SMART SENSE SWITCH BOARD (S3B)

By

Ahsan Baidar Bakht
Usama Zahid Alvi
Rana Muhammad Fahad Aziz
Muhammad Daud Rajwana
Muhammad Haris Ishaque

Submitted to Faculty of Department of Electrical Engineering,

Military College of Signals, National University of Sciences and Technology, Islamabad

in partial fulfilment for the requirements of a B.E Degree in

Electrical(Telecom) Engineering

JUNE 2018

بسم الله الرحمن الرحيم

In the name of Allah, the Most Beneficent, the most Merciful

ABSTRACT

Smart Sense Switch Board(S3B)

In current scenario the technology world is taking a shift into smarter gadgets that have a logical sense to behave and make decisions on basis of some learned behaviors. The gadget to act in a smart logical way requires some pre-knowledge (data set) to make logical decision. The logical decisions are further made based on some logical algorithms.

The project (S^3B) aims to make the electrical switch boards smart in a sense that they can detect the device connected to it. The smart device(S^3B) will be trained on some pre-recorded data sets using a Python IDE running machine learning algorithms on it. Real time detection will be done using Raspberry Pi. Raspberry Pi will be running an integrated version of Python IDE. The device connected to PI will be connected via wireless connection for remote control and intimation about the electrical appliances connected to the smart device. User will be using an interfaced mobile app for the purpose.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis – Smart Sense Switch Board (S3B) carried out by Ahsan Baidar Bakht, Usama Zahid Alvi, Rana Mohammad Fahad Aziz, Muhammad Daud Rajwana and Muhammad Haris Ishaque under the supervision of Dr. Mir Yasir Umair for partial fulfilment of Degree of Bachelor of Electrical (Telecom) Engineering is correct and approved.

Approved By

Dr. Mir Yasir Umair

Department of EE, MCS

Dated: ____________ **____________**

DECLARATION

No portion of the work presented in the dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To great Muslim scientists and researchers of history whose accomplishments have greatly contributed to progress of this modern world and who are of great inspiration to us to excel in the fields of Science and Technology.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support and critical reviews. We would like to thank our supervisor Dr. Mir Yasir Umair for his continuous guidance and motivation throughout the course of our project. Without their help we would have not been able to accomplish anything.

Table of Contents

Chapter 1: Introduction 1

1.1 Overview: 1

1.2 Problem Statement: 1

1.3 Approach: 1

1.4 Objectives: 2

1.5 Deliverables: 2

1.6 Overview of Document: 3

1.6.1 Purpose: 3

1.6.2 Document Conventions: 3

1.6.2.1 Headings: 3

1.6.2.2 Figures: 3

1.6.2.3 References: 4

1.6.2.4 Links to web pages: 4

1.6.2.5 Basic text: 4

1.7 Intended Audience and Reading Suggestions: 4

1.8 References 5

Chapter 2: Literature Review 7

2.1 Preamble: 7

2.2 Demo abstract: PLAID: 7

2.3 Real-Time Recognition and Profiling of Appliances through a Single Electricity: 8

2.4 Neural NILM: 8

2.5 Conclusion: 9

Chapter 3: Design Specification 11

3.1 Introduction 11

3.2 Overall Description: 11

3.3 Product Features: 11

3.4 Operating Environment: 11

3.4.1 Hardware: 12

3.4.2 Software: 12

3.4.3 Design and Implementation Constraints: 13

Chapter 4: Hardware Implementation 15

4.1 Introduction: 15
4.2 Components Used: 15
4.2.1 Raspberry Pi 3: 15
4.2.2 Analog Discovery 2: 16
4.2.3 NodeMcu: 16
4.2.4 Step Down Circuit: 17
4.2.5 Arduino UNO: 18
4.2.6 Switch Board: 18
4.2.7 LCD Display: 19
4.3 Implementation Methodology: 19
4.3.1 Energy Monitoring: 19
4.3.2 How to Start with Raspberry Pi : 20
4.3.2.1 Raspberry Pi Pin Configuration: 22
4.3.2.2 Voltages 22
4.3.2.3 Outputs 22
4.3.2.4 Inputs 22
4.4 Remote Monitoring and Switching: 22
4.4 Recording Data Sets: 23
4.4.1 Recording of Values: 24
Chapter 5: Software Implementation 27
5.1 Concept of Machine Learning: 27
5.1.1 Concept of Supervised Learning: 27
5.2 Machine Learning in our Project: 27
5.2.1 Independent Variables: 28
5.2.2 Dependent Variables: 28
5.2.3 Data Normalization: 29
5.3 Data Acquisition: 29
5.4 Data Sets: 34
5.4.1 Data sets of Bulb: 34
5.4.2 Data sets of Mobile Charger: 35
5.4.3 Data sets of Blower: 36
5.4.4 Data sets of Laptop Charger: 37
5.5 Attribute Acquisition Code: 38

5.5.1 Results of Attribute Acquisition: 41
5.5.1.1 Blower: 41
5.5.1.2 Bulb: 41
5.5.1.3 Laptop Charger: 42
5.5.1.4 Mobile Charger: 42
5.6 Device Detection Code: 43
5.6.1 Code Explanation: 46
5.6.1.1 dataset.shape(): 46
5.6.1.2 dataset.head(): 46
5.6.1.3 dataset.describe(): 46
5.6.1.4 dataset.groupby('Class').size(): 47
5.6.1.5 scatter_matrix(X): 47
5.6.1.6 confusion_matrix(y_test, y_pred): 47
5.6.2 Results: 47
5.7 Graphic User Interface: 49
5.7.1 Graphic User Interface MATLAB code: 49
5.7.2 GUI Result: 53
Chapter 6: Conclusion and Future Scope 55
6.1 Conclusion: 55
6.2 Future Scope: 55
6.2.1 Domestic Market: 55
6.2.2 Energy Efficiency in Products: 55
6.2.3 Multipoint sensing: 56
6.2.4 Road towards Cost Management: 56
6.2.5 Enhancement of Relationship with Home automation startups: 56
Chapter 7: Bibliography 57

List of figures:

Figure 4-1: Raspberry Pi 16
Figure 4-2: Analog Discovery 2 16
Figure 4-3: NodeMcu 17
Figure 4-4: Step Down Circuit 17
Figure 4-5: Arduino UNO........ 18
Figure 4-6: Smart Sense Switch Board 19
Figure 4-7: Implementation Methodology 20
Figure 4-8: Raspberry Pi Pin Configuration 22
Figure 4-9: Remote Switching Methodology 23
Figure 4-10: Data Set Acquisition Setup 25
Figure 5-11: Electrical Signature for Bulb 34
Figure 5-12: CSV file for bulb 35
Figure 5-13: Electrical Signature of Mobile Charger 35
Figure 5-14: CSV for Mobile charger 36
Figure 5-15: Electrical Signature 36
Figure 5-16: CSV for Blower 37
Figure 5-17: Electrical Signature Laptop Charger 37
Figure 5-18: CSV for Laptop Charger 38
Figure 5-19: Attributes of Blower 41
Figure 5-20: Attributes of Bulb 41
Figure 5-21: Attributes of Blower 42
Figure 5-22: Attributes of Mobile Charger 42
Figure 5-23: Results of Device Detection (a) 47
Figure 5-24: results of Device Detection (b) 48
Figure 5-25: Results of Device Detection (c) 48
Figure 5-26: Graphic User Interface 53

Chapter 1: Introduction

Chapter 1: Introduction

1.1 Overview:

Machine Learning is a great field which is an important technique to make devices smart logically. The algorithms are first trained with the help of some pre-recorded data sets and then they are tested with some test set (real time). The accuracy of the algorithm is then recorded to check the output results of the trained algorithm with reference to the actual outputs. This amazing concept can be applied in almost every scenario where there exists a possibility of recording the relevant data sets and training the algorithm for the purposeful output.

1.2 Problem Statement:

In existing switch boards, appliances cannot communicate with the users and vice versa. There is no remote access to appliances without the presence of sensors. There is no source of information being provided about the state of appliances and there is no information provided about the energy consumed by appliances.

1.3 Approach:

An electrical circuit will be used to step down the voltage from 220V to the requirement. The step downed values will be fed to an A/D convertor, the digitalized values will be then fed to the algorithm running in a Python IDE in the form of a CSV file. However, the miniaturized device will be using Raspberry Pi (with an integrated version of Python IDE as its OS). User will be connected to the smart device over a wireless connection via Pi to remotely switch and monitor the electrical appliances connected to the smart device (S^3B).

1.4 Objectives:

The main objectives of our work are:

- To provide remote access to the appliances without connection of a (one-one) sensor.
- To track the power consumption of appliances connected to S^3B.
- To help the user to monitor the time of usage by the appliance and manage it to reduce the power consumption.
- To make our devices smart so that they can communicate with the user. A big NO-NO to the use of sensors for each device for acting smartly – it costs too much.

1.5 Deliverables:

Sr	Tasks	Deliverables
1	Literature Review	Literature Survey and Feasibility Analysis
2	Requirements Specification	Software Requirements Specification document (SRS)
3	Detailed Design	Software Design Specification document (SDS)
4	Implementation	Project demonstration
5	Testing	Evaluation plan and test document
6	Training	Deployment plan
7	Deployment	Complete application with necessary documentation

1.6 Overview of Document:

1.6.1 Purpose:

This document covers detailed review of all major steps involved in development of Smart Sense Switch Board (S3B). These all involved steps acted as guide to the development team and now shall provide insight to the reader that how prototype idea was formulated and then how hardware integration took place, how software was designed and finally tested.

