users

As the technology surrounding smart phone devices has changed over the past few years, we now find a device containing a collection of sensors. We will show an additional usage for the smart phone: identity management. Today's smartphone not only serves as the key computing and communication mobile device of choice, but it also comes with a rich set of embedded sensors, such as an accelerometer, digital compass, gyroscope, GPS, microphone, and camera. Collectively, these sensors are enabling new applications across a wide variety of domains, such as healthcare , social networks, safety, environmental monitoring, and transportation , and give rise to a new area of research called mobile phone sensing.

By leveraging the sensors and the personal information, smartphones have the capability of automatically communicating our identity to others passively and without any interaction from us. A smartphone based identity management system make a myriad of applications possible that make daily activities easier. We continuously adapt our behavior and actions according to the context and situations. Pervasive computing devices such as smartphones that ubiquitously accompany humans must adapt accordingly.

Smart phones are pervasive devices and studies have shown that adults carry their phones along with them. Furthermore, smart phones are "personal" devices – unlike other devices like personal computers and laptops, phones are seldom shared by more than one individual. In other words, smartphones can be considered less of a "phone" in the traditional sense and more of an identity management device.

Modern smartphones are equipped with a variety of sensors, including GPS, accelerometers, Wifi and Bluetooth among others. Moreover, smart phones already carry personal and sensitive information of the user. By leveraging these sensors and the personal information, smartphones have the capability of automatically communicating our identity to others passively and without any interaction from us.

A smartphone based identity management system make a myriad of applications possible that make daily activities easier. Some illustrative examples include: automatic checking into hotels; automatic entry to office without need for additional identification tag or ID; faster/easier order and payment process at drive through places; automatic notification to emergency services; file sharing and messaging among attendees in a meeting room; auto login to computer and network services like email.

Several applications have started leveraging smartphone features to make day-to-day life easier and more convenient for users. For instance, Google Wallet uses built-in near-field communications (NFC) chip in the smartphones to make secure payments in a store . ATT Alerts is a location-based text messaging alerts service from ATT, and it proactively delivers messages and offers straight to ATT wireless customers' phones

Studies have shown that the current applications are not as secure as they seem to be. For instance, relay attacks to compromise the security of Google Wallet are detailed in . For this attack only an NFC reader device, a notebook computer and some average programming skills were necessary. The report further details that by having the relay app run on multiple devices, a network" of Google Wallets could be created. The attacker can use load balancing techniques to evenly distribute payments among devices and to select a device with a stable network connection for each payment transaction.

The primary reason why the relay attacks and other similar attacks are successful is that applications rely only on one smartphone feature – NFC. The relay device can simply forward the communication to the payment system, thus compromising the application. However, if multiple features of the smartphone are utilized, such attacks can be easily thwarted. For instance, along with NFC capability, localization techniques – both outdoor (GPS) and indoor - can be utilized to verify the proximity of the phone to the payment system. Further, for more secure applications, other capabilities of the smart phone like camera and microphone can be leveraged for undisputable identification of the user.

Proposed system

Fog data mining is an important strategy for IoT in order to reduce the cloud storage requirement, the energy consumption, and package transformation across the wireless network. Each individual sensor or a set of sensors carries out sort of lowpower processing on the acquired data to discover the novelty patterns. Authors in introduced the concept of edge mining for IoT and studied the efficiency of three different edge mining methods on data transmission and energy reduction.

Edge data mining approaches can be categorized into timeseries forecasting, and event-based approaches. In time-series forecasting, two prediction models are working synchronously in the sensor-level and sink (i.e. the base station) nodes where the time-series data is analyzed on the low-level sensors to detect anomalies. The data is only sent to the sink node if the measurement data is different from the predicted one by some constant. Many approaches have been used including stochastic, and regression-based methods. For example, authors in and used Kalman filter and dynamic probabilistic model respectively to reduce data transmission rate from sensor nodes. In authors extend the prior dual model prediction on sensors and sink by transmitting only a state vector estimate instead of the raw data when the error on the sink node exceeds a constant value. Auto-Regressive (AR) model to reduce the processing achieved by sensors and limit the transmitted data to be the coefficients of the model. In authors introduced a simple linear model called derivative-based prediction to predicate the data by computing the gradient of the ending points of a collected data over a short period. Regression-based methods are much simpler than stochastic based methods and easy to implement.

The event-based approaches work similarly to time-series forecasting, but it does not have to replicate all the data generated in the sensor-level on the sink node. The raw data is quantized into some events or categories, and it only pushes the data to the sink node when these events happened. For examples, authors introduced a postural activity monitoring system that recognizes nine different human postures from the accelerometer data using a decision tree algorithm. Similarly in, proposed fall detection system for elderly people based on accelerometers and gyroscopes data. In 100 accelerometers reading is used to detect railway bridge health. In real-time forest fire detection is proposed based on neural networks by analyzing in-networking measurements gathered from multiple sensors to predicate weather index that only sent to the sink node. In authors summarized 105 the measurement data as a histogram to reduce the packet reduction ratio. However, most of these methods are directly classifying the raw sensor data which is sensitive to noise and requiring much more time to process.

