

2016

Smart Fog Computing Interface for Healthcare Domain Applications

Admir Monteiro
University of Rhode Island, admir_monteiro@my.uri.edu

Follow this and additional works at: <https://digitalcommons.uri.edu/theses>

Terms of Use

All rights reserved under copyright.

Recommended Citation

Monteiro, Admir, "Smart Fog Computing Interface for Healthcare Domain Applications" (2016). *Open Access Master's Theses*. Paper 1046.
<https://digitalcommons.uri.edu/theses/1046>

This Thesis is brought to you by the University of Rhode Island. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu. For permission to reuse copyrighted content, contact the author directly.

SMART FOG COMPUTING INTERFACE FOR
HEALTHCARE DOMAIN APPLICATIONS

BY

ADMIR MONTEIRO

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2016

MASTER OF SCIENCE THESIS

OF

ADMIR MONTEIRO

APPROVED:

Thesis Committee:

Major Professor Kunal Mankodiya

Leslie Mahler

Qing (Ken) Yang

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2016

ABSTRACT

Technology is advancing tremendously in this day and age where more and more devices are being deployed in the market of digital devices; so much so that the market is being oversaturated with many different kinds of digital devices to simplify our everyday needs in life. This advancement can have negative consequences, however, there may also be benefits that stem from this technological advancement- especially in healthcare.

One possible benefit is to utilize the device's capabilities to give real time feedback to people with medical diagnoses. Furthermore, that device could be connected to a stream of other devices to have a greater impact towards diagnosing and treating patients' health concerns. This platform is known as the Internet of Things (I.o.T.). The I.o.T. is a network of wireless physical devices that are capable of communicating with one via embedded systems, microcontroller devices coupled with sensors, or any other device that is capable of acquiring data.

In this thesis, we investigated the speech of people with Parkinson's disease (PWPD) by applying the I.o.T. computing interface architecture to provide a better solution for diagnosis involving healthcare providers, patients, and administrators. First, we utilized a localized or mobile approach to acquire audio data from a patient with Parkinson's disease. We investigated the different capabilities of wireless connected devices such as a micro-controller to collect audio data from PWPD, transfer the data securely over the network, process the data for specific means of data extraction and diagnosis, save sensitive audio data extraction, and provided a structural means of data interaction for the user of the processed data to be visualized in a Graphical User

Interface (GUI) setting. Our novel means of data collection, transfer, and processing which deploys a micro-controller allows for a localized network usage that minimizes security risks and improves the mobile device's processing which in turn saves on power consumption, and allows for quick extracted information to be saved on the cloud that decreases the risk for privacy issues. Our method also allows for doctor and patient interactions where a doctor has the ability to evaluate the user's (who also happens to be the patient) progress without having to physically meet the user. We expect that our methodology and implementation of the I.o.T. architecture will improve treatment of PWPd.

ACKNOWLEDGMENTS

First of all, I would like to thank my Lord and Savior, Jesus Christ, without whom none of this work would have been possible. I am so grateful to God for giving me the strength and courage to be able to handle all of the work and challenges throughout the work for this thesis. Through all the stressful times and scares, the Lord saw me through, and for that I am grateful for his grace and mercy.

Second, I would like to thank my family and my professors who helped me along the way. Many thanks to my main professors, Dr. Kunal Mankodiya, Dr. Leslie Mahler, and Dr. Ken Qing Yang who were able to bear with me and offer criticism, advice, and guidance along the way so that I could be finalize the master's thesis. My family, my mother in particular, I am so thankful for. Though some days were tearful, you knew best how to encourage me, and for that I am blessed.

Furthermore, I would like to thank all of my friends who have been my side since the beginning of all this work.

Last but not least, I would like to thank other students and colleagues who have been involved in this project either since the beginning or being able to come onto the team while the project was taking place. I would like to thank all of the students who throughout the study was able to provide me with refreshing insight.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER 1	1
INTRODUCTION.....	1
1.1 BACKGROUND.....	1
1.2 MOTIVATION.....	3
CHAPTER 2	7
REVIEW OF THE LITERATURE	7
2.1 I.o.T. INTELLEGENGE	7
2.2 FOG COMPUTING.....	9
2.3 I.o.T. ARCHITECTURE.....	11
2.4 HEALTHCARE AND I.o.T.....	14
CHAPTER 3	17
METHODOLOGY	17
3.1 HARDWARE LAYER	17
3.2 I.o.T. MIDDLEWARE	20
3.2.1 COMMUNICATION/ACCESS LAYER	20
3.2.2 SOFTWARE/DATA INTERFACE.....	22
3.3 APPLICATION LAYER.....	24
CHAPTER 4	26
DESIGN AND IMPLEMENTATION / FINDINGS	26
4.1 OVERALL ARCHITECTURE DESIGN.....	26
4.2 HARDWARE (EDGE)DESIGN AND IMPLEMENTATION	28
4.3 DATABASE APPROACH/IMPLEMENTATION OF BACKEND	32
4.4 APPLICATION DESIGN/IMPLEMENTATION OF FRONT END	36

CHAPTER 5	39
CONCLUSION	39
5.1 DISCUSSION AND RESULTS	39
5.2 CONTRIBUTIONS	43
5.3 FUTURE RESEARCH	44
5.4 TOOLS USED.....	45
BIBLIOGRAPHY	46

LIST OF TABLES

TABLE	PAGE
Table 1. Explains the ability of Fog Computing individually and as nodes in relation to the Network and the Cloud	10
Table 2. Devices and protocols of the I.o.T. platform/Layers	13
Table 3. Power consumption of the Intel Edison during various operations	19
Table 4. Selected speech tasks for the Fog Computer	31
Table 5. Description of some of the Android application designs	38
Table 6. Benchmarking the fog computer for computing the clinical speech features.	42
Table 7. Tools used for Implementation of the Smart Computing Interface	45

LIST OF FIGURES

FIGURE	PAGE
Figure 1 Predicted market share of I.o.T. applications by 2015	8
Figure 2. I.o.T.Platform.....	12
Figure 3. The Internet of Things applications ranking.....	16
Figure 4. Intel Edison Fog Computing Board.....	19
Figure 5. View of the Communication layer implementation.....	21
Figure 6. Flowchart of the User authentication and privileges for specific Accounts..	25
Figure 7. Overall Structure of the I.o.T Platform overview.....	26
Figure 8. Database Schema.....	35
Figure 9. Android application designs	36
Figure 10. Android application designs	37
Figure 11. Pathological loudness computations.....	39
Figure 12. Samples of average loudness calculations.....	39
Figure 13. Clinical features computed on the Intel Edison.....	40
Figure 14. Fog computations of three clinical features for speech files	41

CHAPTER 1

INTRODUCTION

The first section (1.1) of the introduction includes the background for the study. Section 1.2 describes the motivation for the study. Lastly, section 1.3 explains the research objectives and some of the inquiries raised throughout the study.

1.1 BACKGROUND

As the internet continues to grow, more and more devices connect and are trafficking on many networks. The need to minimize traffic on these networks with physical devices and sensors locally is crucial to the infrastructure of the internet. There are numerous applications that these devices can be used for since so many sensors can be combined with physical devices and microcontrollers. I.o.T. applications include electronic-health, energy management for city grids, HVAC (Heating, Ventilation, and Air Conditioning) systems for smart homes, wearable sensing technology, items for the retail industry, among others. I.o.T. connected devices create a gateway and platform that addresses the complex challenges that arise in the above mentioned applications to facilitate data acquisition, processing, as well as storage in safe, private, and secure manner. By utilizing the internet of things and its components such as the embedded computer systems, also known as the Fog Computer, dealing with everyday challenges can be simplified because of a less complex and layered architecture interface that can be used to facilitate a domain for healthcare applications.

The Internet-of-Things is defined as a framework that can interconnect sensors, actuators, and the cloud, communicating via the internet or other wireless networking capabilities such that end-users can benefit from connected intelligence [1]. For

example, Cisco coined the term "Fog Computing" [2] to describe a new category of devices and services that can provide computing intelligence at the edge of the network. Fog computing is defined as a distributed computing paradigm that fundamentally functions as a middleman between the sensors and/or actuators and the cloud [3]. Fog computing generates three tier architecture (clients \leftrightarrow fog \leftrightarrow cloud). The key benefits of adopting fog computing over traditional cloud computing include; reduction in network traffic [4] and data storage on the cloud [5]; low response latency in I.o.T. applications demanding real-time actuations such as smart traffic lights [6].

Fog computing is increasingly penetrating the area of healthcare, especially to improve tele-health and telemedicine infrastructure that promise to cope with the rising healthcare needs in elderly population and individuals with chronic conditions around the world. For example, fog computing upgrades the standards of body sensor networks for medical signal processing, energy-efficient computing, and privacy and security [7]. Fog computing could help healthy individuals keep track of their wellbeing through various wearable gadgets and smartphone apps when merged with I.o.T. [8].

Another part of the interface between the embedded computing and the cloud is the middleware. Middleware is an interface where another layer of the network has the ability for data transfer and storage. Middleware acts as interface for the communication of the connected devices occur. Middleware essentially provides the Application Programming Interface (A.P.I.) for the layer where the embedded computing communicates with the application information needed for integration and details which are handled in the background [9].

1.2 MOTIVATION

A system based on I.o.T. provides an ability to diverse populations and diverse fields to associate a need created challenges and environments that arise in our society with a technology solution. The development of the I.o.T. requires the use of heterogeneous computing where an API and interconnection of different devices and processing are required. Our motivation for utilizing this system arose where a challenge in the healthcare domain was met. We were motivated to construct a simplified but robust layered system where a healthcare challenge that requires data collection and diagnostics could be utilized and supported in the structure of our study and research in this thesis.