1.6.2 Document Conventions:

1.6.2.1 Headings:

Headings are prioritized in a numbered fashion, the highest priority heading having a single digit and subsequent headings having more numbers, per their level.

All the main headings are titled as follows: single digit number followed by a dot and the name of the section (All bold Calibri Light, size 18, Centered).

All second level sub headings for every sub section have the same number as their respective main heading, followed by one dot and subsequent sub heading number followed by name of the sub section (All bold Calibri Light, size 16).

Further sub headings, i.e. level three and below, follow the same rules as above for numbering and naming, but different for font (All bold Calibri, size 14).

1.6.2.2 Figures:

All figures in this document have captions and are numbered. Context and flow diagrams are based on UML standards.

1.6.2.3 References:

All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document.

1.6.2.4 Links to web pages:

All links have been provided with underlined font, the title of the web page or e-book is written at the top of the link and the title may be searched on google to pinpoint to the exact address.

1.6.2.5 Basic text:

All other basic text appears in regular, size 12 Calibri Light. Every paragraph explains one type of idea.

1.7 Intended Audience and Reading Suggestions:

The intended audience for this Document include the project supervisor, “SMART SENSE SWITCH BOARD” syndicate, BETE 51, UG project evaluation team, and other stakeholders at EE Department, MCS.

For better understanding, the document is divided into chapters:

- In chapter 1 an Introduction to Document and System is provided.
- Chapter 2 covers the requirement specifications part and covers Functional, Non-Functional Parts Requirements, resources required, and constraints involved
- Chapter 3 covers the Design Specifications which provide an in-depth view of how the systems is developed and how the functionalities are distributed.
- Chapter 4 discusses the hardware implementation of S3B.
- Chapter 5 gives detail about software implementation.

Developers: (Project Group)

To be sure that they are developing the right project that fulfills the requirements provided in this document.

Testers: (Project Group, Supervisor)

To have an exact list of the features and functions that must respond according to requirements.

Users:

To get familiar with the idea of the project and how to use/respond in failure situations and suggest other features that would make it even more functional.

Project Supervisor: (Dr. Mir Yasir Umair)

This document will be used by the project supervisor to check and guide the group about the understanding and implementation of the requirements properly and completely during the development lifecycle.

Project Evaluators: (EE Dept. MCS)

To know the scope of the project and evaluate the project throughout the development for grading.

1.8 References

More about Subject Project can be retrieved from project development team.

Ahsan Baidar Bakht
BETE 51 MCS NUST
Group Leader

Email: ahsanakram95@gmail.com

Chapter 2: Literature Review

Chapter 2: Literature Review

2.1 Preamble:

In existing switch boards, appliances cannot communicate with the users and vice versa. There is no remote access to appliances without the presence of sensors. There is no source of information being provided about the state of appliances and there is no information provided about the energy consumed by appliances.

Smart Sense Switch Board will be able to identify the devices based on machine learning algorithms without the use of sensors. There were no publicly available data sets for electrical appliances. The data sets will then be used to train Machine learning algorithm to identify the electrical appliances connected to the board in real life scenario and monitor the power consumed by devices remotely.

In the past few years a lot of work has been done on home automation and internet of things but all that solutions are using sensors to identify and control the devices but, in our case, we are using Machine learning algorithms for detection and then a graphic user interface for remotely monitoring the device.

Following are the few excerpts from the research papers related to the project:

2.2 Demo abstract: PLAID:

A Public Dataset of High-Resolution Electrical Appliance Measurements for Load Identification Research - The goal of PLAID is to provide a public library for high-resolution appliance measurements that can be integrated into existing or novel appliance identification algorithms. PLAID currently contains measurements for more than 200 different appliance instances, representing 11 appliance classes, and totaling more than a thousand records.

2.3 Real-Time Recognition and Profiling of Appliances through a Single Electricity:

Sensor - Sensing, monitoring and actuating systems are expected to play a key role in reducing buildings overall energy consumption. Leveraging sensor systems to support energy efficiency in buildings poses novel research challenges in monitoring space usage, controlling devices, interfacing with smart energy meters and communicating with the energy grid. In the attempt of reducing electricity consumption in buildings, identifying individual sources of energy consumption is key to generate energy awareness and improve efficiency of available energy resources usage

2.4 Neural NILM:

Deep Neural Networks Applied to Energy Disaggregation – Energy disaggregation estimates appliance-by-appliance electricity consumption from a single meter that measures the whole home's electricity demand. Recently, deep neural networks have driven remarkable improvements in classification performance in neighboring machine learning fields such as image classification and automatic speech recognition. In this paper, we adapt three deep neural network architectures to energy disaggregation: 1) a form of recurrent neural network called 'long short-term memory' (LSTM); 2) denoising autoencoders; and 3) a network which regresses the start time, end time and average power demand of each appliance activation.

2.5 Conclusion:

Considering, the existing solutions and possible limitations, a solution is required that uses efficient techniques (such as machine learning) and is smart enough to identify the devices without the use of any sensors and can record the data sets of electrical devices which are not previously available and then train the device on these data sets for accurate detection of devices and a system to remotely monitor the power consumption of the device.

Chapter 3: Design Requirements

Chapter 3: Design Specification

3.1 Introduction

The system requirement and specification for smart sense switch board are covered in this chapter. This chapter is meant to outline the features and requirements of “S3B”, to serve as a guide to the concerned people on one hand and a software validation document for the prospective client/stakeholders on the other.

3.2 Overall Description:

The idea of this project is the realization of a smart device that can identify electrical devices based on Machine learning algorithms and to record the data sets electrical devices and then train the device on these data sets ton differentiate between the electrical signatures of selected devices and to remotely monitor the power consumption of the device connected to the S3B.

3.3 Product Features:

The key features of Smart Sense Switch board are as follows:

1. The data sets are recorded for electrical appliances that were not publicly available.
2. Use of Machine Learning Algorithms rather than sensors for the detection of appliance.
3. Remote Monitoring of electrical appliances.
4. A graphic interface for the users for easy handling and observing power consumption and device status.

3.4 Operating Environment:

The sub-sections below give a brief description of environment, hardware & software-based requirements for operation of “S3B”.

3.4.1 Hardware:

“S3B” shall operate, either directly or indirectly, with the following hardware:

Raspberry Pi: It shall act as mini-computer with all necessary software installed on it. It shall run the python IDE for machine learning algorithms.

Arduino: it is used for the function of power monitoring and sampling at the rate of 8khz. Arduino will be connected to the NodeMcu for wireless connection with the device.

Power Cable: it is used for transmission of electricity to the device.

Switch Board: switch board and socket is used for connection of electrical appliance in Smart Sense Switch Board.

NodeMcu: NodeMcu is used for remotely connecting to the device for monitoring purpose.

Step Down Circuit: A step down circuit is used to step down the main voltage source for obtaining the required parameters (Voltage and Current) which serve as input for detection of appliances.

Digilent Analog Discovery: Analog discovery is used in the process of recording data sets and distinguishing the electrical signatures of appliances.

3.4.2 Software:

- Linux: Raspbian
- IDE: Enthought Canopy
- IDE: Spyder
- IDE: MATLAB
- Waveforms
- Windows operating system
- Arduino

3.4.3 Design and Implementation Constraints:

- Internet Connection between the NodeMcu and graphic user interface is necessary for S3B to receive commands and send feedback accordingly.
- The device shall only work in a complete known voltage range.
- Step down Circuit is only designed to step down 220-240 Volts.
- S3B will only detect the devices for which it has been trained to work with and not all electrical appliances.

Chapter 4: Hardware Implementation

Chapter 4: Hardware Implementation

4.1 Introduction:

This chapter describes the hardware design of project ‘Smart Sense Switch Board’. The chapter is meant to detail the design of features and requirements of ‘Smart Sense Switch Board’, to serve as a guide to the users on one hand and a software validation document for the prospective client on the other. It also includes detailed descriptions, sequence diagrams and various flow charts.

4.2 Components Used:

Following are the components used and their description:

4.2.1 Raspberry Pi 3:

Raspberry Pi 3 is the device which will be used in this project. Following are the specifications of raspberry pi:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot

Figure 4-1: Raspberry Pi

4.2.2 Analog Discovery 2:

The Digilent Analog Discovery 2, developed in conjunction with Analog Devices, is a multifunction instrument that allows users to measure, visualize, generate, record and control mixed signal circuits of all kinds. The project uses the discovery for A/D and making the data sets in the form of a CSV file.

Figure 4-2: Analog Discovery 2

4.2.3 NodeMcu:

NodeMcu is an open source IOT platform. It provides wireless access to our device and connects communicates with the board through ThingSpeak which is used for online data transfer. It has a range of around 50 meters to connect with access point and then we can remotely monitor the power consumption through GUI.