3.1 Background and challenges

Hajj attracts a massive crowd, once a year, from almost every country in the world in Makkah and Madinah, the two holy cities, where each individual has to perform some predefined activities at certain geolocations within a specific time duration. A greater percentage of the crowd is composed of males and females of older age who take part in the pilgrimage once in a lifetime. Among the challenges the massive crowd face are difference in languages, needed services, physiological conditions, and technical backgrounds. Although many IoT-based, context-aware smart city solutions have been proposed in the literature, the system has to take into consideration the unique geographic location-based constraint of the venue, and the size and scale-free characteristics of the crowd model. As a result, a key challenge of the framework is to support semantic services that do not require much technical knowledge.

Because the crowd mostly comprises older adults, many of them are unaware of 5G-based Although many IoTbased, context-aware smart city solutions have been proposed in the literature, the system has to take into consideration the unique geographic location-based constraint of the venue, the size and scale-free characteristics of the crowd model. As a result, a key challenge of the framework to support semantic services that do not require much technical knowledge. offline maps, ways of location sharing through maps, and other IoT technologies. For example, the IoT data need to be presented to older pilgrims in such a way that they can visualize and subscribe to the IoT data, and then get semantic notification without many hurdles. Another instance can be the body sensor network (BSN) data of a pilgrim's smartphone, which can be mapped to the health profile of a pilgrim to infer the threshold level of a green route requirement and then share this information with the framework so that the framework can provide a green path avoiding dust and other risk-prone geo-zones. Moreover, the framework can proactively notify the pilgrim semantically if he/she is about to enter a risky zone. Other examples of appealing semantic services can be finding lost pilgrims (as most of them have similar clothing), bookmarking tents (since the tents look similar), locating points of interest via semantic paths, and identifying emergency paths around accidents and congestion zones. (For example, in 2015, a stampede caused the death of more than 700 pilgrims).to quickly set up the ad hoc network of the four types of context sources, including registering the IoT nodes that are available around the city. This network, called the Hajj Social Network, follows the scale-free network pattern, which has to be formed each year from scratch.Because of the growing crowd every year during Hajj, the framework shown in Fig. 3.1 has the network grows in size as the massive crowd starts registering the framework at the onset of their journey and shrinks as the pilgrims leave the country.

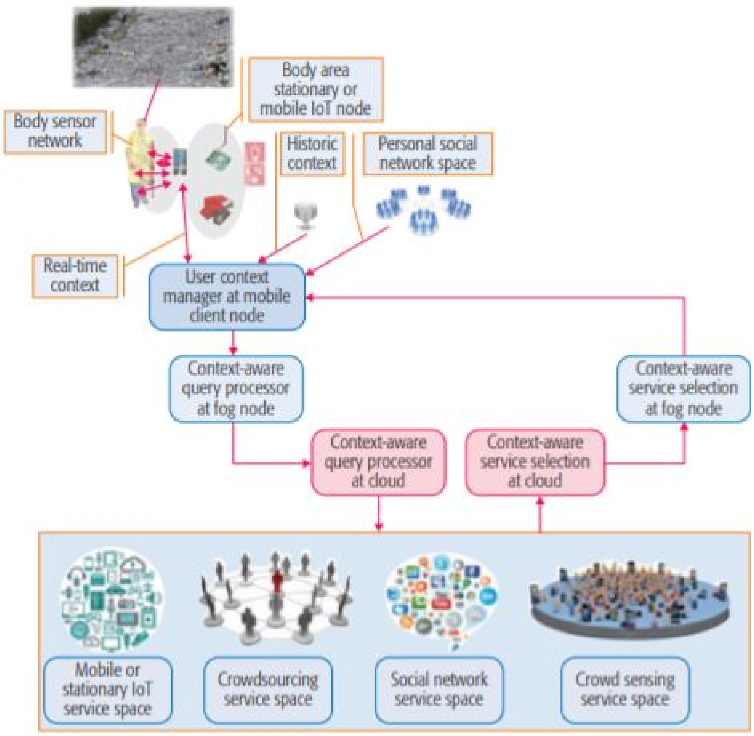


Fig. 3.1: Context-aware framework of the proposed system

3.2 Applicational example

Fall needs to be attentively considered due to its highly frequent occurrence especially with old people - up to one third of 65 and above year-old people around the world are risk of being injured due to falling. Furthermore, fall is a direct or indirect factor causing severe traumas such as brain injuries or bone fractures. However, timely medical attention might help to avoid serious consequences from a fall. A viable solution to solve this is an IoT-based system which takes advantage of wireless sensor networks, wearable devices, Fog and Cloud computing. To deliver sufficient degree of reliability, wearable devices working at the core of a fall detection system, are required to work for prolonged period of time. In this paper we investigate energy consumption of sensor nodes in an IoT-based fall detection system and present a design of a customized sensor node. In addition, we compare the customized sensor node with other sensor nodes, built on general purpose development boards. The results show that sensor nodes based on delicate customized devices are more energy efficient than the others based on general purpose devices while considering identical specification of micro-controller and memory capacity. Furthermore, our customized sensor node with energy efficiency selections can operate continuously up to 35 hours.

Reliability of a fall detection system is multifaceted. We argue that the most important ones include fall detection accuracy, operational time of the system and ability to timely deliver a notification about a fall. Core building blocks of fall detection systems are usually embedded in wearable device accelerometers and gyroscopes, or cameras. In series of works, authors focus on improving the accuracy of fall detection system by combining several building blocks. In Casilari et al. use an accelerometer and a gyroscope, embedded in a smart phone and a smart watch, in authors exploit a depth camera together with an accelerometer-based wearable to increase the accuracy. Other works target quality (sensitivity, specificity, accuracy) of fall detection algorithms, as it sets the scope of their application. Although energy efficiency of a system directly affects duration of its work, and therefore, its reliability, it lacks due attention. A wearable device designed to fulfil the demands for both preserving patient's lifestyle and providing constant monitoring, should be capable of working for prolonged periods autonomously.