The healthcare challenge that was addressed in this study was a solution for speech deficits in PWP. In 2010, the National Institute of Deafness and Other Communication Disorders (NIDCD) reported that 7.5 million individuals in the USA alone have speech disorders [10]. Parkinson's disease (PD) is also categorized as one of such communicative disorders that not only affects motor activities but also impairs the speech and voice functions in over 90% of patients [11]. Patients with PD experience a combination of speech impairments in various ways including: reduced vocal loudness; a breathy or harsh voice quality; imprecise consonants and distorted vowels; and reduced voice pitch (fundamental frequency) variation. Hence, patients with PD go to speech language pathologists (SLPs), who diagnose their speech impairments and then give them personalized speech treatments targeted to restore or improve the impaired speech functions. The challenge to treatment occurs when the patients are given speech exercises as homework on their mobile devices by the SLP. PWP have a progressive

neurological disorder that involves motor and non-motor symptoms that can interfere with the successful completion of home speech exercises that are vital to the speech treatment regime. Finally, we seek tele-health technologies to monitor the progression of their patients remotely.

1.3 OBJECTIVE (PROBLEM STATEMENT) AND INQUIRIES

We utilized the architecture of the I.o.T. system, including Fog Computing, which provides an interface where multiple devices are used for interaction between numerous users involved. This technology allowed for a configurable platform where different users of our system could interact with one another in multiple instances. The work included created a mobile application on the Android platform so the administrator, doctor, and patient could interact with one another to accomplish better diagnosis of PWPD. We called our work, F.I.T: A Fog-driven Internet of things interface for smart watch-based speech treatments.

F.I.T started with the goal of analyzing recorded speech from patients that we gathered on the fog computer. We wanted to implement a fog interface to process and analyze information needed from the audio files quickly. Then we advanced our work further so we could include the interconnection of physical devices, software, and people who could be helped because of it.

The goal of the study was *to develop an I.o.T based system where a gateway (Fog, Edge) computer could participate in layered systems to interface with the cloud and different kind users to provide a task specific solution.* Furthermore, we wanted to develop a system that was user intuitive for most of the process after the initial setup.

To complete the aforementioned goal, the following research questions were addressed:

1. What was the main objective of the Fog computer in the I.o.T. based system?

2. What other connected devices, components, and software were going to take part in the research?
3. How would we be able to design software in an administrative structure to assist our users?
4. How were we going to design our system so that it is configurable?
5. How were we going to design our system so the ease of use could be as simple so that setup was as minimal as possible?

CHAPTER 2

REVIEW OF THE LITERATURE

The structure of this chapter will define I.o.T. and its application to the healthcare domain. In the first section we will explain the I.o.T with some advantages of it and its architecture design setup. The following sections will seek to explain the required modules to design and implement an I.o.T system.

2.1 I.o.T. INTELLEGENCE

According to Gartner Research, world's leading information technology research and advisory company, I.o.T is defined as '*the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment*'. These states and environments require a certain intelligence to take place during interaction and communication of the physical objects.

The intelligence of I.o.T. is its ability to use a variety of physical devices in an intelligent fashion to solve and simplify a societal problem. This intelligence has been taken notice by many educational and professional institutions. I.o.T is offering a great market for many. It is now being predicted that the physical devices that will make I.o.T based systems will estimate about 212 billion globally by the end of 2020 [12]. Furthermore, according to McKinsey Global Institute, the number of the devices that communicate with another (M2M: machine to machine) has increased by 300% in the last five years [13]. Furthermore, I.o.T. is beginning to influence the market economically at such a rapid pace in many different fields. In figure 1 we see an example of this. [14] [13].

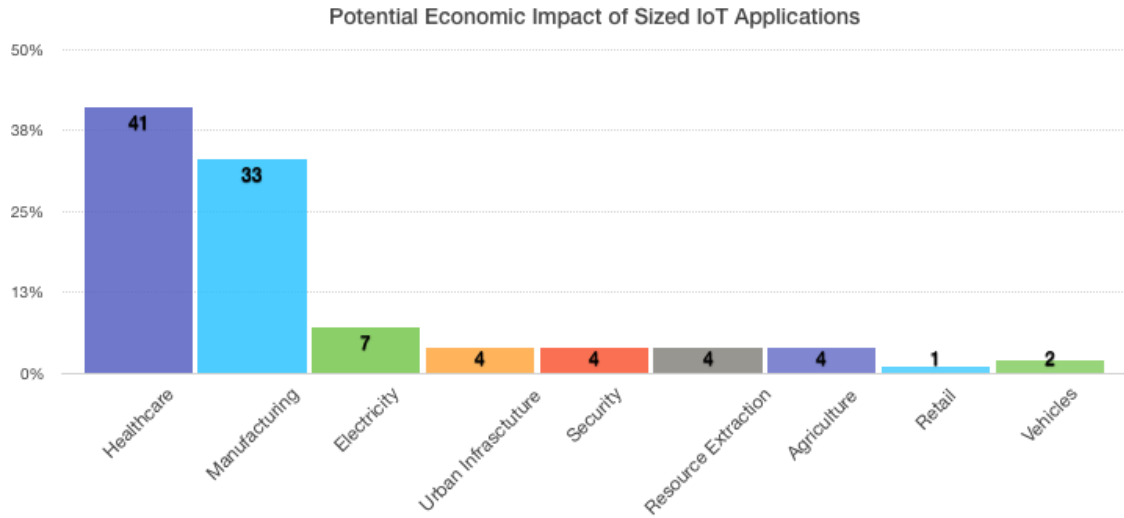


Figure 1. Predicted market share of I.o.T. applications by 2015

2.2 FOG COMPUTING

According to Cisco, Fog computing are physical devices which are nodes employed anywhere with the connectivity to the network and has the following operations: computing, storage, and network connectivity [15]. Because of these attributes, there are multiple reasons that fog computing is becoming a factor in today's market. Following are some are the reasons why the Fog is beneficial and below is a table illustrating how different setups of fog computing works and how it is interacted with the cloud [15]:

Cost:

Large networks are constantly being buffered with tons of data that are circulating, bottlenecks are more susceptible to occur as those same resources are being utilized. In the case of Fog Computing, bottlenecks are less frequent to take place since there is less amounts of data that is being moved throughout the network since it is centralized. This can be less cost in terms of bandwidth transmission in the network [16].

Network Expedience:

The ability to process and analyze data closer to its originating source can significantly minimize network latency [15]. This expedited computation of the data to be executed. This is also a benefit since many services require very low latency in regards to response to an action. In areas like banking, traveling, energy, health, I.o.T. applications are playing a huge role in effective computing with as low latency as possible [16].

Security and Privacy:

I.o.T plays a significant role in the security and privacy conversation since the vast amount of data involved is sensitive personal information that can create personal safety. In essence, Fog Computing can have its own implementation of security and privacy to prevent hackers from stealing important information just as other security procedures that is used in the information technology environment [15] [16].

	Fog Modes closest to I.o.T Devices	Fog Aggregation Nodes	Cloud
Response time	Milliseconds to subsecond	Seconds to minutes	Minutes, days, weeks
Application examples	M2M communication Haptics2, including telemedicine and training	Visualization Simple analytics	Big data analytics Graphical dashboards
How long I.o.T. data is stored	Transient	Short duration: perhaps hours, days, or weeks	Months or years
Geographic coverage	Very local: for example, one city block	Wider	Global

Table 1. Explains the ability of Fog Computing individually and as nodes in relation to the Network and the Cloud

In conclusion, Fog computing allows for faster processing and communication since it is closer to the source of the data, decreases the risk for data theft traveling on the network, and increases the protection of the data, minimizing any network bottlenecks if the data becomes immense.

2.3 I.o.T. ARCHITECTURE

In this section we will define the I.o.T. structured layered system that most of our work took advantage of. The following will define each of the layer specifically and the table 2 [14] to follow shows what kind of devices and/or platforms that can be used at each layer. The layers of the Architecture are the following [17]:

- Hardware Layer (Fog layer)
 - The purpose of this layer is to encompass physical objects such as temperature sensors, or any kind of sensor that collects data such as embedded computers, actuators, displays for notifications, etc. Moreover, the purpose of this layer can be to process information, store, and or communicate the information or processed information with another interface.
- Communication Layer (Access Layer)
 - The purpose of this layer is to set up the communication required with the other platforms that data can be managed and or transferred to.
- Software Layer (Data interface Layer)
 - The purpose of this layer is to store the compiled information received from the hardware layer so that it organized efficiently for the application layer.
- Application Layer
 - The purpose of this layer is to realize a form of data visualization. This layer can take shape in many different forms for various types of applications. This layer can be specific to the nature of the application.

□ Security Layer

- This is not a typical layer of the I.o.T. platform, although as mentioned in the above section, the implementation of security at any layer can be beneficial especially depending on the kind of data that is being collected.