Figure 4-3: NodeMcu

4.2.4 Step Down Circuit:

The step-down circuit is implemented by using resistors and capacitor to step down a input voltage of 220V to 10-20V. Three connections has been taken out from the circuit as Current, Voltage and Common to use as input for Analog discovery. Analog discovery will give output data sets. These values serve as input parameters for ML process.

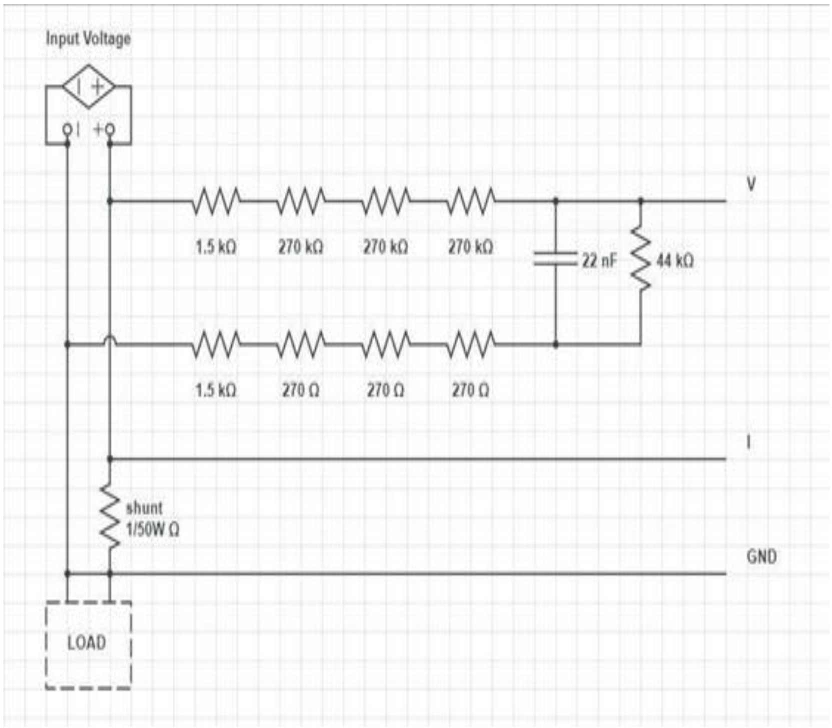


Figure 4-4: Step Down Circuit

4.2.5 Arduino UNO:

It is a microcontroller board which uses the ATmega328P .It has 14 digital I/O pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack and a reset button. It contains complete package to support the microcontroller; we can simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. We are using Arduino for the purpose of power monitoring and data transfer between user and our Smart Switch Board and communicating with NodeMcu for data transfer.

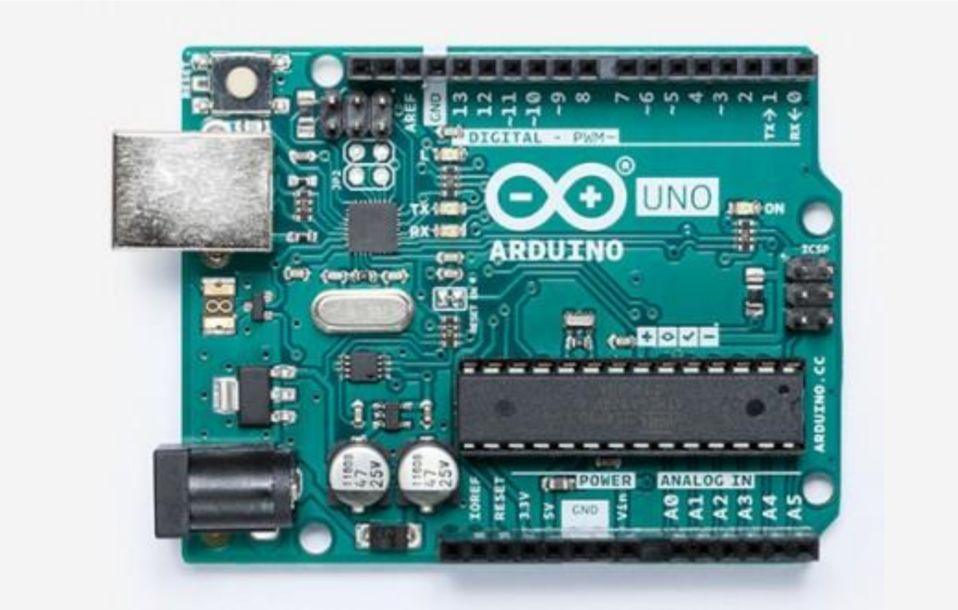


Figure 4-5: Arduino UNO

4.2.6 Switch Board:

A Switch Board is used is used as a connection point for electrical appliance to be detected and the power is supplied through this connection point.

4.2.7 LCD Display:

LCD display is used for displaying the results, the name of the appliance connected, and the power consumed by that electrical appliance:

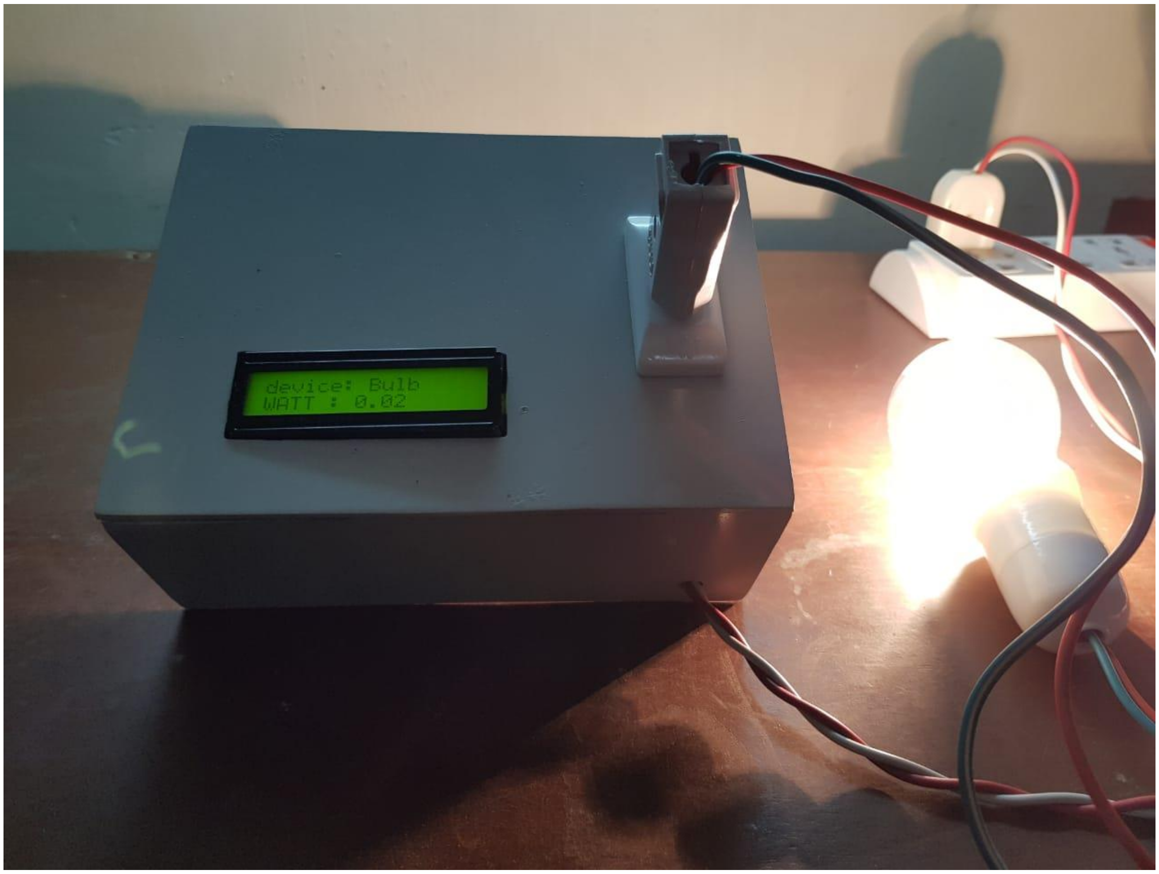


Figure 4-6: Smart Sense Switch Board

4.3 Implementation Methodology:

Following methodology is used while implementing and integrating the hardware of Smart Sense Switch board:

4.3.1 Energy Monitoring:

Following is the basic methodology which is employed while doing the energy monitoring. Appliance will be connected to a step-down circuit board to provide low voltage that will be fed into a high-speed A/D convertor. The digitalized sampled values at 8k Hertz will be then fed into the Digital Input pins of Raspberry P. An integrated version of Python IDE as an OS which will be running a machine learning code for smart detection of electrical appliances. User will relate to the complete scenario with a wireless connection via Pi for smart detection and monitoring of electrical appliances connected to S3B.