This requirement makes a wearable a bottleneck of a whole system when considering duration of work.

General purpose development devices e.g. Arduino Uno, Fio are widely used as the central computational part of a sensor node of fall detection systems. Due to large current draw, these general purpose systems cannot be considered energy efficient. In order to improve energy efficiency, a delicate device can be utilized. In addition, the relationship between sensors' sampling rate and communication data rate is not examined in detail.

An accelerometer which provides three-dimensional acceleration values, together with the wireless transmission mechanism (to the gateway), is another source of energy drain. By a harmonious combination of accelerometer sampling rate and transmission rate, both the fall detection system's requirements and energy efficiency can be achieved. Another variable which influences energy characteristics is communication interfaces those are used in the sensor nodes and their data rates (i.e. SPI, I 2C). To the best of our knowledge, the actual issues which limits energy efficiency when considering overall primary energy consuming source of a sensor node in the fall detection system have not been elaborately investigated. Therefore, in this paper, we design a sensor node for evaluating factors impacting on energy consumption of the sensor node in the fall detection system. These investigated factors include parameters within micro-controller (i.e. type and frequency, communication interfaces), 3D accelerometer sampling rate, Bluetooth technology (i.e. classic, low energy and data rate). In addition, we compare several sensor nodes constructed from general purpose devices with our design in terms of energy efficiency.

The system architecture, shown in Fig 3.2, consists of three main parts: sensor nodes, a gateway with a fog layer unit and a back-end system described as follows:

A sensor node consists of at least three primary components: 3D accelerometer sensor, micro-controller and wireless communication module. Data is gathered from the 3D accelerometer via a communication interface such as UART, SPI or I 2C. Depending on particular fall detection systems, the collected data can be pre-processed or kept intact before Fig. 3.2.

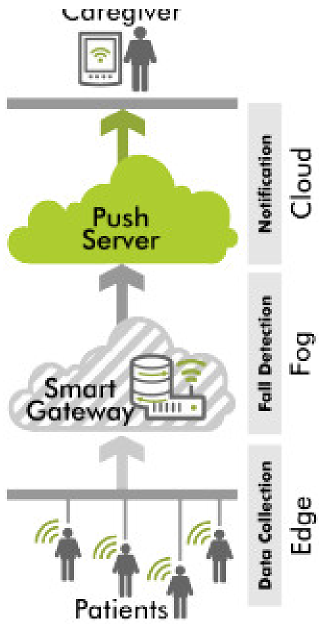


Fig. 3.2: The three layers of system architecture: edge, fog and cloud. Measurements collected by wearable devices in the edge layer are processed in the fog layer while cloud layer provide information to caregivers

The three layers of system architecture: edge, fog and cloud. Measurements collected by wearable devices in the edge layer are processed in the fog layer while cloud layer provide information to caregivers.being transmitted to a gateway wireless. Frequently, collected raw data without pre-processing (i.e. wavelet transformation, or neural filtering) is transmitted as such. Complex processing mechanisms, like fall detection algorithms based on hidden Markov model, are carried out at the gateway because of significant computational requirements. In our fall detection system, Bluetooth is utilized for wireless communication between sensor nodes and a gateway.

Alongside with primary features of receiving data from sensor nodes and transmitting the data to Cloud servers, a gateway with fog computing in a IoT-based fall detection system can offer advanced services such as data processing (i.e. complex filtering mechanisms, or data fusion), data compression, security, push notification service, local storage, fall detection algorithms and decision making. Depending on particular fall-detection applications, specific services might be proffered. In our fall detection system, a gateway equipped with a Bluetooth module and an Ethernet module is used for receiving raw data from sensor nodes and transmitting the data to Cloud, respectively. In addition, the gateway provides a push notification service, local storage and a fall detection mechanism. The collected data is processed with a fall detection algorithm. When a fall is detected, the gateway triggers the push notification service for notifying caregivers in real-time. Local storage is used for storing both user data for some periods of time and service data i.e. algorithms.

The back-end system includes cloud servers and an user terminal. When the Cloud receives a signal from the gateway's push notification service, it notifies appropriate party via realtime messages. An end-user (i.e. caregiver) can then view these messages with an Internet browser or a mobile application. In our system, the system checks reply messages from endusers after sending notification messages in order to verify that fall notification messages are properly received. The endusers can reply with an affirmative messages via a browser or an appropriate application to acknowledge the fall case.

The gateway is implemented on Rasberry Pi v3 as it is equipped with a compelling 4-core 1.2 GHz CPU, 1 GB RAM and extensible storage, which can guarantee that complex algorithms (i.e. push notification and fall-detection) run smoothly with low latencies. Additionally, Bluetooth classic, Bluetooth low energy, and Ethernet are supported. Therefore, functionality of the gateway can be conveniently implemented without needing supplementary hardware components. Ubuntu is used for managing the gateway due to its benefits such as customization, security support, users support and diversified administration tools. With Ubuntu, all tasks can be run fairly and possible hardware race condition can be avoided. Because of its performance, ease of use, scalability and secure nature, the MySQL together with local storage are used for storing 3D acceleration data, user records, and essential data used by the services.