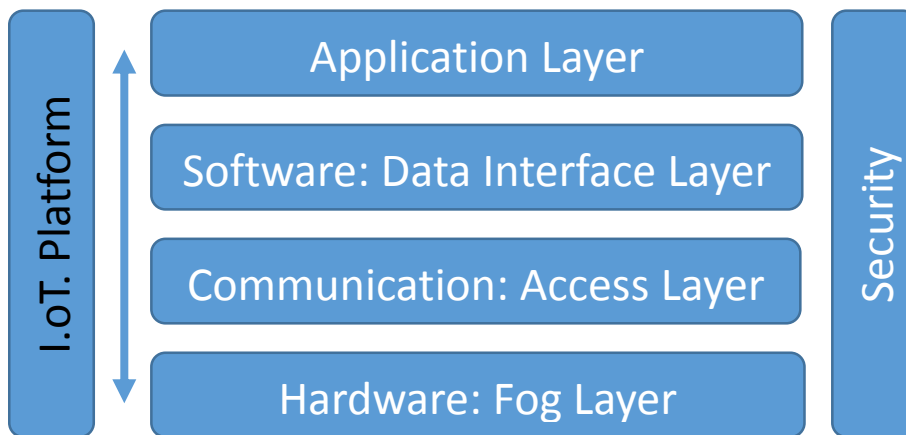


Figure 2. I.o.T. Platform

I.o.T. Elements		Samples
Identification	Naming	EPC, uCode
	Addressing	IPv4, IPv6
Sensing		Smart Sensors, Wearable sensing devices, Embedded Sensors, Actuators, RFID tag
Communication		NFC, UWB, Bluetooth, BLE, IEEE 802.15.4, Z-wave, WiFi, WiFiDirect, LTE-A
Computation	Hardware	SmarThings, Arduino, Phidgets, Intel Galileo, R-Pi, Smartphones, BeagleBone, etc.
	Software	OS(Contiki, TinyOS, LiteOS, RI.o.T. Os, Android); Cloud(Nimbits, Hadop, etc.)
Service		Identity-related(shipping), Information Aggregation(smarter grid), Collaborative-Aware(smarter home)Ubiquitous(smarter city)
Semantic		RDF, OWL, EXI

Table 2. Devices and protocols of the I.o.T platform/Layers

2.4 HEALTHCARE AND I.o.T.

I.o.T. is being integrated in many different kind of industries. One of the applications relevant to the topic of this thesis is the healthcare field. In fact, many of the applications of the healthcare field pertaining to I.o.T. are mobile applications and wearable devices that collect bodily information for many different kinds of diagnosis and interpretation. Furthermore, regarding how prevalent healthcare is and will become, not only are there economic benefits in the healthcare field when it comes to I.o.T, but also health factors with less stress on physicians and healthcare providers to manage their patients.

In the healthcare realm, it is crucial for healthcare providers to supplement their practice with technology to maximize effectiveness of treatment. The treatment that I.o.T brings to the healthcare industry and in our research is the ability to monitor the patient and diagnose a patient from a remote location. One of the ways to accomplish this is by the means of wearable devices and their ability to also connect to the network. From versatile patient monitoring such as smart-watches, wearable devices or even mobile devices, the healthcare industry tremendously benefits from I.o.T. especially when finances are the topic of conversation. Figure 3, illustrates the potential of the healthcare application in I.o.T.

According to the Kaiser Family foundation, hospital stays are costly; the average daily cost for a patient's one night stay in the hospital was over \$1,700[18]. For most of population this can be very bearing. Remote monitoring can be beneficial in minimizing such high costs since wearable devices such as smart watches and fitness trackers are relatively inexpensive. Another cause for concern in the health care industry

is the cost of drugs. According to Forbes, an average of \$55 million is what it takes to develop an approved drug. Also, there are others who are making a multi-million-dollar profit for making unapproved drugs. It is a possibility that using the I.o.T. system, some of these costs may be minimized [19]. Some industries are embedding technology in drugs to keep track of them at different stages of the manufacturing and development process [20]. Other companies like Proteus Digital Health have developed pills that track the consumption if and when a patient has consumed their medication [21].

Even with all of aforementioned possibilities of the I.o.T. based systems in the healthcare field, security is still a major concern. Some systems in the I.o.T realm are being developed in such a way that when sensitive data is being collected for privacy measures, specific information about the patient is intentionally not managed in no other place rather than the Fog itself. For example, in our research, we do not allow for the storage of the patients' data to be stored on the Fog computer for more than a week. We also make sure that when the information is passed to the node or Fog computer that belongs to the patient, we encrypt the connection before the data is passed to the edge node for further analyses.

Lastly, according to I.O.T. Analytics, wearable technology and connected health technology (Tele-health) is increasing rapidly according to figure three [22]. This was part of our motivation to take on this thesis research project.

This section covered some of the benefits and possibilities of the I.o.T systems in the healthcare industry and some of the concerns and how for our research we chose to develop it.

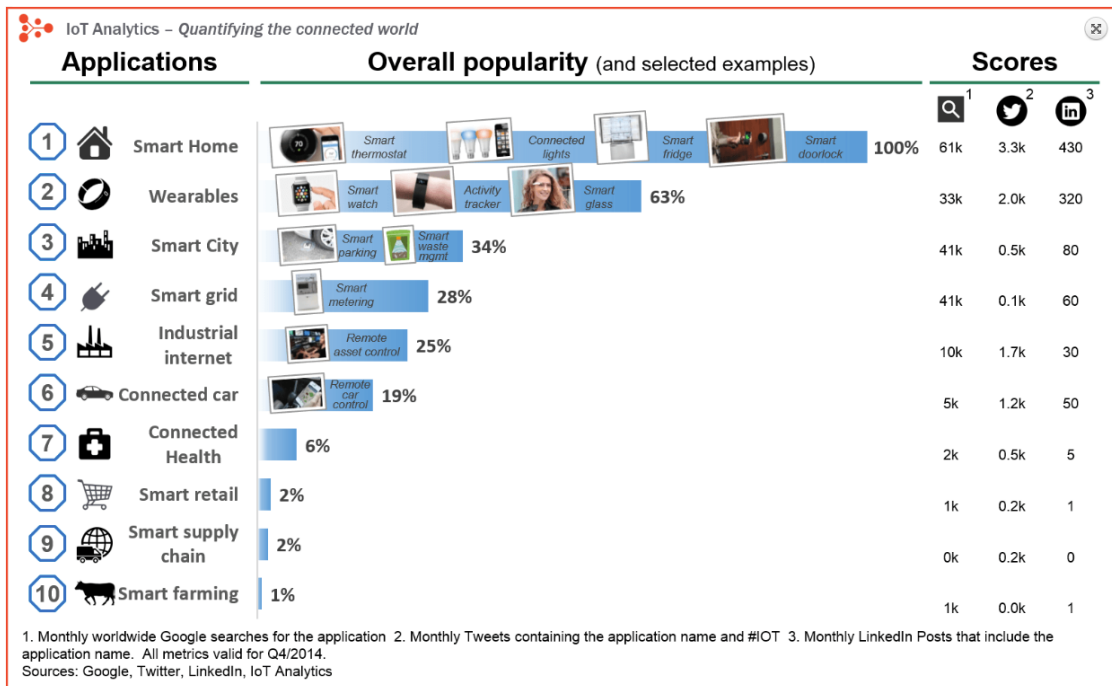


Figure 3. The Internet of Things applications ranking

CHAPTER 3

METHODOLOGY

In this chapter, we describe what each layer of the I.o.T platform that was used in our overall scheme of the research thesis. The structure of the chapter follows just as the I.o.T platform is designed for most projects. The first section will explain the hardware layer in our project and what hardware we used exactly. The second section focuses on the middleware, the mechanism which explain the interconnection of the software communication that took place during our research project. Lastly, we will focus on the application layer which focuses on the platform we used to accomplish this layer.

3.1 HARDWARE LAYER

At this level, the edge is another term used for Fog computing, we have been working with a particular hardware for over a year now. We have learned a great deal from working with this device. It is a remarkable device that has been designed and developed by Intel for specific uses in an I.o.T based system.

According to Intel, the intel Edison is ‘designed for expert makers, entrepreneurs, and some industrial I.o.T. applications, the Intel® Edison compute module provides ease-of-development for a range of prototyping projects or commercial ventures when performance matters [23]. The Intel Edison is a Linux OS based on the Yocto operating system, so you can compile C/C++ files, run Python, Node.js, java, goLang, and other kinds of scripts.

The Intel Edison was chosen for our research because of its size and configurability capabilities. The essential components of the Intel Edison Board are that

it sports a System on a Chip (SoC) 22-nm Intel SoC that includes a dual-core, dual-threaded Intel Atom™ CPU at 500-Mhz and a 32-bit Intel Quark™ micro-controller at 100 MHz. It has a random access memory of 1 GB LPDDR3 POP memory (2 channel 32bits @ 800MT/sec), a flash storage of 4 GB eMMC (v4.51 spec), wireless Broadcom 43340 802.11 a/b/g/n; Dual- band (2.4 and 5 GHz) with the option for on board antenna or external antenna [23]. Its compactness brings the wireless and Bluetooth 4.0 hardware. Another reason we used the Edison as the middle man Fog computer, was that we are able to increase its storage with an expansion slot which the Edison itself can be attached to an Arduino breakout kit board. The dimensions of the Edison are 60mm x 29mm x 8mm (with it attached to the mini breakout board). Additionally, the Edison is a low power embedded board. The power consumption of the Intel Edison is 3.3V - 4.5V @ ;1W [24]. Attached is a table description (table 3[24]) of some tasks and their power consumption current and voltages.

The Fog-driven cloud can provide complex computing because is has the capability for massive data storage, which in our case we decided to have a backup of data storage on the Fog computer along with the ability of placing it on the database. This ability of the cloud can also be utilized to perform long-term analysis including pattern recognition, machine learning and deep learning.

Activity	Power(mW)	Voltage(V)	Current(mA)
Booting up (peak)	984	8.2	120
Using Wifi 5GHz	680	8.5	80
Typing (serial port)	346	8.65	40
Idle	88	8.8	10
Powered off	45	8.9	5

Table 3. Power consumption of the Intel Edison during various operations



Figure 4. Intel Edison Fog Computing Board (Left: Edison with the Mini breakout kit, Right: Arduino Breakout kit with the Edison)

In conclusion, because of its compact size, its capability, its power consumption, and its configurability to change size, the Intel Edison was chosen to be our Fog Computer.