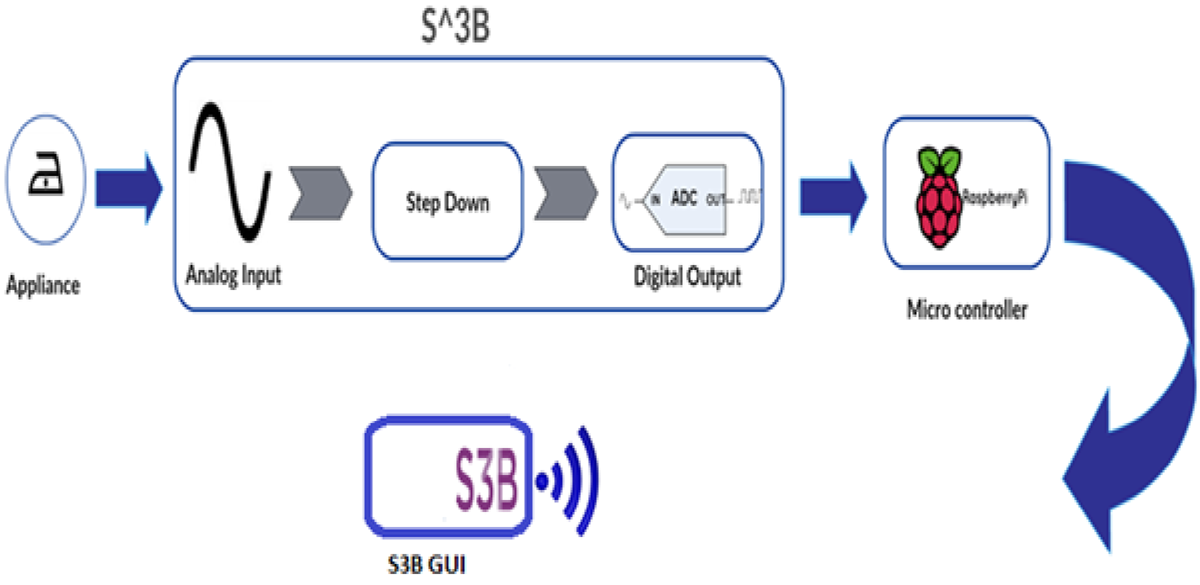


Figure 4-7: Implementation Methodology

4.3.2 How to Start with Raspberry Pi :

Raspberry Pi controller is a bit professional to use as far as the operating system installation is concerned, the configuration is concerned and thus the respective manipulation is concerned

- **Step 1**: The first step in this process is to format the SD card that is to be installed with the operating system of the raspberry. Firstly what you have to do is to format the SD card, have an SD card reader, install an SD card formatter and thus format the card completely.
- **Step 2:** The second step is to install the Zip File that has the Updated Raspbian Operating System and thus we have to write the Zip file in the SD card so that SD card act as an hard drive that has operating system Raspbian

Stretch Lite with Desktop from the Raspberry Pi website in it for the Raspberry Pi

- **Step 3**: Connect all the Auxiliary requirements with the cables i.e. HDMI to LCD, Mouse , keyboard, camera ,Ethernet and thus attach all the cables with the required ports in the raspberry pi depending on the particular requirements for which the ports are designated
- **Step 4**: When Raspbian begins to load a bunch of lines of code will appear. This will continue until the boot process has completed. Then, the Raspbian Home screen will appear. You will need to configure your Raspberry Pi system in order to add your location, date, and time. Click **Menu** in the upper left corner of the screen. Select **Preferences** in the dropdown menu. Select **Raspberry Pi Configuration** under Preferences. Click on Set Locale to set your location. Click on Set time zone to set your local time Click on Set Keyboard… to set your keyboard language.
- Raspberry Pi is thus connected with all the requirements. In order to change the password we can change the settings also. One important thing to know is that Username is by Default Raspberry and the password is Raspberry Pi in case the operator is unable to find it anywhere
- One thing for sure it must be kept in mind that to install anything or to delete anything rather any kind of information that is to be proceeded in the Raspberry pi needs a terminal window to be opened and all commands are prompted in that window same way as in we prompt them in command prompt of the windows. In order to install any library or software or install or delete something in the raspberry we are supposed to open the terminal window and perform the respective functions as scheduled.
- Another thing to be kept in mind is to update and upgrade the raspberry before the execution of anything, or installation of any library or deletion of any software it is the need of hour to first upgrade the Raspberry and then

update it and it was a major thing that helped saved many problems during the executions in raspberry

4.3.2.1 Raspberry Pi Pin Configuration:

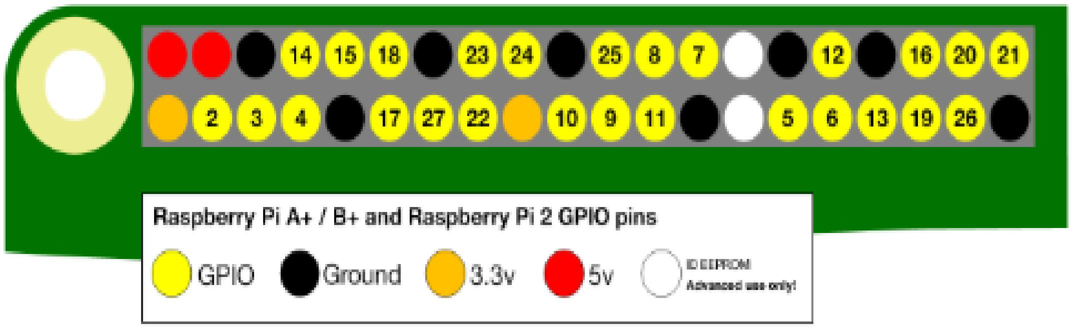


Figure 4-8: Raspberry Pi Pin Configuration

4.3.2.2 Voltages

2 pins are there for 5 Volt purpose and rest 2 pins for 3.3 volt are also present, also there are some ground pins that are for 0V purpose and in addition to them are some other 3.3 V pins also

4.3.2.3 Outputs

GPIO pin allotted as an output pin can be set to high 3.3 Voltage or ground 0 voltage.

4.3.2.4 Inputs

GPIO pin allotted as an input pin can be read as high (3V3) or low (0V). It is comfortable with the help of internal pull-up or pull-down resistors. GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be formulated using software.

4.4 Remote Monitoring and Switching:

For remote switching the NodeMcu is used, NodeMcu is connected wirelessly with the graphic user interface of S3B via internet. Thingspeak which is a free online cloud platform is used for integrating NodeMcu with the graphic user interface. The data is transferred from NodeMcu to the online cloud servers and then displayed on the GUI of S3B.

Following figure shows the methodology used for remote switching:

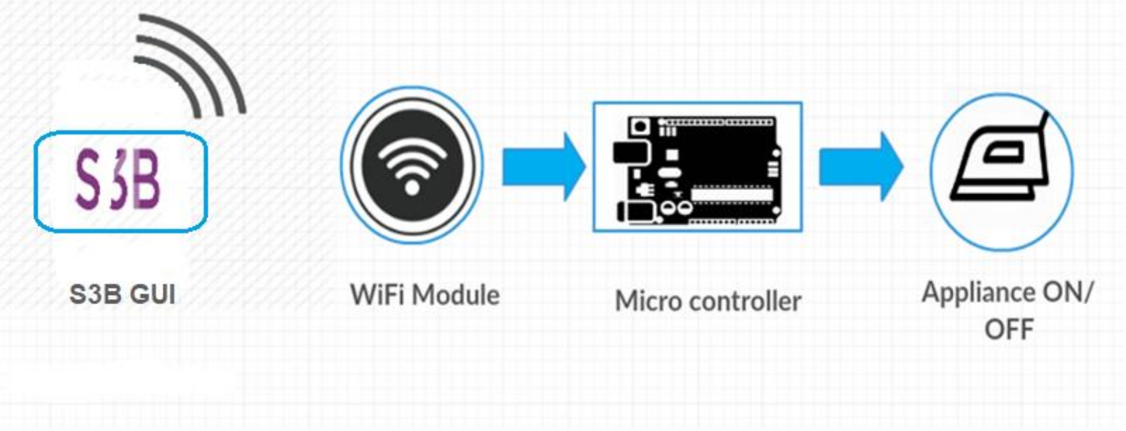


Figure 4-9: Remote Switching Methodology

4.4 Recording Data Sets:

The first and the most important job of the voltage and current values recording was the formulation of the data set that may help afterwards for the manipulation of the algorithms and thus the performing of machine learning on them that is basically for decision making afterwards.

Eventually, the need of the hour was to create a data set that may help to read in a good tabular form and thus the importance of the data is the performing of machine learning algorithms. To manage data accordingly so that we may have significant information of what the temperature and humidity is according to the time and there is a need to manage the data to perform specific algorithms accordingly

The most important need is to maintain a specific correlation between the data so that a proper decision can be processed. It is needed to have perfect information of what the internal and external temperature is according to the time. So, it was supposed to maintain the data in an EXCEL format to get data accordingly to the time and the values.

The Excel file serves best to provide information from a specific row and column and it is very easy to extract data from an excel file and we can extract information from a CSV file easily whenever and however required as far as coding is concerned.

Firstly, the voltage from main source is step down using the step-down circuit explained in Figure 4. The 220-240 volts were step down to 20-30 volts for data acquisition, but the power supplied for normal working of electrical appliance was not varied rather the circuit was implemented in parallel to the main voltage source.

After stepping down the voltage the values of current and voltage sent through probes into the channels of Analog discovery. Analog discovery converts the analog signals into digital form and makes a CSV file through these recorded values. The sampling frequency used was 8khz. After this step the attributes of data are acquired through attribute acquisition code and then this data is used for training the machine learning algorithms.