The algorithm first removes noise with a digital secondorder Butterworth filter for posture detection and dynamic analysis. The output from the filter is used for calculating fall-feature parameters which are then checked against the first simple threshold. An example of the SVM threshold comparison method is shown in Fig. 3.3.

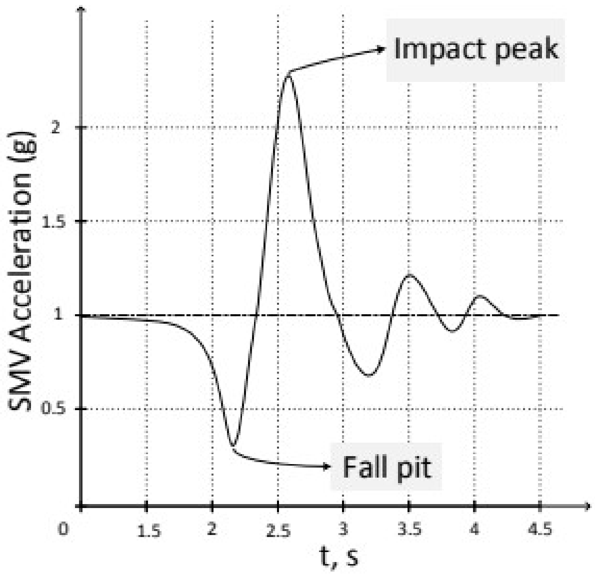


Fig. 3.3: Acceleration changes in time during a fall.

The threshold is decided when the specificity is the highest with a sensitivity of 100threshold, it indicates that a possible fall has occurred. The output of possible falling cases is applied to the second threshold. Finally, based on the output of verifying these parameters with the second threshold, a decision about the fall is drawn. A summary of processing 3D accelerometer data with the fall detection algorithm is shown in Fig. 3. When an actual fall is detected, the gateway immediately sends a push notification to the caregiver. The push-notification is primarily implemented in the gateway with an assistance of Pushbullet. Pushbullet provides full API for creating push notification applications which allows sending text, files, instant messages in real-time between different types of devices (server, gateway, mobile phone, computer).

The back-end of the system includes cloud service and a terminal application. End-users such as family members, doctors or caregivers can use any Internet browser or a push notification enabled smart phone application to subscribe to the push notification server. In addition to receiving push notification messages, end-users can confirm that a fall case is noticed via replying messages by using the ”confirmation” button in an application or a browser’s interface.

For implementing a sensor node, several hardware components including ADXL345 three-dimensional accelerometer sensor, micro-controller and Bluetooth module are used.

ADXL345 is a high resolution and low power consumption digital three-dimensional accelerometer sensor. It provides 16bit two complement output, which incorporates 3 dimensional acceleration along x, y, z axis. It consumes only 90 µA for providing output values at a rate of 400 Hz. It is equipped with SPI and I 2C interfaces for communicating with a microcontroller. In order to achieve a fair and diversified comparison, an ATMega328P and ATMega32U running at 16 MHz are primarily used in our evaluation. Accordingly, Arduino Uno board, Arduino Micro, a customized device are evaluated. The customized device is constructed from ATmega328P 16 Mhz micro-controller, 16kHz crystal oscillator and set of capacitors and resistors. Although the device can operate at both 3.3V and 5V, 3.3V is used for our experiments.

3.3 System design

The proposed framework aims to provide context-aware smart city services through the following: collecting IoT, crowdsourced, and social media data; storing the incoming multimedia data initially at the distributed fog nodes and finally to the big data repository; inter-correlating context-aware clustering of crowds; and then rendering back spatio-temporal services to each individual based on his/her context.As shown in Fig. 3.4, different smart to the massive crowd or a set of IoT nodes through the proposed platform, which leverages both fog and cloud computing architecture . The framework generates visualization results for each requester in a personalized fashion. Figure 3.1 shows details about different components within the framework,city applications allow a query or task to be sent by a pilgrim or city resident

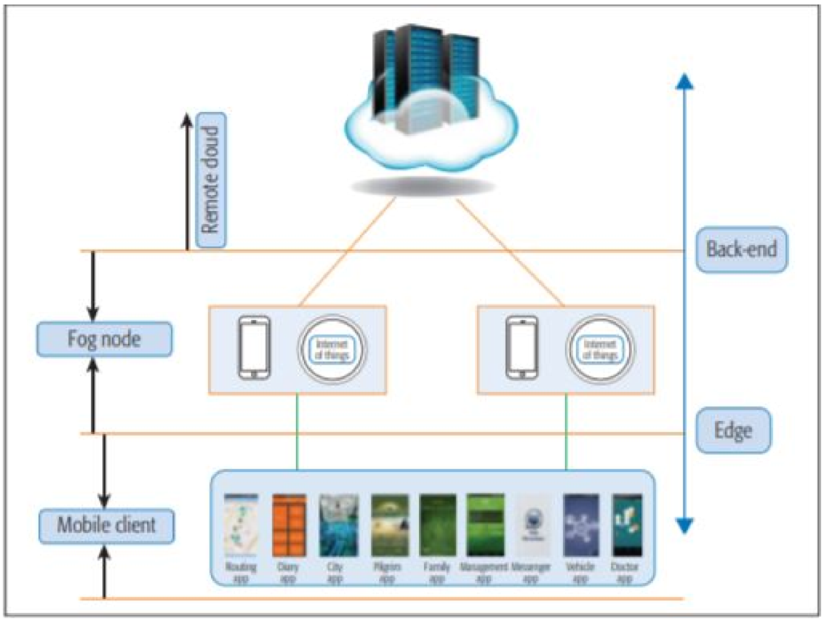


Fig. 3.4: High-level communication architecture among mobile clients in the edge, fog, and cloud environment.