3.2 I.o.T. MIDDLEWARE

The purpose of the middleware is to provide means of communication between the software and hardware involved in the I.o.T. platform to perform specific machine to machine tasks. According to Digital Service Cloud, the role of the middleware is to allow and enable the interaction of the internet of things hardware and software to allow for smoother communication [25]. In the case for our research thesis, we took on the challenge to design a seamless as possible software that expert users will have as little as possible knowledge but be able to interact with the system easily.

3.2.1 COMMUNICATION/ACCESS LAYER

The purpose of this layer in our study was to make sure that the data that we were collecting was being able to communicate in a secure way to the Fog computer. Furthermore, we wanted to create a haven where the patients' collected information was not jeopardizing and it was still private in whichever means it was communicated. We had to deal with this in two stages. The first stage was how we received the data collected. For our application, the data collection happened in the following way: we created a mobile application of which we utilized the microphone sensor to collect audio data which our system is built upon. Since our framework is built upon having audio data, we developed a system which as soon as the data is recorded on the phone it is able to be deleted and instantly thereafter a connection is made with the Fog computer. Our development was to create a server client implementation. On the Intel Edison, we developed an encrypted server that always listened for a connection and received an audio file. Once the file was received, we were able to proceed with the file for further processing. The second stage where the communication layer was implemented when

all of the extraction of audio files were processed. At this stage, once the extraction of the all of the important information is processed, a connection with our MYSQL database is made. Once the connection is successful, we are then able to upload the extracted data to the correct table in the database based on the on the ssid (Service set identifier) of the Edison, which is unique for every Intel Edison.



Figure 5. View of the Communication layer implementation

3.2.2 SOFTWARE/DATA INTERFACE

In the software layer of the research thesis, we utilized numerous software packages and framework to complete the task. In this section we will explain each of the software/data interface that was utilized.

On the Fog computer, we used the following software to complete the task of voice processing, extracted data storage, data transfer and upload, etc.:

- Christian Herbst Library for Voice Analysis
- Praat tool (doing phonetics by computer)
- ftplib library for file transfer protocol for the audio file
- tinyDB (Database for Sensor Networks)

Each of the software or applications have the following function on the Fog Computer:

- Christian Herbst library for voice analysis is a collection of open source Python scripts made by an Austrian voice scientist which can be used for analyzing data from human mammalian vocalizations. It also used for graphic and visual representation but since we are using an embedded system that function is not available. We were able to use some of the functions necessary from this library and others we had to copy and customize to fit our needs. The library is The following functions are a part of the scripts [26]:

- general utility scripts (generalUtility)
- digital signal processing (DSP) functionality (dspUtil and myWave)
- interacting and utilizing Praat (praatUtil and praatTextGrid)

- functionality for creating graphs ([matplotlibUtil](#))
- Praat is a free scientific open source software package tool which is used for analysis of speech in phonetics. It is a tool that is commonly used by SLPs and linguistics. It provides multiple capabilities ranging from spectrographic analysis, articulatory synthesis, neural networks, etc [27].
- ftplib is a python module which defines and implements file transfer protocol class in python. Just as a normal FTP client, ftplib allows us to do things such as list directories, download and upload files from and to the server, and do multiple other tasks. The reason why we decided to use this is that it gave us the ability to save any of the audio file that the Fog computer processes on [28].
- TinyDB is compact and lightweight database that can be used for query processing for information extraction [29]. TinyDB is packaged in many different ways. One of the ways is that that it is also a module for python. TinyDB, much like MySQL, offers a simple interface to perform queries just as one would perform with SQL. The reason that we used TinyDB instead of using other applications like mariaDB is because it allows for power efficient in-network processing algorithms [29].

3.3 APPLICATION LAYER

The application layer of our I.o.T. based system for the research thesis involved making an application so that all users of our study could collaborate and interact with one another. In this section we explain the application we have been developing and intend to continually develop.

We decided to make an Android application for our study. The purpose for using an Android application was to allow the use of software which made data collection seamless with little effort, and a unique profile and task for each user. In our case there were three different kind of users since each user had different tasks to complete throughout various steps of the application process. Figure 6 shows the hierarchy of the user privileges for the three kind of different users for the desired design scope of the research thesis. In the flowchart, we can clearly see the design scope of the privileges that took place for each of the user and also the account authentication structure.

We used a restful service to program the application so that we could gather information. REST (Representational State Transfer) is an architectural style of information transfer using HTTP methods such as GET, POST, PUT, and DELETE to allow HTTP commands to perform different actions depending on the backend server programming implementation [30]. This is very useful and beneficial since it is widely used by all commercial devices. We implemented a restful service since we wanted to create an architecture where registration was possible, authentication was possible, and data gathering for visual purposes was also possible. To use the RESTful service, all we had to do was use the HTTP library embedded in Android.

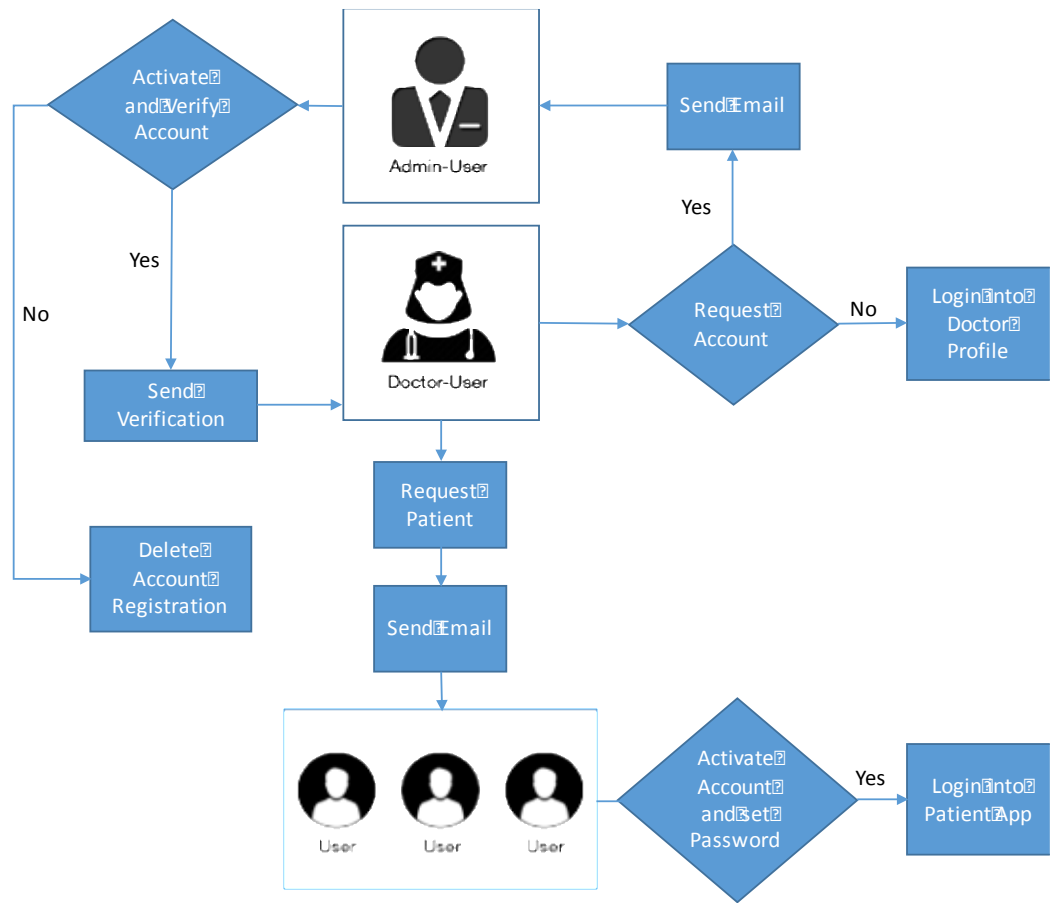


Figure 6. Flowchart of the user authentication and privileges for specific accounts

CHAPTER 4

DESIGN AND IMPLEMENTATION / FINDINGS

In this section, we explain and present all of the design and implementation of the research thesis from the Fog Computer, the Database, to the mobile application. We first explain the overall structure of the study and the involvements of each front and back end user.

4.1 OVERALL ARCHITECTURE DESIGN

For the overall research, the beginning idea and approach began with Fog computing only. As time went along we started to think bigger: to the cloud since it offers the ability to save massive amounts of information. We then realized that we had to create a complete I.o.T based system where all the platforms were developed and functioning. We realized that there had to be multiple users, end-users and front end-users. In figure 7, we see the overall structure that we set our goals on to develop in an

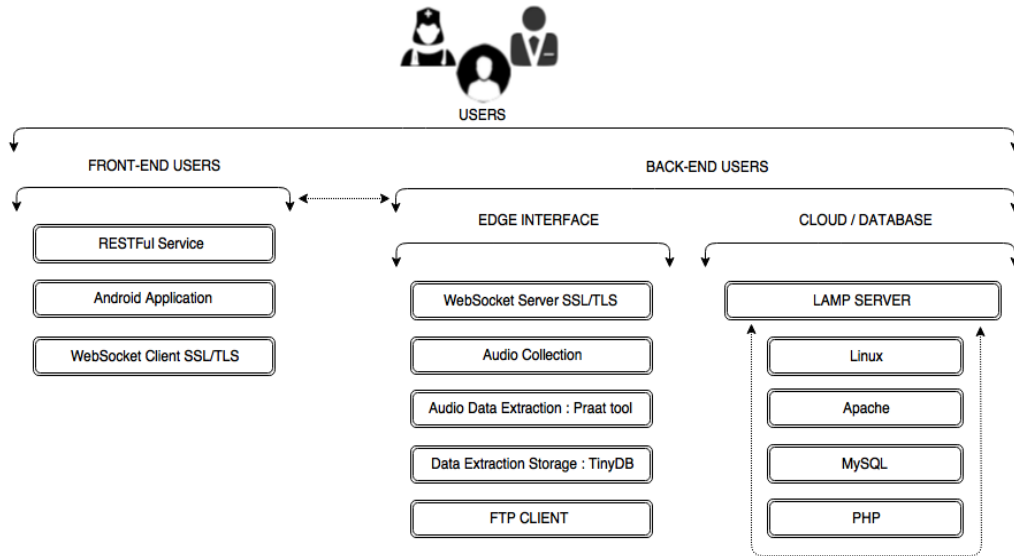


Figure 7. Overall Structure of the I.o.T Platform overview overview method. In figure 8, we are able to see a more detailed and descriptive view of the overall design.