The features of the CSV file are a table that gives us data about unique electrical signature of each device and thus the most importantly we can extract the information through any row and column and thus can be manipulated in any easy and effective manner that may be necessary in some situations.

4.4.1 Recording of Values:

The basic purpose of the project was to record the values for selected devices. Thus it is the basic purpose to record the values continuously after a specific time so that a proper correlation can be maintained in this regards .As for machine learning it is the basic duty of the developer to provide device the capability of recording data that we may call training of the machine in this session the product records the physical data and thus manipulates it afterwards .Thus for the training of the data it is necessary to record information and for this purpose there needs to be a functionality in this regards that may provide the back hand facility to run the code multiple times. It isn’t a for or a while loop to be circulated what we need is to run the script again and again after a specific time mentioned so that training of the information can be prosecuted and what ultimately happens at the end is we get a proper data set on which further action is taken.

Probes are connected to the step- down circuit outputs and then the voltage and current values are then fed to the channels of analog discovery, these values are then transferred to the data acquisition code and then a CSV file is compiled for each data set.

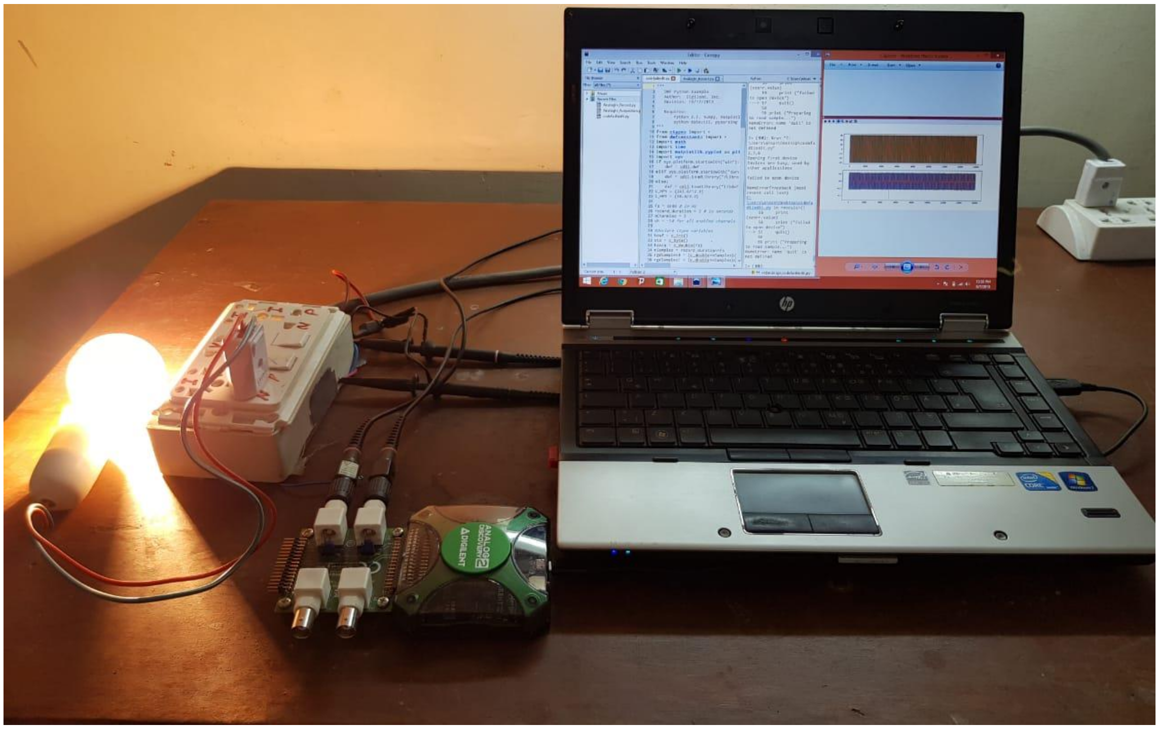


Figure 4-10: Data Set Acquisition Setup

Chapter 5: Software Implementation

Chapter 5: Software Implementation

5.1 Concept of Machine Learning:

Machine Learning is an amazing field which is helping in making the lives and gadgets smarter. It all starts with a data set which is required for training a ML algorithm and then improving the results for accuracy. Basically, what happens is that an algorithm predicts based on learnt behavior with help of a pre-defined data set. As the time passes the machine keeps on learning based on algorithm trained with data sets and accuracy is achieved.

5.1.1 Concept of Supervised Learning:

A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way. Supervised learning, in the context of artificial intelligence and machine learning, is a type of system in which both input and desired output data are provided. Input and output data are labeled for classification to provide a learning basis for future data processing. Thus, this is a way we can supervise a device in pertaining to the input output pairs of which a certain correlation is maintained to perform the specific task. The use of this correlation between input and respective output pair is the pure definition of the supervised learning. A great work is being done in the field of supervise learning in which an input output pair is correlated and thus the supervised learning is performed respectively.

5.2 Machine Learning in our Project:

In our project we have made data sets for electrical appliances including bulb, blower, laptop charger, mobile charger sampled at 8k Hertz in the form of a data set. Two variables in the data set include the current and the voltage values of above mentioned electrical appliances. Further the processed CSV data sets include four variables that include max, min, SD, average values. Electrical appliances connected to smart sense switch board will be detected using the ML algorithm.

By the learning of information, a correlation was supposed to be maintained by the user and thus it was supposed to learn that at what time of the day which device is connected to the smart switch board. The machine also must learn an exception routine in which specific appliance has consumed maximum power and as far as fluctuation of voltage is concerned the routine is learnt in this manner. The need of the project was to record the data sets of the electrical appliances and keep in view the specific electrical signature of appliance by which our algorithm makes the smart switch board learns.

5.2.1 Independent Variables:

The machine learning algorithms must show some specific variables as dependent variables and some as independent. That means there are some variables among which a correlation is maintained, and an output is generated in this manner. The parameters that are independent are made to be correlated with the dependent variables and thus the output is generated in a certain specific and effective form.

- Voltage
- Current
- Maximum Variable
- Minimum Variable
- Standard Deviation
- Average Value

5.2.2 Dependent Variables:

Using algorithms, the purpose is to predict certain electrical device connected in this manner. The need in our project is to predict a dependent variable that is the class of electrical appliance in our case and that is to be predicted. The correlation of dependent and independent variable is to be maintained and thus in this regards dependent variable is to be predicted in this manner what in our case is the class and name of the electrical appliance and that is further processed using the wireless connection to transmit the data on cloud. The dependent variable is:

Class (Name) of Electrical Appliance

5.2.3 Data Normalization:

The first and foremost challenge in machine learning is to have data set that is in normalized form and a specific correlation can be maintained in them. The voltage and current were recorded in the form of digitalized values, so it was difficult for the algorithm to identify a specific relation using high voltage and current values. SO those data sets were stepped down. This was a way to normalize the data recorded in the CSV to remove the high voltage and current values because they could not be fed to ADC and Pi.