We assume that on behalf of a smart city environment, numerous stationary IoT devices have been deployed around the city for collecting different smart city events. In addition, we assume each person within a massive crowd has a smartphone and is optionally surrounded by a BSN and may carry a subset of IoT devices, whether stationary or mobile, forming a body IoT network. We further assume that the smartphone has 4G/5G Internet connectivity through which the user can be connected to his/her personal social networks. The built-in smartphone sensors, the sensors within the BSN, and the IoT sensors together allow collecting real-time user and ambient contexts.

As shown in Fig. 3.4, smartphones and stationary IoT nodes can act as fog nodes that can store the historical context of a person within a pre-defined geo-location. As shown in Fig. 3.5 real-time and historical contexts are combined using the 3A model and unified to a the user context manager in the user's smartphone generic context by. user. As shown in Fig. 3.4, the framework provides different smart city applications through which a user can request a variety of services, such as Q1 = "Show me a dust-free path from point A to point B at this moment."

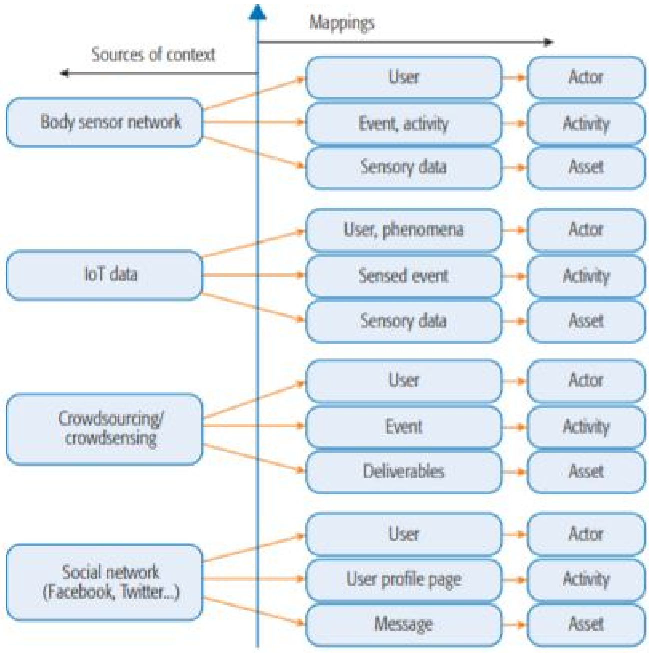


Fig. 3.5: Raw context data from diversified sources are transformed to a unified 3A model

The user query available on the smartphone application is assimilated with the user context available from diversified sources in the context-aware query processor at the fog node. Finally, the fog node sends a query to its peer component, that is, the context-aware query processor at the cloud, for further analysis.

The cloud framework consists of four different types of platforms: the mobile or stationary IoT service, crowdsensing service, social network service, and crowdsourcing service. A user query such as Q1 might be responded to by one, a subset, or all of the four sources. For example, a dust-free path can be found by obtaining real-time data from the IoT dust sensors deployed around a city, from the query response available via crowdsourcing, where dust levels can be inferred by human sensors or mobile IoT sensors. The dust-free path query can be shared on a social network, such as Facebook or Twitter, and the results are available via social network services. Finally, congestion and other city weather-related information can also be available from mobile crowdsensing services. Assimilating all available sources of information can lead to a rich set of responses for the requester. It has to be noted that the sustainability of the framework depends on motivation through proper incentive models and it should be energy-efficient. In our proposed sustainability model, every stakeholder spends some money and hence expects some reward, the details of which are beyond the scope of this work. For example, a user query such as "Show me the rain overlay of the city road network so that a water-clogged path can be avoided" might result in a huge amount of multimedia data, which needs to be processed by the cloud and fog nodes instantly.

3.4 Adding semantics to the system

To add semantics to the system, we follow the 3A model shown in Fig. 3.5, where we assimilate different units of contextual data, combine them, and use semantic rules to define higher-level contexts. For example, raw sensory data available from IoT sensors are grouped together to deduce high-level semantic overlays. Figure 3.6 shows a scenario where two sensory data streams are grouped together and checked against some rule-based thresholds that determine the type of semantic zone. Figure 3.7 shows multiple the semantic definition scenarios that require defining a green zone based on multiple types of sensory data and their thresholds.