4.2 HARDWARE (EDGE)DESIGN AND IMPLEMENTATION

We developed the hardware layer Intel Edison as our main Embedded network connected board. There were numerous libraries, modules, and software that were installed on the Edison so that we could implement and program the overall hardware design and interface of the Fog Computer.

For the first programming task, we had to make sure that we had to transfer the recorded audio file by the patient user. To do this, we needed to understand a couple of programming concepts and protocols. These protocols were to use socket streaming using TCP wrapped in SSL/TLS (Secure Socket Layer/ Transmission Layer Security) sockets for security.

Sockets provide the communication framework for devices using different protocols such TCP and UDP which could then be wrapped in secured sockets. TCP, Transmission Control Protocol, is a networking protocol which requires guaranteed and reliable delivery and in order files. TCP is a protocol that is connection oriented and bi-directional which means both devices can send and receive files sent to one another. Each point of the connection involves an I.P. (Internet Protocol) address and a port number [31] so the connection can be made with a specific device. Furthermore, (for the reasons of security on the privacy sake of the users' information that is being collected), we wrapped the TCP sockets in SSL Sockets. SSL (Secure Sockets Layer) is network communication protocol that allows for encrypted authentication for network sockets from the server and the client side. This connection protocol allows for the data transfer that server and client to remain private and secure. To complete this task, we used the python module: SSL and socket. Additionally, to create the certifications for

the server and the client, we used a command line program called OpenSSL (an open source project that provides a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols [32]).

Socket implementation with SSL :

```
key, cert = "/home/root/Coding/certificates/server/privKey.key", \  
            "/home/root/Coding/certificates/server/server.pem" # Key  
and cert created with OPENSSL  
context.load_cert_chain(certfile=cert, keyfile=key) # Load the  
certifications  
value = '''ifconfig | grep "inet " | grep -v 127.0.0.1 | grep -v  
192.* | awk '{print $2}' '''  
ip = get_IP_Edison(value) # get the IP of the Edison  
HOST= "172.30.79.191"  
PORT = 5000 # calling the port and host / needs to be of the edison  
bindsocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #  
create a normal socket  
bindsocket.bind((HOST, PORT)) # Bind the socket or create it  
bindsocket.listen(1) # make the socket listen on five connections  
print 'Listening..on IP:' + str(HOST) + " and port:" + str(PORT)  
while True:  
    newsocket, fromaddr = bindsocket.accept() # accept the  
connection  
    print 'accepted connection from' + str(fromaddr)  
    connstream = context.wrap_socket(newsocket, server_side=True  
, do_handshake_on_connect= True) # Create the SSL Socket and wait for  
connection
```

Once all the SSL certification keys (for client and server being the Android device and Intel Edison respectively) were built, we just ran the secure sockets on the server (Intel Edison) and continuously listened for a connection so that we can receive a file. We then renamed the file with the date and time so that it can be ready for processing (See code above with comments).

As soon as the audio file was completely transferred, the connection is closed and the processing begins. For the processing and analysis, we utilized the Christian Herbst python library and praat tool which the python library takes advantage of (see chapter 3, section 3.2.2). The audio used for the development and implementation are described

in table 3. Some of the processing parameters (averages of parameters) that we calculated from the audio files are:

- Perceptual Loudness:
 - Loudness is the perceived intensity of an audio signal. It is a mathematical quantity based on various auditory models.
- Zero-Crossing Rate:
 - Zero crossing rate is used to analyze and recognize voiced-unvoiced parts of speech. It is a number of times the speech signal magnitude crosses the value of zero [33].
- Spectral Centroid:
 - Spectral centroid computes the center of gravity, the frequency area that most of the signal energy is focused. It gives indication to distinct sounds that are produced which sound very 'bright' [34]
- Short-time Energy
 - Short-time energy was used to provide a representation that reflects amplitude variations of unvoiced and or voiced segments of an audio file [33].

Tasks	Description	Clinical Measures
T1	Pronouncing sustained "Ah" sound.	Loudness, ZRC, SC, STE
T2	High-pitch (rising) /a/ sound (5-sec limit)	Loudness, ZRC, SC, STE
T3	low-pitch (falling) /a/ sound (5-sec limit)	Loudness, ZRC, SC, STE

Table 4. Selected speech tasks for the Fog Computer

Other parameters using the Christian Herbst python library that we were able to calculate were: Jitter ratio, average harmonicity, period variability index, root mean square, and fundamental frequency. Parameters such as jitter, shimmer, shimmer percentage, and many more will be additional parameters to be processed in the future.

We are thankful to the patients with Parkinson's disease for their co-operation in using the EchoWear and FIT technology. The presented work is supported by a grant (No: 20144261) from Rhode Island Foundation Medical Research.

4.3 DATABASE APPROACH/IMPLEMENTATION OF BACKEND

From the overall concept of the overall structure from figure 7 above (section 4.1), in the back-end users section, the database/cloud server implementation was implemented with PHP and MySQL. First, we had to set up a LAMP (Linux, Apache, MySQL, PHP) server which is an open source web platform for development on the Linux system using Apache for web servicing, MySQL as the database system for management and storage, and PHP the language for server interaction with applications [36].

The heart of the matter of the back-end design is the relational database development. We designed a database that revolves not around the users of the I.o.T. based system but also one that the Fog computer could easily interface with the database. The design concept of the database for our system was at first based on the users. We created three tables that were only about the users (patients and healthcare providers). One table was created for the data extraction information of the patient. The following will explain each table and all of their field content and attributes.

Model of table with description for each filed in the Tables of the database:

- Users Table:
 - idUser: (INT:11): identification for any kind of user (healthcare provider or patient) auto-incremented
 - firstName: (VARCHAR:45): first name of healthcare provider or patient
 - lastName: (VARCHAR:45): last name of healthcare provider or patient.
 - password: (VARCHAR:45): password of healthcare provider or patient.
 - email: (VARCHAR:45): email of healthcare provider or patient.

- userType: (INT:11): type of user: 1 for doctor, 0 for patient.
- verification: (TINYINT:1): Account verification for the healthcare provider or patient value, 1 for Active, 0 for not activated.

□ Doctors Table

- id_doctorsTable: (INT:11): auto-incremented identification for healthcare provider.
- Users_idUser: (INT:11): identification of healthcare provider in the users table.
- department: (VARCHAR:45): department that healthcare provider belongs to.

□ patientsTable

- id_patients: (INT:11): identification of patients in the patient table.
- Users_idUser: (INT:11): identification of patient in the user table.
- doctorsTable_id_doctorsTable: (INT:11): identification of healthcare provider for the patient that he/she belongs to.
- notificationAlert: (INT:4): notification time of how event patient needs to record and audio.
- edison_ssid: (VARCHAR:6): Service set identification number of the Intel Edison that belongs to the patient.

□ dataTable

- iddataTable: (INT:11): identification of the inputted extracted data of the audio file.

- patientsTable_id_patients: (INT:11): identification of the patient id who the data belongs to.
- Fundamental_Frequency: (INT:11): fundamental frequency calculated of the audio file.
- Intensity: (INT:11): average intensity calculated of the audio file
- fileDuration: (INT:11): length of the audio file.
- time:(TIMESTAMP): time at which the dataset is inputted in the data table.
- ZRC: (INT:11): average Zero crossing rate of the audio file.
- SC: (INT:11): average Special Centroid of the audio file.
- STE: (INT:11): average short time energy of the audio file.

All of the above information can be seen in the following figure (8), where all of the tables involved relate to one another. All of the users are entered in the users table, all of the healthcare providers are entered in the doctors table plus the department they belong in. All of the patients are entered in the patients table where they have also an identification of which healthcare provider they belong to. In the data table the extracted audio information is entered from the processed audio file on the Intel Edison.