5.3 Data Acquisition:

Data was acquired using Analog Discovery after the main voltage was step down to 10-20 volts. The Data Acquisition code written in Python was compiled using Enthought canopy. The data sets of the four selected devices e.g. bulb, mobile charger, blower and laptop charger were acquired using following python code:

```
from ctypes import *
from dwfconstants import *
import math
import time
import matplotlib.pyplot as plt
import sys


if sys.platform.startswith("win"):
   dwf = cdll.dwf
elif sys.platform.startswith("darwin"):
   dwf = cdll.LoadLibrary("/Library/Frameworks/dwf.framework/dwf")
```

```
else:
    dwf = cdll.LoadLibrary("libdwf.so")
V_MPY = (243.0/12.0)
I_MPY = (10.6/2.0)

fs = 8000 # in Hz
record_duration = 2 # in seconds
nChannles = 2
ch = -1# for all enabled channels

#declare ctype variables
hdwf = c_int()
sts = c_byte()
hzAcq = c_double(fs)
nSamples = record_duration*fs
rgdSamples0 = (c_double*nSamples)()
rgdSamples1 = (c_double*nSamples)()
cAvailable = c_int()
cLost = c_int()
cCorrupted = c_int()
fLost = 0
fCorrupted = 0

#print DWF version
version = create_string_buffer(16)
dwf.FDwfGetVersion(version)
print (version.value)
```

```
#open device

print ("Opening first device")

dwf.FDwfDeviceOpen(c_int(-1), byref(hdwf))


if hdwf.value == hdwfNone.value:

    szerr = create_string_buffer(512)

    dwf.FDwfGetLastErrorMsg(szerr)

    print (szerr.value)

    print ("failed to open device")

    quit()


print ("Preparing to read sample...")


#set up acquisition

dwf.FDwfAnalogInChannelEnableSet(hdwf, c_int(0), c_bool(True))

dwf.FDwfAnalogInChannelRangeSet(hdwf, c_int(0), c_double(20))


dwf.FDwfAnalogInChannelEnableSet(hdwf, c_int(1), c_bool(True))

dwf.FDwfAnalogInChannelRangeSet(hdwf, c_int(1), c_double(15))


dwf.FDwfAnalogInAcquisitionModeSet(hdwf, acqmodeRecord)

dwf.FDwfAnalogInFrequencySet(hdwf, hzAcq)

dwf.FDwfAnalogInRecordLengthSet(hdwf, c_double(nSamples/hzAcq.value))


#wait at least 2 seconds for the offset to stabilize

time.sleep(2)
```

```
#begin acquisition

dwf.FDwfAnalogInConfigure(hdwf, c_int(0), c_int(1))

print ("   waiting to finish")


cSamples = 0


while cSamples < nSamples:

   dwf.FDwfAnalogInStatus(hdwf, c_int(1), byref(sts))

   if cSamples == 0 and (sts == DwfStateConfig or sts == DwfStatePrefill or sts == DwfStateArmed) :

      # Acquisition not yet started.

      continue


   dwf.FDwfAnalogInStatusRecord(hdwf, byref(cAvailable), byref(cLost), byref(cCorrupted))


   cSamples += cLost.value


   if cLost.value :

      fLost = 1

   if cCorrupted.value :

      fCorrupted = 1


   if cAvailable.value==0 :

      continue


   if cSamples+cAvailable.value > nSamples :
        cAvailable = c_int(nSamples-cSamples)


    # get samples
    dwf.FDwfAnalogInStatusData(hdwf, c_int(0), byref(rgdSamples0, 8*cSamples), cAvailable) # copy channel 0 samples
    dwf.FDwfAnalogInStatusData(hdwf, c_int(1), byref(rgdSamples1, 8*cSamples), cAvailable) # copy channel 1 samples
    cSamples += cAvailable.value



print ("Recording finished")
dwf.FDwfDeviceCloseAll()


if fLost:
    print ("Samples were lost! Reduce frequency")
if cCorrupted:
    print ("Samples could be corrupted! Reduce frequency")


f = open("F.csv", "w")
#for v in rgdSamples0:
#    f.write("%s\n" % v)
#f.close()
rgpy0=[0.0]*len(rgdSamples0)
rgpy1=[0.0]*len(rgdSamples1)
for i in range(0,len(rgpy0)):
    rgpy0[i]=rgdSamples0[i]
    rgpy1[i]=rgdSamples1[i]
for v0,v1 in zip(rgdSamples0, rgdSamples1):
  f.write("%s,%s\n" % (v1,v0))
f.close()
 plt.subplot(211)
plt.plot(rgpy0)
plt.subplot(212)
plt.show()
```

5.4 Data Sets:

After the code is compiled using Enthought Canopy the CSV file is obtained for data sets of each electrical appliance. The data values are voltage and current values of the selected electrical appliances. These values are then combined in a Excel sheet for further processing. We recorded a total of 1600 data sets (400 for each appliance) for training our ML algorithm.

We enabled two channels of Analog Discovery for data set acquisition, Channel 1 was used for recording current values and channel two for voltage values.

The data sets for selected appliances are shown below:

5.4.1 Data sets of Bulb:

Electrical Signature for bulb is as follows:

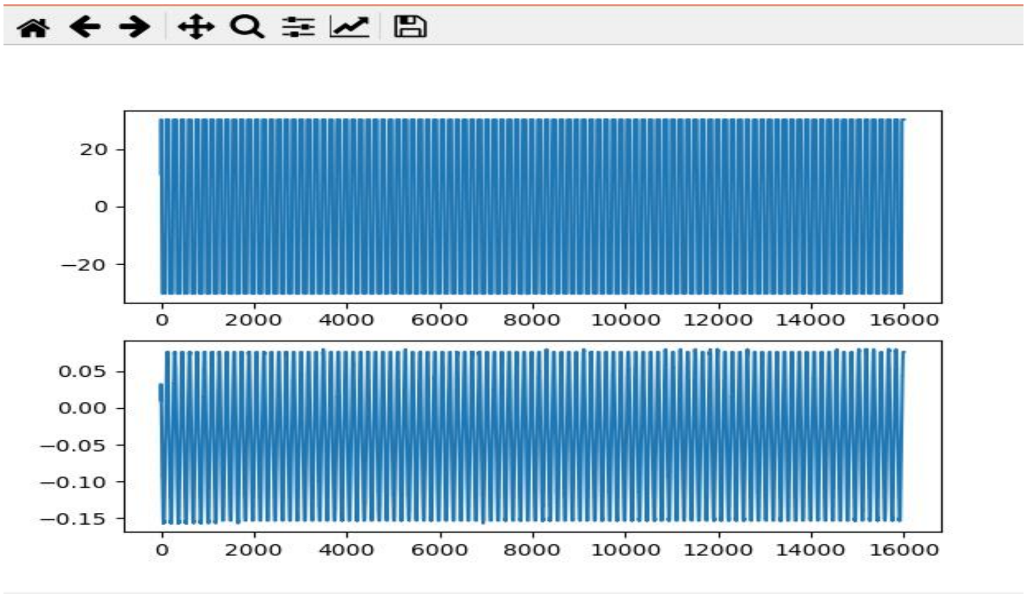
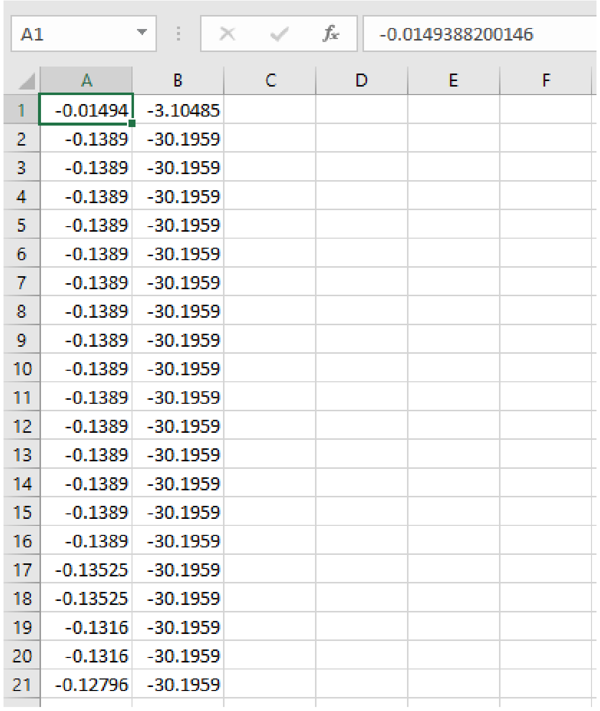


Figure 5-11: Electrical Signature for Bulb

The CSV file for one of the data sets of Electric Bulb are as follows:



A1 | -0.0149388200146

	A	B	C	D	E	F
1	-0.01494	-3.10485				
2	-0.1389	-30.1959				
3	-0.1389	-30.1959				
4	-0.1389	-30.1959				
5	-0.1389	-30.1959				
6	-0.1389	-30.1959				
7	-0.1389	-30.1959				
8	-0.1389	-30.1959				
9	-0.1389	-30.1959				
10	-0.1389	-30.1959				
11	-0.1389	-30.1959				
12	-0.1389	-30.1959				
13	-0.1389	-30.1959				
14	-0.1389	-30.1959				
15	-0.1389	-30.1959				
16	-0.1389	-30.1959				
17	-0.13525	-30.1959				
18	-0.13525	-30.1959				
19	-0.1316	-30.1959				
20	-0.1316	-30.1959				
21	-0.12796	-30.1959				

Figure 5-12: CSV file for bulb

5.4.2 Data sets of Mobile Charger:

Electrical Signature acquired for mobile charger is as follows:

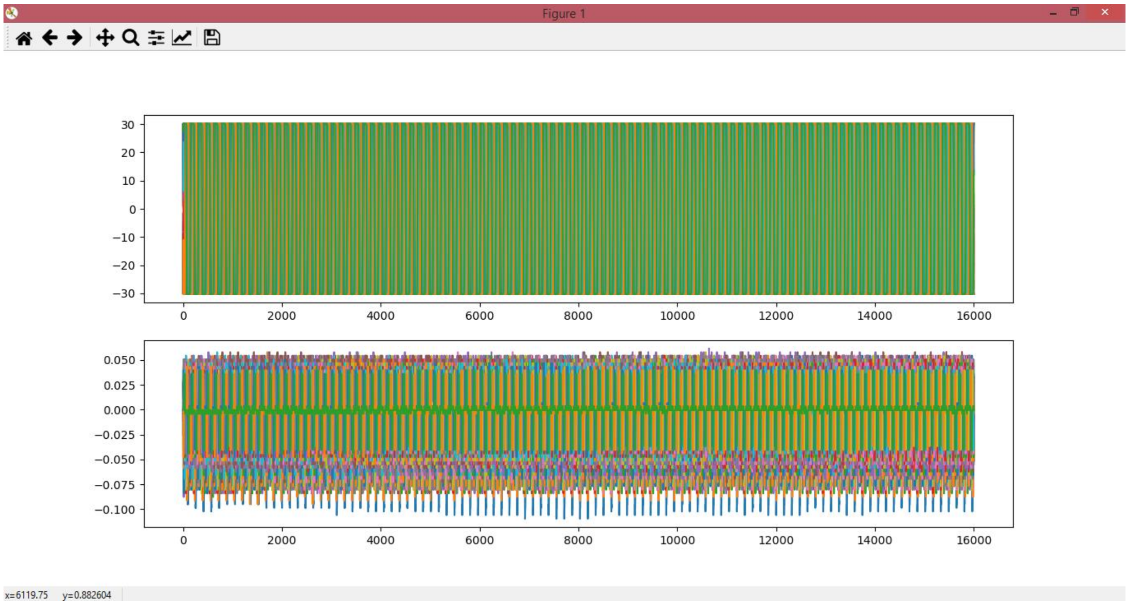
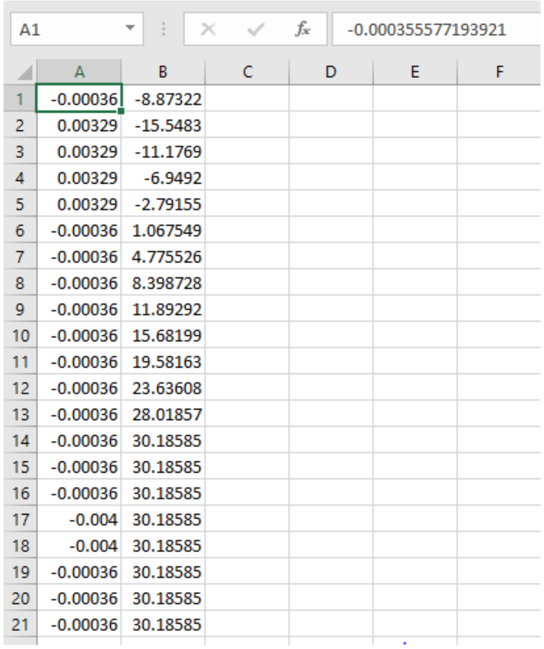