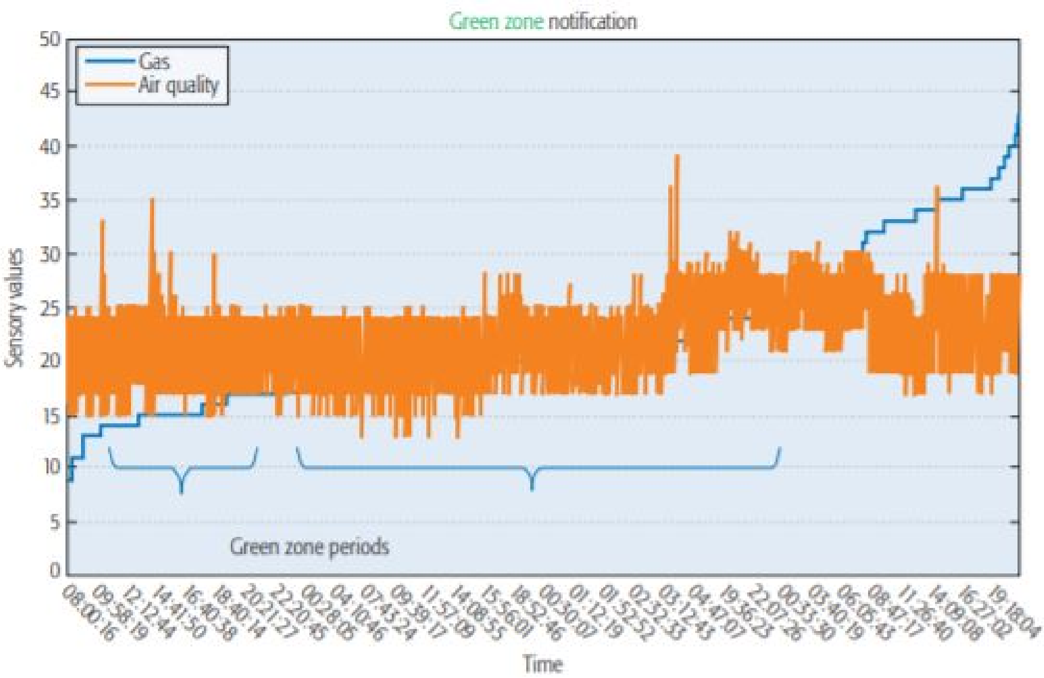


Fig. 3.6: Gas and air quality data assimilation to define a green zone

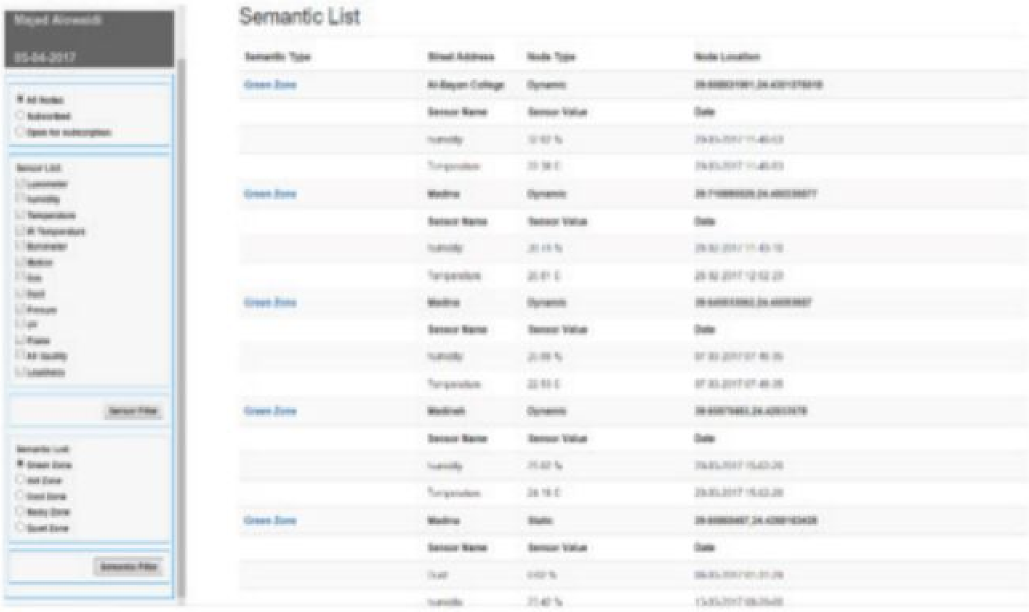


Fig. 3.7: Defining different thresholds of of the green zone based on multiple types of raw sensory data

Definition of the green zone can be customized based on standards or one's comfort level. It shows some sample scenarios of semantic values added as an overlay. For proof of concept, we used more than 10 sensors housed within each mobile or stationary node and used them to define overlays, such as green zone, crowded zone, dusty zone, windy zone, extremely hot zone, high UV zone, and water-clogged zone. Figure 3.8 shows an instance of four types of overlays represented by four IoT nodes. Once expanded, the node shows multiple types of metadata, such as the location of the node, instantaneous value of sensory data, and semantic threshold of each type of sensory data.