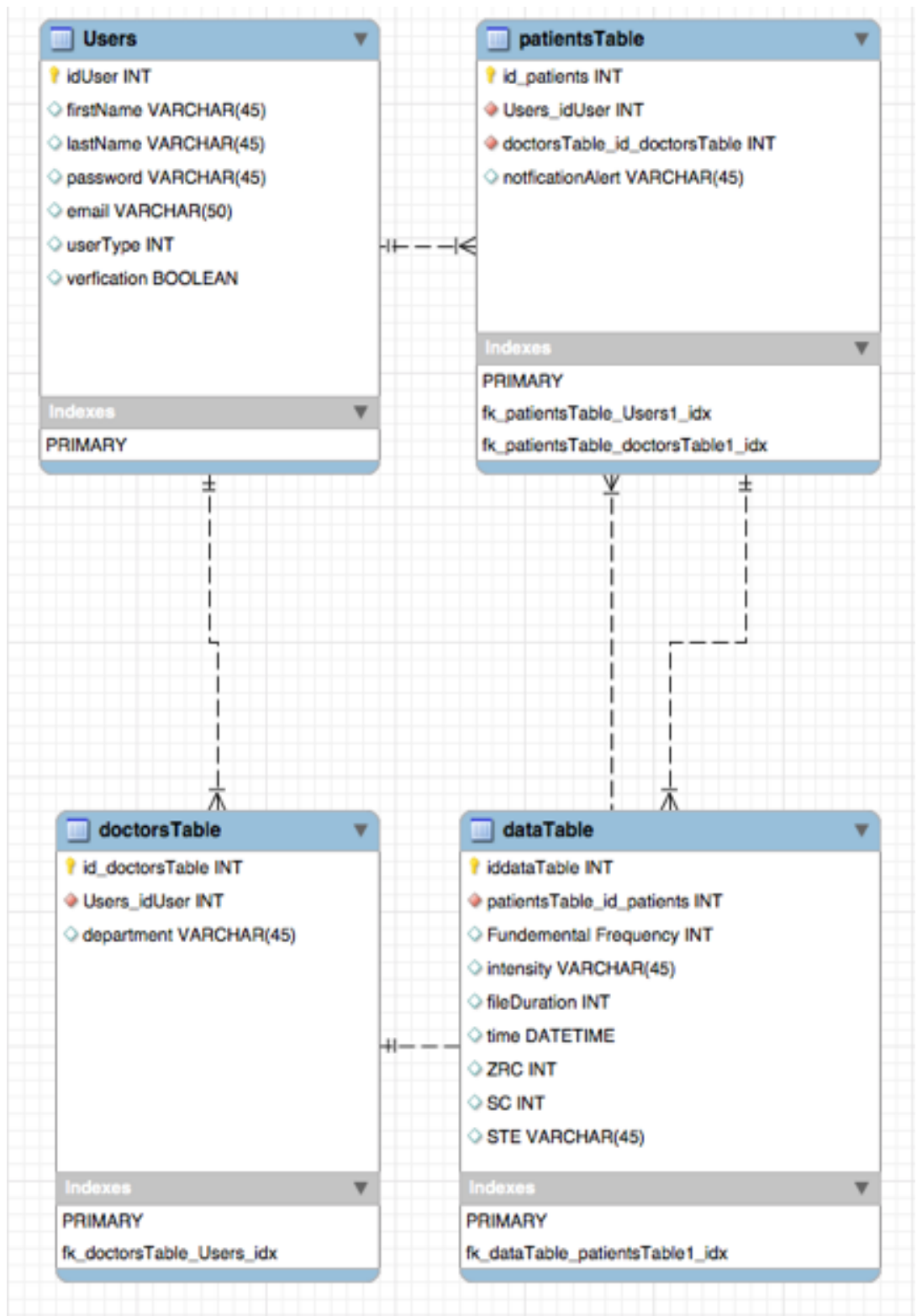
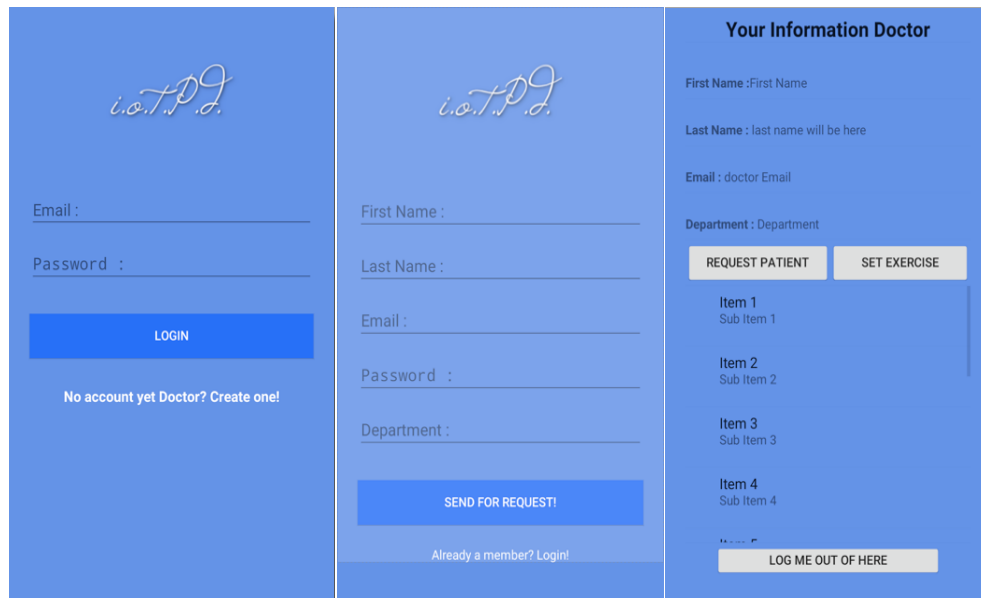


Figure 8. Database Schema

4.4 APPLICATION DESIGN/IMPLEMENTATION OF FRONT END

For the front-end user implementation, as previously mentioned, we designed an Android application so that both users and healthcare providers could interact with one another. Also, as mentioned before, we developed our app to take advantage of our web service by using the REST protocol (see chapter 3, Section 3.3 for more information). The reason we used REST is because we don't have complicated data operations taking place and it is simpler to implement for future development. For every REST request of data information gathering, we returned a JSON ((JavaScript Object Notation) which is a format of data-interchange between programs [37]) object with information that we deciphered in the android application.

The application allowed for the access to the two kind of users for almost every



1 Login Page

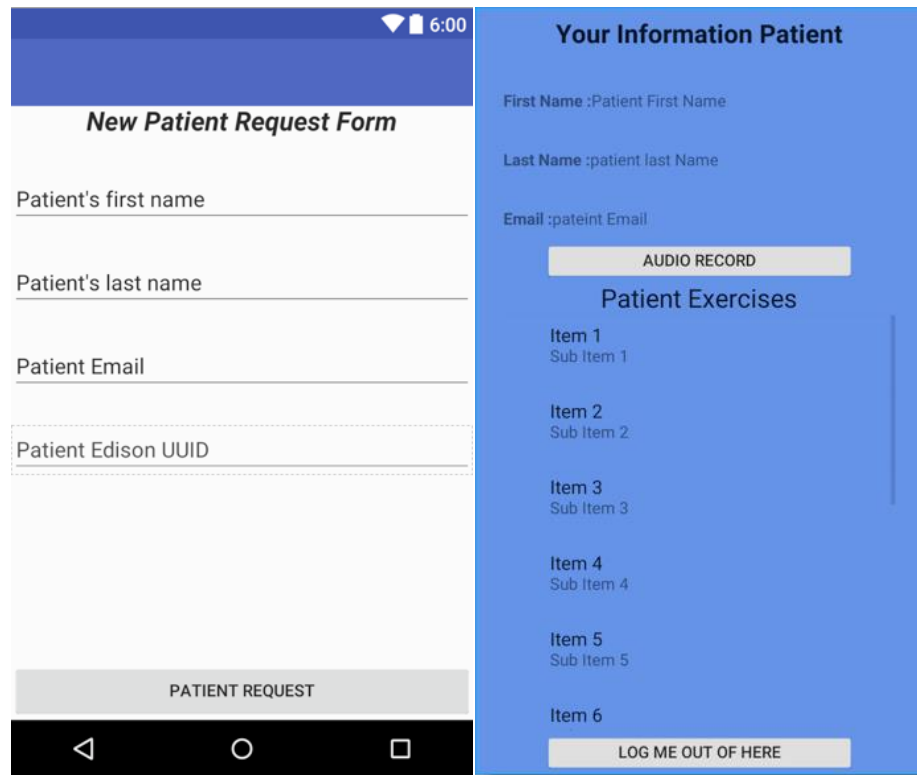
2 Registration for Doctor

3 Doctor's Profile

Figure 9. Android application designs

action except with a couple of exceptions as seen in the figures below. Both of the patients and healthcare providers are allowed for login and viewing their profile, however the profiles are different in it that the doctor allows for a patient to be registered

and set exercise notification for the patient. The patient though, can only view their information and their visual data. Please review figure 6 above again for a visual of this. Following are screenshots of some of the app development that we have implement and have functional thus far:



4 Doctor requesting patient page

5 Patient's Profile

Figure 10. Android application designs

App Page	Description
1 Login	Login page for the doctor and/or patient as long as they are verified
2 Doctor's Registration	Registration page for the doctor, based on the privileges structure(see figure 6)
3 Doctor's Profile	Profile view of the doctor with all the doctor's information and other Doctor's actions for the patient
4 Doctor requesting Patient	Page that the doctor can request for a patient so they can have an account
5 Patient Profile	Profile of the Patient with all their information and an option to record audio data, and all of their exercises which they can view

Table 5. Description of some of the Android application designs

CHAPTER 5

CONCLUSION

5.1 DISCUSSION AND RESULTS

The results obtained while conducting the research was produced as a result of the processing that we were capable to program on the Intel Edison. Some of the results from the processing audio files included calculating the Intensity of the audio files. In figure 11 we see the instantaneous loudness of speech signals computer by the Fog computer. The figure illustrates five different normal ‘ah’ pitch audio files (see table 4