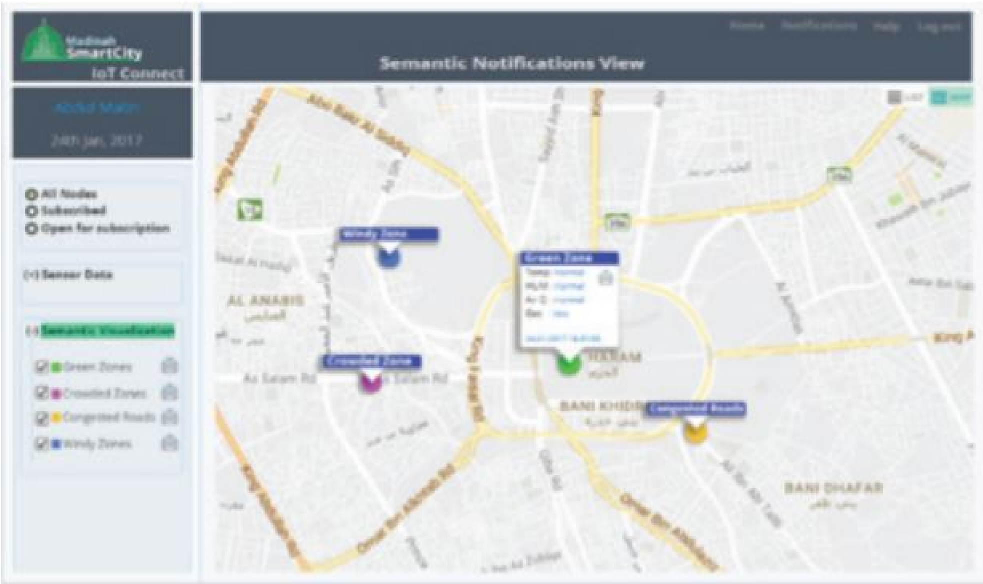
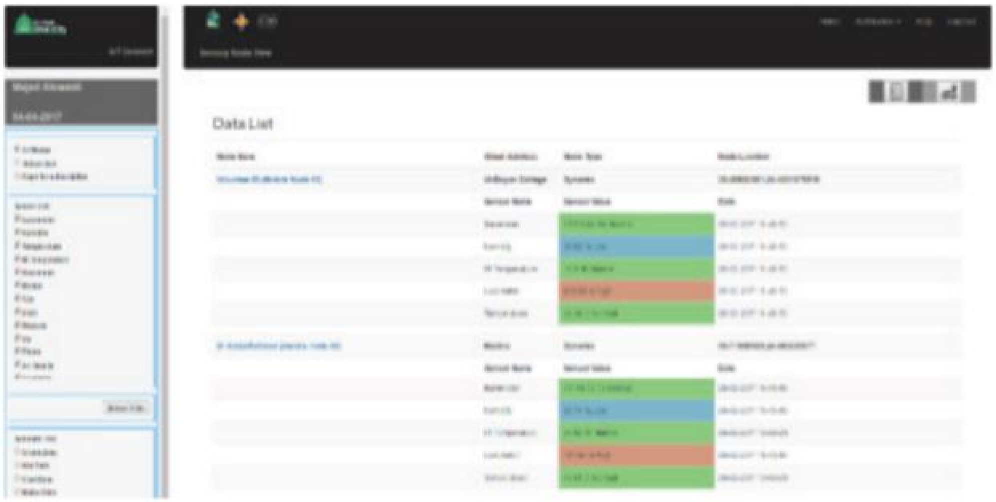


Fig. 3.8: Visualizing semantic zones on the map

Fig. 3.9: Interface where pilgrims can subscribe to any IoT node and sensory stream housed within the IoT node

Figure 3.9 shows the process by which a user can subscribe to either the raw sensory data belonging to a particular IoT node or a semantic value. For example, a user's subscription can be: S1: “Alert me whenever any road on my way to the office has black ice.” S2: “Alert me whenever the dust level of routes along which I usually walk crosses the threshold defined by my doctor.” S3: “Alert me of green zones while calculating the route between point A and point B.”

Figure 3.10shows the semantic tags associated with the sensory data to which one has subscribed in the dashboard.The framework deduces each user’s context based on IoT sensory data to which the user has subscribed, personal social network domain, crowd sourcing and crowdsensing data, and BSN data.



(c)

Fig. 3.10: Live sensory data available from each IoT node with semantic tagging based on the threshold.

All data types are finally assimilated by the user-context manager running in the user’s smartphone. These data are also made available to the fog node via Bluetooth Low Energy (BLE), WiFi Direct, or 5G device-to-device (D2D) spectrum. As soon as the user context is changed, the framework detects the new context, and the services to which the user has subscribed in this context are offered. Moreover, the system proposes a subset of the user’s personal social network based on the given context.Figure 3.11 shows the semantic visualization of IoT data to which user can subscribe.

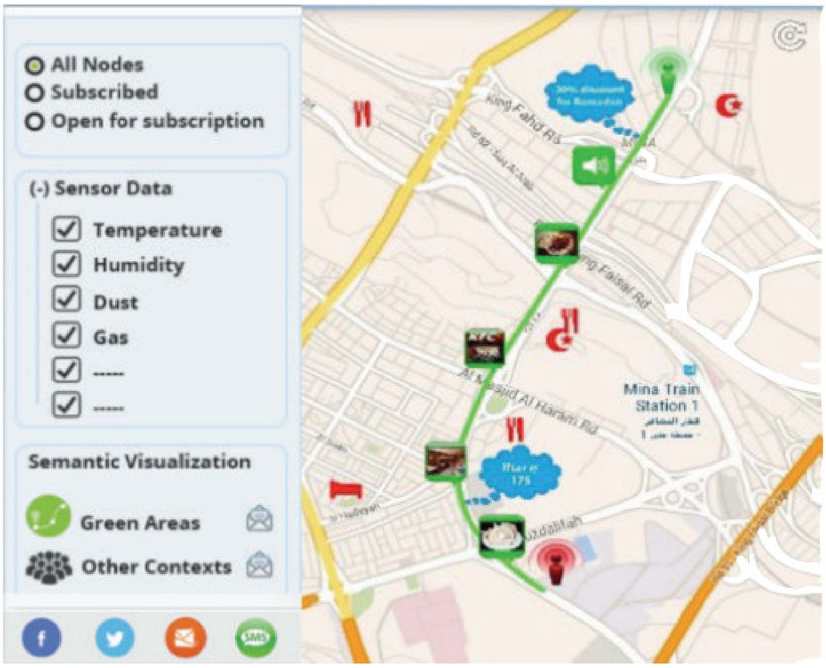


Fig. 3.11: Semantic visualization of IoT data to which user can subscribe.

The proposed framework is found to be energy-efficient. The distribution of fog nodes is done in such a way that the framework does not use much energy in providing services to the stakeholders. Two experiments were performed to measure the actual energy consumption (in kilowatt-hours) by the framework. In the first experiment, energy consumption per hour in a randomly chosen day was measured. In the second experiment, energy consumption per day over two consecutive weeks was measured.

for these two experiments. The predicted energy consumption is also shown in the plots. From the left plot, we see that even in the peak hours the framework consumed less than 0.5 kWh energy. From the right plot (where day 1 represents Sunday), we find that the maximum energy consumed by the framework was less than 50 kWh. During weekends, the energy consumption fell below 30 kWh. From these experiments, we can conclude that the proposed framework is an energy-efficient one. In our future experiments, we would like to measure the energy consumption during the Hajj period.

We have developed several smartphone applications to collect contextual information from BSNs, the IoT, social networks, and crowdsourcing applications. We have also developed a smartphone application to act as a fog node that was deployed during the testing of the system. For this purpose, we used both mobile and stationary IoT nodes. These IoT nodes were tested under different scenarios, such as within a smart home environment, driving in a city, indoors in a hospital, the external environment at high temperature, wind, rain, dust, lighting, and UV indices, in green environments, and during the day and at night, to name a few.

The server was developed using Amazon Cloud Services, details of which can be found in. We deployed numerous stationary IoT nodes around the city of Madinah, Saudi Arabia. (Thanks to Al Bayan Foundation for the Scientific Research Unit of the University of Prince Muqrin, Madinah. Thanks to KACST, Saudi Arabia, and for the support from Majed Alowaidi and Dr. Abdulmotaleb El Saddik of the University of Ottawa and Delwar Hossain and Syed Abdullah of the Next Generation Media Lab.) Moreover, we have distributed mobile IoT nodes among many of our volunteers, who regularly use IoT nodes and help us rigorously test different semantic aspects of the framework.

Fig. 3.12: Instance of real-life deployment scenario where IoT data has been used for semantic visualization

Figure 3.12 shows a mobile edge/fog node that can cache the IoT data and deduce semantic annotations per sensory stream. The fog node can be configured with WiFi or 3G communication protocols to share the semantic notifications with the server-side cloud architecture, show volunteers who use mobile IoT nodes while capturing the phenomena of interest. Figure 6e shows a stationary IoT node deployed in a residential area, while shows a semantic routing based on live IoT data. The framework is aimed to support the smart city initiative of the city of Madinah, which hosts more than 10 million pilgrims and visitors a year.

Conclusion

In this article, we have proposed a semantic fog computing framework that can help achieve energy efficiency and sustainability by distributing the cloud load to spatially distributed fog nodes and providing incentives to different stakeholders of the framework, respectively. The framework is aimed at providing semantics to the IoT, body sensor data, and crowdsourcing and crowdsensing data. We envision testing the framework on a large scale during Hajj 2018, the annual Muslim pilgrimage to Mecca, where a massive number of pilgrims can leverage the IoT and fog computing paradigms. In future work, we also plan to add a blockchain-based security layer to the system.

References

[1] M. A. Rahman, E. Hassanain, and M. S. Hossain. "Towards a Secure Mobile Edge Computing Framework for Hajj," IEEE Access, vol. 5, no. 2017, May 2017, pp. 5:11,768–81.

[2] M. A Rahman, and M. S. Hossain, "Location-Based Mobile Crowdsensing Framework Supporting a Massive Ad Hoc Social Network Environment," IEEE Commun. Mag., vol. 55, no. 3, Mar. 2017.

[3] Liu, H. Shen, and X. Zhang, "A Survey of Mobile Crowdsensing Techniques: A Critical Component for the Internet of Things," Proc. ICCCN '16, Waikoloa, HI, Aug. 1–4, 2016..

[4] H. Yu et al., "Quality and Budget Aware Task Allocation for Spatial Crowdsourcing," Proc. AAMAS '15, pp. 1689–90, Istanbul, Turkey, May 4–8, 2015

[5] T. N. Gia et al., "IoT-Based Fall Detection System with Energy Efficient Sensor Nodes," Proc. IEEE NORCAS '16, Copenhagen, Denmark, Nov. 1–2, 2016.

[6] P. Hu et al., "Fog Computing Based Face Identification and Resolution Scheme in Internet of Things" IEEE Trans. Ind. Informatics, vol. 13, no. 4, 2017, pp. 1910–20.

[7] H. M. Raafat et al., "Fog Intelligence for Real-time IoT Sensor Data Analytics," IEEE Access, vol. 5, no. 2017, pp. 24,062–69.

[8] B. S. Negash et al., "Leveraging Fog Computing for Healthcare IoT," A. Rahmani et al., Eds., Fog Computing in the Internet of Things, Springer, 2018, pp. 145–69.

[9] J. Baliga et al., "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," Proc. IEEE, vol. 99, no. 1, Jan. 2011, pp. 149–67.