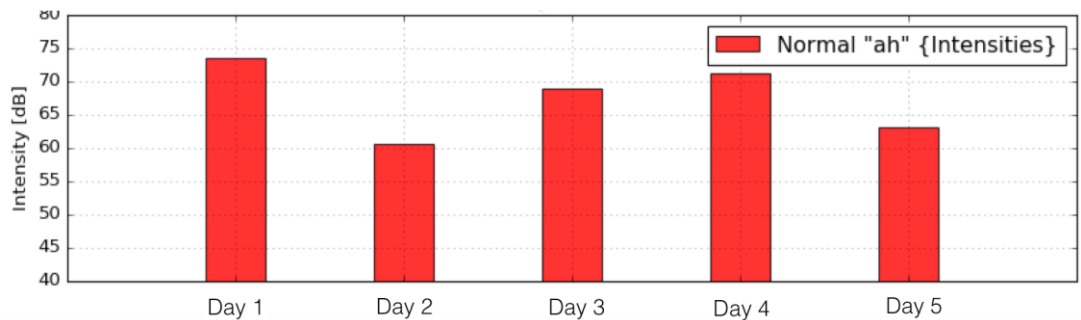


Figure 11. Pathological loudness computations for one of the participants with Parkinson's disease

in section 4.2 for explanation) throughout a week that the intensities were calculated from. Furthermore, since we had the ability to conduct multiple experiments from Parkinson’s patients in different instances of time, we were able to also visualize the

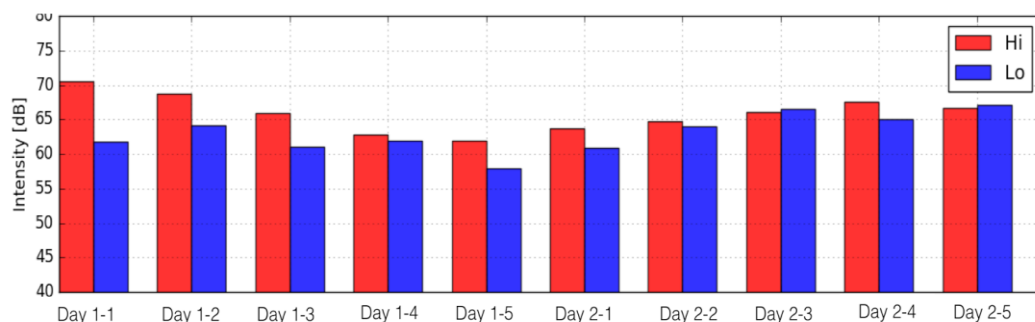


Figure 12: Few samples of average loudness calculations for the task 2 (rising pitch) and task 3 (falling pitch) performed by one of the participants. These calculations were performed by

results of intensities. Figure 12, shows an example of multiple exercise audio files with low and high pitches for a 2-week time period (see table 4 in section 4.2 for explanation).

Other clinical speech features such as zero crossing rate (ZCR), the spectral centroid (SC), and short-time energy (STE) (for more information on the features, please review section 4.2) were also parameters that we were able to successfully process from the audio files. As a result, figure 13 illustrates an example of an audio file’s amplitude signal and the above mentioned features calculated. Furthermore, just with the

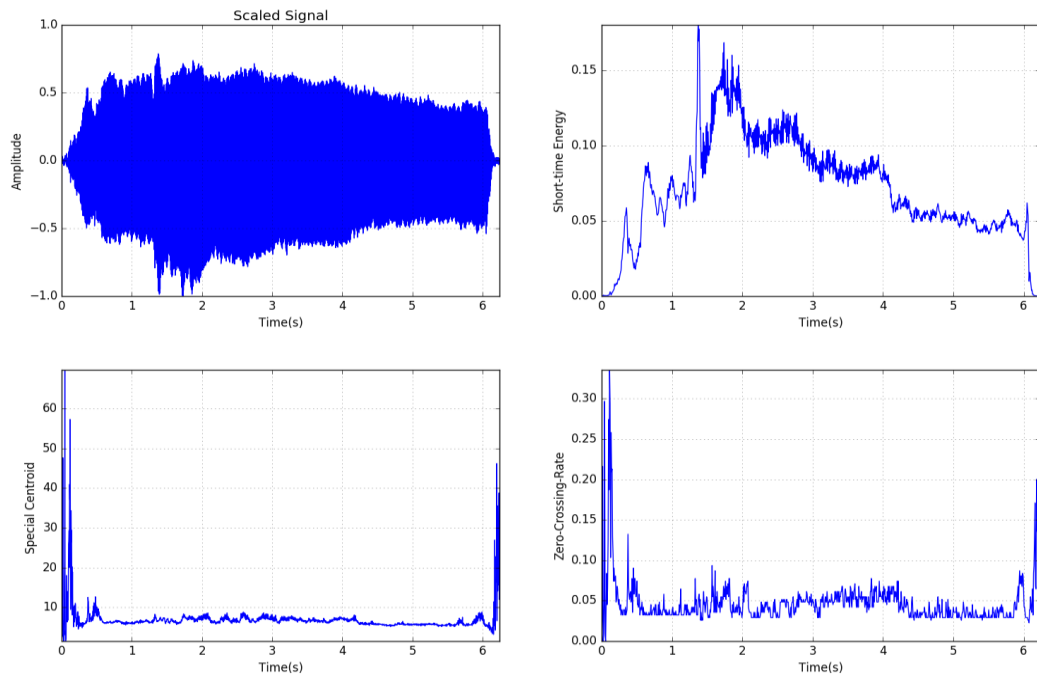


Figure 13. Clinical features computed on the Intel Edison

intensities calculated for numerous audio files, we did the same with ZRC, SE, STE parameters where the average for each audio file is calculated. Figure 14 illustrates the averages of the SRC, SE, and STE parameters that were computed on the Fog computer.

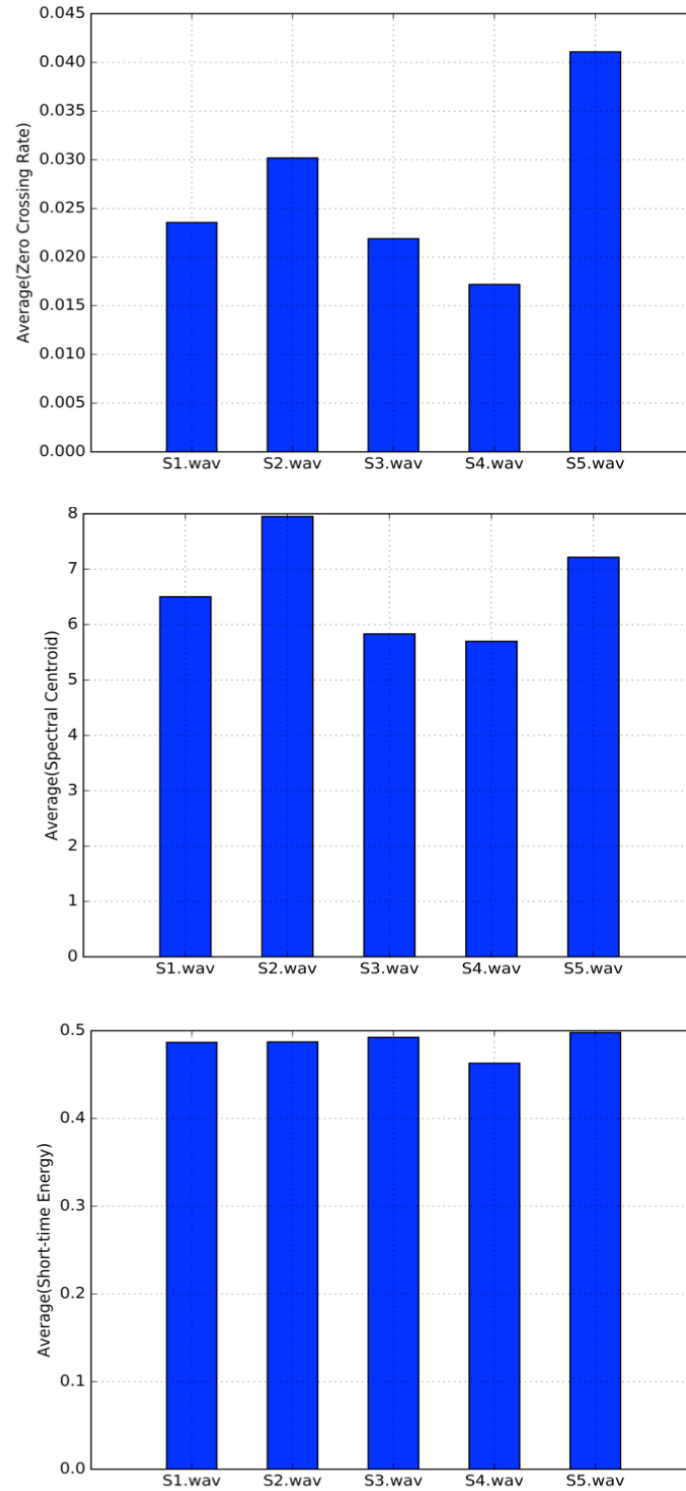


Figure 14. Fog computations of three clinical features for speech files.

The results obtained while conducting the research can be seen above. The results provide and enables us with the the analysis of speech signals in remote non-clinical settings such as homes and possibly other locations with network connectivity. To compute the above results, we utilized the Intel Edison as the Fog computer with a service-oriented architecture to collect, process and extract clinically-relevant features, and store the information extracted. We demonstrated as part of the results from our research that the Fog computer in an I.oT. based system provides great computational intelligence, configurability, and interoperability. As a proof this, we benchmarked the calculations of the clinical features and the results are shown in Table 6.

Speech Tasks	Processing Time(s)	File Duration(s)	Size (kB)
S1	2.34	6.24	551
S2	2.33	6.18	545
S3	2.12	5.62	496
S4	2.28	6.08	537
S5	1.86	4.96	438
Total time	10.94	29.08	2567

Table 6. Benchmarking the fog computer for computing the clinical speech features: zero crossing rate, special centroid, and short-time energy

5.2 CONTRIBUTIONS

The goal of this research was to design a full course I.o.T platform from the ground up involving every layer implementation. Our aim was to develop an I.o.T service surrounding a database-centric approach with multiple user interactions. We decided to develop an application also revolving the database-centric approach. Furthermore, most importantly, we developed the I.o.T system with Fog computer in mind to handle the data collection, processing, storage, etc. Essentially we utilized the Fog to be the main hub of our research. From first processing audio files which was capable of handling complex signal processing algorithms such as psychoacoustic analysis to extract clinical features, to storing the raw data and features that are query-able by the cloud and on the Fog by demand. The Fog is a useful layer in the platform that can have plenty of benefits especially in machine-to-machine interaction.

5.3 FUTURE RESEARCH

Our I.o.T. based system includes all of the functional components that are required, however, much work still needs to be done to perfect the system to function seamlessly. The following is a list of content and direction that the research can take:

- Although security is implemented at audio data transfer, upgrading the encryption would be beneficial
- A new form of Edison search and connection to send data needs to be implemented instead of giving the exact I.P. for the Edison.
- The android application could implement some of the encryption instead of server code implementing this since information could be hacked when an http request happens at login or register.
- Create an actual API for the mobile application to use instead of making a single request every time are posting to requesting data.
- Implement a notification system that the patient is notified when to complete a task.

5.4 TOOLS USED

Languages	Tools used	Usage
MySQL	SequelPro, DataGrip	Run Queries and check Tables for testing purposes
PHP	PhpStorm, Postman	Code the server code to return the data and information to the mobile application
Python	PyCharm	Code the data Processing, storage, transmission, and interfacing with database
Android	IntelliJ, Android Studio	Coded the Client transfer code and the complete mobile application
All languages	Atom, Sublime Text2	Editors which can code all languages

Table 7. Tools used for Implementation of the Smart Computing Interface

BIBLIOGRAPHY

- [1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. “*Fog computing and its role in the internet of things.*” Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp.13–16.
- [2] L. Darrell, “*Fog to replace cloud; cisco seeks early mover advantage.*” July 2014. July 31, 2014. <<http://www.bidnesstc.com/business/fog-to-replace-cloud-cisco-seeks-early-mover-advantage/Ded>>.
- [3] Dastjerdi, Amir Vahid, et al. “*Fog Computing: Principals, Architectures, and Applications.*” *arXiv preprint arXiv:1601.02752* (2016).
- [4] Luan, T.H., Gao, L., Li, Z., Xiang, Y., We, G. and Sun, L., 2016. “*A View of Fog Computing from Networking Perspective.*” *arXiv preprint arXiv:1602.01509*.
- [5] Aazam, Mohammad, et al. “*Cloud of Things: Integration of I.o.T. with Cloud Computing.*” *Robots and Sensor Clouds*. Springer International Publishing, 2016. 77-94.
- [6] Nellore, Kapileswar, and Gerhard P. Hancke. "A survey on urban traffic management system using wireless sensor networks." *Sensors* 16.2 (2016):157.
- [7] Nguyen Gia, T., Jiang, M., Rahmani, A. M., Westerlund, T., Mankodiya, K., Liljeberg, P., & Tenhunen, H. (2015). “*Fog Computing in Body Sensor Networks: An Energy Efficient Approach.*” In *IEEE International Body Sensor Networks Conference (BSN'15)*.
- [8] Ahmad, Mahmood, et al. “*Health Fog: a novel framework for health and wellness applications.*” *The Journal of Supercomputing* (2016): 1-19.
- [9] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, “*Role of Middleware for Internet of Things.*” *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 2, no. 3, pp. 94–105, 2011.
- [10] Mahler LA, Ramig LO, and Fox C. “*Evidence-based treatment of voice and speech disorders in Parkinson disease.*” *Current Opinion in Otolaryngology & Head and Neck Surgery*, vol. 23, no. 3, pp. 209–215, 2015.
- [11] “*Health information from the National Institute for Deafness and Other Communication.*” <<https://www.nidcd.nih.gov/health/statistics/statistics-voice-speech-and-language>>

- [12] J. Gantz and D. Reinsel, “*The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east.*” *IDC iView: IDC Anal. Future*, vol. 2007, pp. 1–16, Dec. 2012.
- [13] Manyika, James, et al. “*Disruptive technologies: Advances that will transform life, business, and the global economy.*” Vol. 12. New York: McKinsey Global Institute, 2013.
- [14] Al-Fuqaha, Ala, et al. “*Internet of things: A survey on enabling technologies, protocols, and applications.*” *Communications Surveys & Tutorials, IEEE* 17.4 (2015): 2347-2376.
- [15] Cisco. “*Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are.*” Cisco. Cisco and/or its affiliates. 2015. 04 Mar. 2016. <http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf>.
- [16] “The Future of Cloud Computing: Fog Computing and the Internet of Things.” DATAVERSITY.” DATAVERSITY. 13 July 2015. 22 Sept. 2015 <<http://www.dataversity.net/the-future-of-cloud-computing-fog-computing-and-the-internet-of-things/>>.
- [17] “*I.o.T. Platforms Market Report 2015-2021. I.o.T. Analytics Market Insights for the Internet of Things.*” I.o.T. Analytics, Jan. 2016. 03 Apr. 2016. <<http://iotanalytics.com/product/iot-platforms-market-report-2015-2021-3/>>.
- [18] “*1999 - 2014 AHA Annual Survey.*” Copyright 2015 by Health Forum, LLC, an affiliate of the American Hospital Association. Special data request, 2015. <<http://www.ahaonlinestore.com.>>
- [19] Harpham, Bruce. “*How the Internet of Things Is Changing Healthcare and Transportation.*” *CIO*. CIO, 8 Sept. 2015. 03 Apr. 2016. <<http://www.cio.com/article/2981481/healthcare/how-the-internet-of-things-is-changing-healthcare-and-transportation.html>>.
- [20] Herper, Matthew. “*The Truly Staggering Cost of Inventing New Drugs.*” *Forbes*. Forbes Magazine, 10 Feb. 2012. 03 Apr. 2016. <<http://www.forbes.com/sites/matthewherper/2012/02/10/the-truly-staggering-cost-of-inventing-new-drugs/2/#2c09511e4856>>.
- [21] Abramovich, Giselle. “*15 Mind-Blowing Stats About the Internet of Things.*” *CMO.com*. CMO.com by Adobe, 7 Apr. 2015. 3 Apr. 2016. <<http://www.cmo.com/articles/2015/4/13/mind-blowing-stats-internet-of-things-iot.html>>.

- [22] Lueth, Knud. “*The 10 Most Popular Internet of Things Applications Right Now.*” *I.o.T. Analytics Market Insights for the Internet Of Things*. I.O.T. Analytics, Feb. 2015. 03 Apr. 2016. <<http://iot-analytics.com/10-internet-of-things-applications/>>.
- [23] “*Intel® Edison-One Tiny Module, Hardware Guide.*” Intel, <<http://www.intel.com/content/www/us/en/support/boards-and-kits/000005808.html>>. 16 Mar. 2016. 04 Apr. 2016
- [24] Hirsch, Michael. “*Measured Power Consumption of Intel Edison.*” *SciVision*. 1 Nov. 2014. 07 Apr. 2016. <<https://scivision.co/measured-power-consumption-of-intel-edison/>>.
- [25] Gupta, Mohit. “*Role of Middleware in the Internet of Things.*” *Digital Service Cloud*. Digital Service Cloud, 10 July 2014. 07 Apr. 2016. <<http://www.digitalservicecloud.com/iot-middleware.html> >
- [26] Herbst, Christian. “*Christian's Python Library: A Python Library for Voice Analysis.*” Laboratory of Bio-Acoustics, Dept. of Cognitive Biology, University Vienna. 04 Apr. 2016. <<http://homepage.univie.ac.at/christian.herbst//python/> >
- [27] Boersma, Paul, and David Weenink. “*Praat: Doing Phonetics by Computer.*” 04 Apr. 2016. <<http://www.fon.hum.uva.nl/praat/> >
- [28] “*21.13. Ftplib — FTP Protocol Client.*” *FTP Protocol Client*. Python Org. 04 Apr. 2016. <<https://docs.python.org/2/library/ftplib.html>>
- [29] “*TinyDB: A Declarative Database for Sensor Networks*”. University of California, Berkeley. 04 Apr. 2016. <<http://telegraph.cs.berkeley.edu/tinydb/>>
- [30] Karagiannis, Vasileios, et al. “*A survey on application layer protocols for the internet of things.*” *Transaction on I.o.T. and Cloud Computing* 3.1 (2015): 11-17.
- [31] “*What Is TCP/IP (Transmission Control Protocol/Internet Protocol).*” 07 Apr. 2016. <<http://www.computerhope.com/jargon/t/tcpip.htm>>.
- [32] “*OpenSSL.*” 5 Apr. 2016. <<https://www.openssl.org/>>.
- [33] Bachu, R. G., et al. “*Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal.*” American Society for Engineering Education (ASEE) Zone Conference Proceedings. 2008.

[34] Ramig, Lorraine O., et al. “*Changes in vocal loudness following intensive voice treatment (LSVT®) in individuals with Parkinson's disease: A comparison with untreated patients and normal age-matched controls.*” *Movement Disorders* 16.1 (2001): 79-83.

[35] “*LAMP Stack.*” 06 Apr. 2016. <<https://www.turnkeylinux.org/lampstack>>.

[36] “*Introducing JSON.*” 06 Apr. 2016. <<http://www.json.org/>>.