Figure 5-13: Electrical Signature of Mobile Charger

The CSV file for Mobile Charger:



A1 | -0.000355577193921

	A	B	C	D	E	F
1	-0.00036	-8.87322				
2	0.00329	-15.5483				
3	0.00329	-11.1769				
4	0.00329	-6.9492				
5	0.00329	-2.79155				
6	-0.00036	1.067549				
7	-0.00036	4.775526				
8	-0.00036	8.398728				
9	-0.00036	11.89292				
10	-0.00036	15.68199				
11	-0.00036	19.58163				
12	-0.00036	23.63608				
13	-0.00036	28.01857				
14	-0.00036	30.18585				
15	-0.00036	30.18585				
16	-0.00036	30.18585				
17	-0.004	30.18585				
18	-0.004	30.18585				
19	-0.00036	30.18585				
20	-0.00036	30.18585				
21	-0.00036	30.18585				

Figure 5-14: CSV for Mobile charger

5.4.3 Data sets of Blower:

Electrical acquired for blower is as follows:

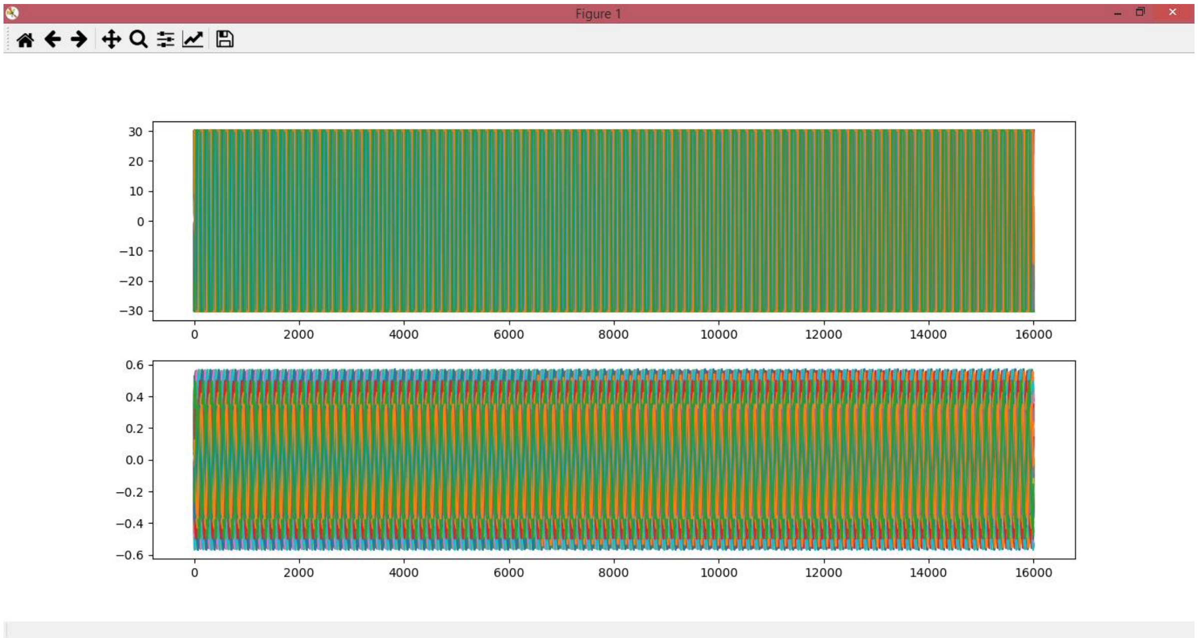
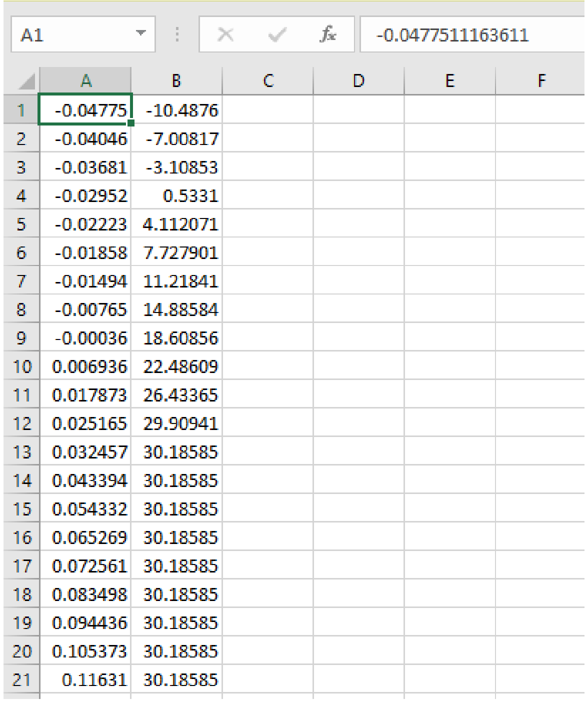


Figure 5-15: Electrical Signature

The CSV file for Blower is as follows:



-0.0477511163611

	A	B	C	D	E	F
1	-0.04775	-10.4876				
2	-0.04046	-7.00817				
3	-0.03681	-3.10853				
4	-0.02952	0.5331				
5	-0.02223	4.112071				
6	-0.01858	7.727901				
7	-0.01494	11.21841				
8	-0.00765	14.88584				
9	-0.00036	18.60856				
10	0.006936	22.48609				
11	0.017873	26.43365				
12	0.025165	29.90941				
13	0.032457	30.18585				
14	0.043394	30.18585				
15	0.054332	30.18585				
16	0.065269	30.18585				
17	0.072561	30.18585				
18	0.083498	30.18585				
19	0.094436	30.18585				
20	0.105373	30.18585				
21	0.11631	30.18585				

Figure 5-16: CSV for Blower

5.4.4 Data sets of Laptop Charger:

Electrical Signature acquired for laptop charger:

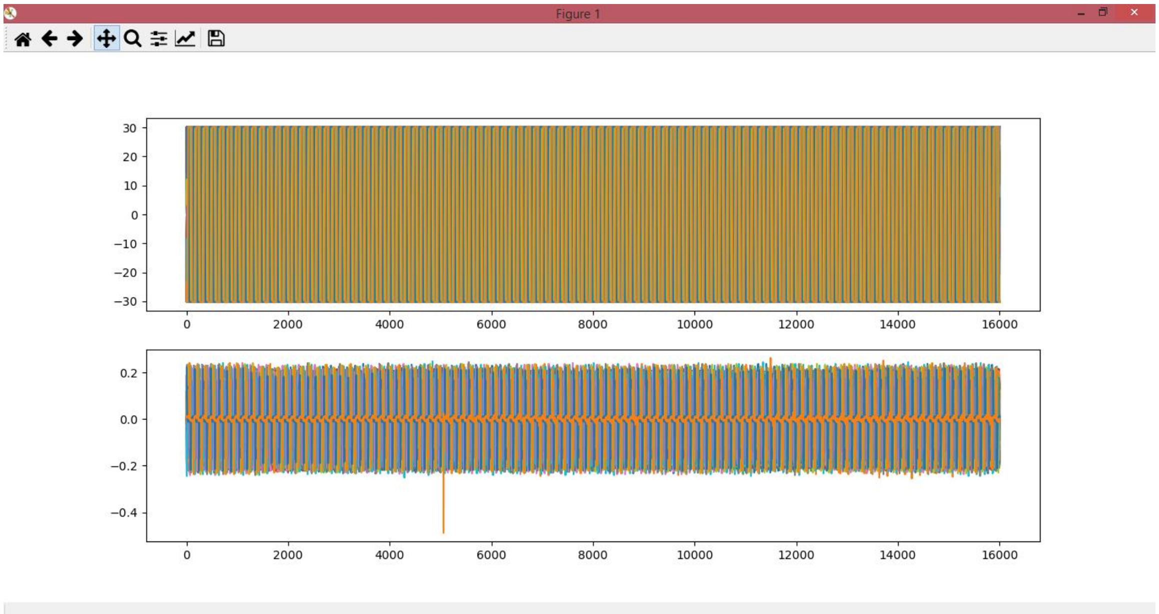
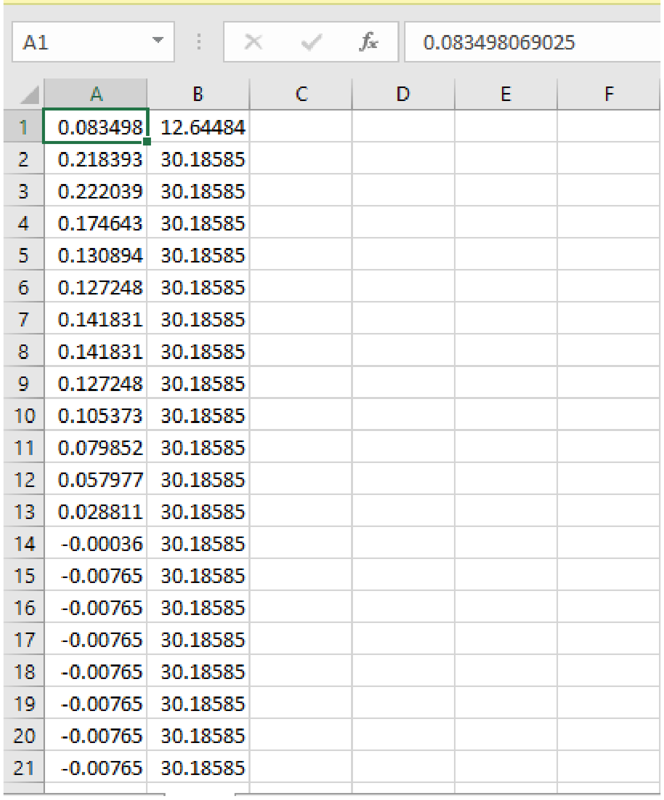


Figure 5-17: Electrical Signature Laptop Charger

The CSV file of data set of Laptop Charger:



0.083498069025

	A	B	C	D	E	F
1	0.083498	12.64484				
2	0.218393	30.18585				
3	0.222039	30.18585				
4	0.174643	30.18585				
5	0.130894	30.18585				
6	0.127248	30.18585				
7	0.141831	30.18585				
8	0.141831	30.18585				
9	0.127248	30.18585				
10	0.105373	30.18585				
11	0.079852	30.18585				
12	0.057977	30.18585				
13	0.028811	30.18585				
14	-0.00036	30.18585				
15	-0.00765	30.18585				
16	-0.00765	30.18585				
17	-0.00765	30.18585				
18	-0.00765	30.18585				
19	-0.00765	30.18585				
20	-0.00765	30.18585				
21	-0.00765	30.18585				

Figure 5-18: CSV for Laptop Charger

5.5 Attribute Acquisition Code:

After the CSV files are made for each data set of the four appliances the next step is to acquire the attributes of the data set based on which detection is possible. In our case four attributes are selected and acquired from each CSV file. The four attributes selected in our case are:

1. Maximum Value
2. Minimum Value
3. Average Value
4. Standard Deviation.

The CSV file is imported and after that the four attributes are acquired using python code.

The Python code for data acquisition compiled using Spyder IDE is as follows:

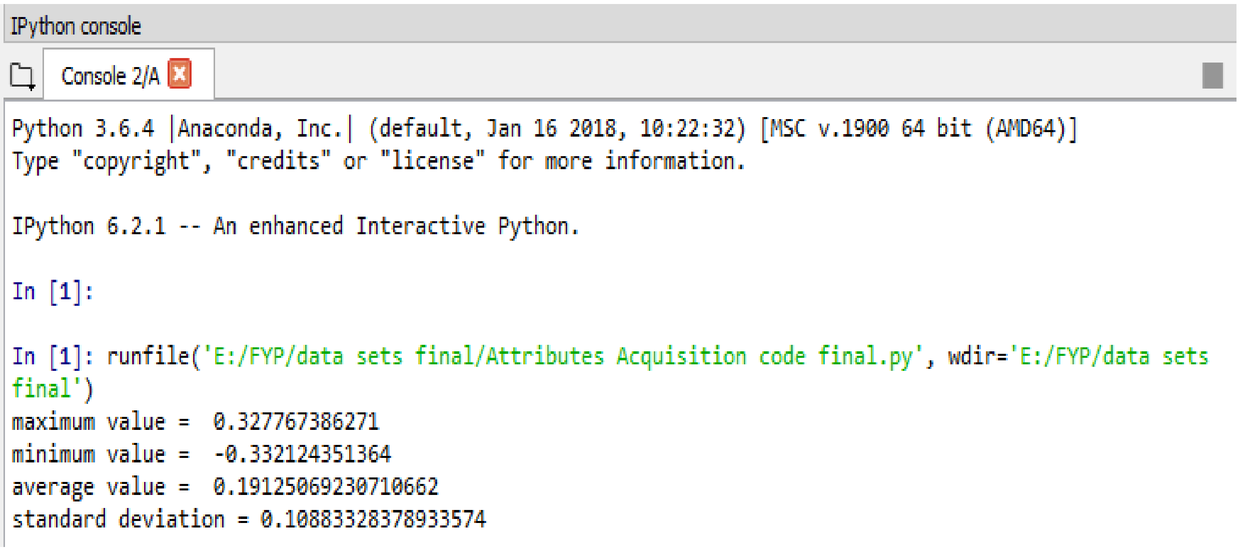
```
import csv
import numpy

with open('4.csv', 'r') as csv_file:
reader = csv.reader(csv_file, delimiter=",")

rownum = 0
a = []
for row in reader:
a.append (row)
rownum += 1


# making list from CSV coloumn

arr=[]
first_values_as_ints = [float(line[0]) for line in a]
for x in first_values_as_ints:
arr.append(x)

# maximum and minimum values

mx=max(arr)
print('maximum value = ', mx)
```

```
mi=min(arr)

print('minimum value = ', mi)


# length of list

l=len(arr)


#total sum and average


i=0

total=0.0

for i in range (0,l):

if arr[i]<0:

arr[i]=(arr[i])*(-1)

total=total+arr[i]

i+=1

avg=total/l

print('average value = ',avg)


# standard deviation


sd=numpy.std(arr)

print('standard deviation =', sd)
```

5.5.1 Results of Attribute Acquisition:

Results of attributes of four devices are shown below:

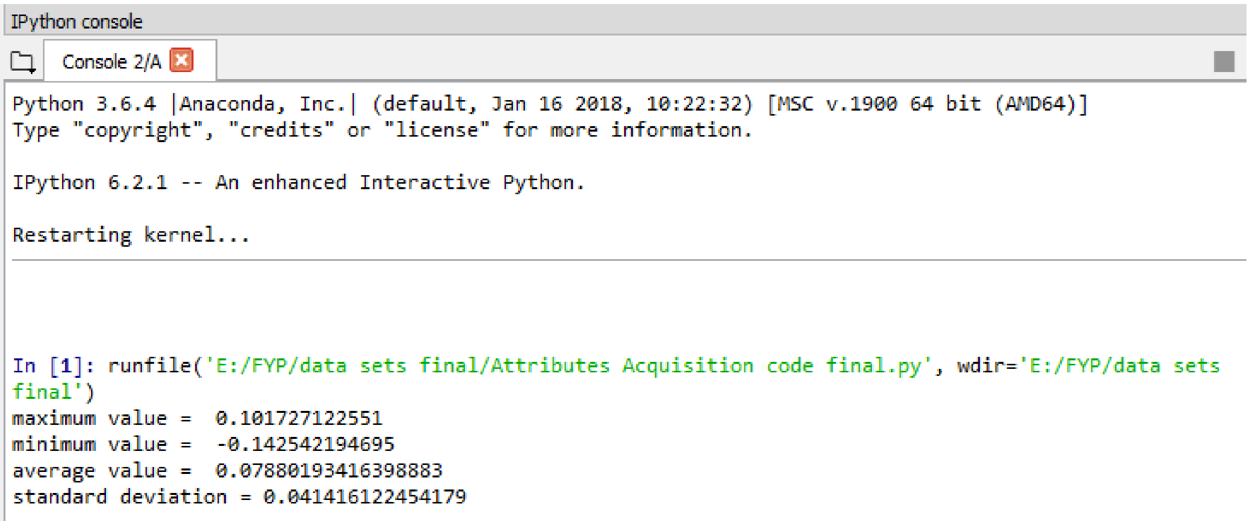
5.5.1.1 Blower:



```
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

In [1]:

In [1]: runfile('E:/FYP/data sets final/Attributes Acquisition code final.py', wdir='E:/FYP/data sets
final')
maximum value =  0.327767386271
minimum value =  -0.332124351364
average value =  0.19125069230710662
standard deviation = 0.10883328378933574
```

Figure 5-19: Attributes of Blower

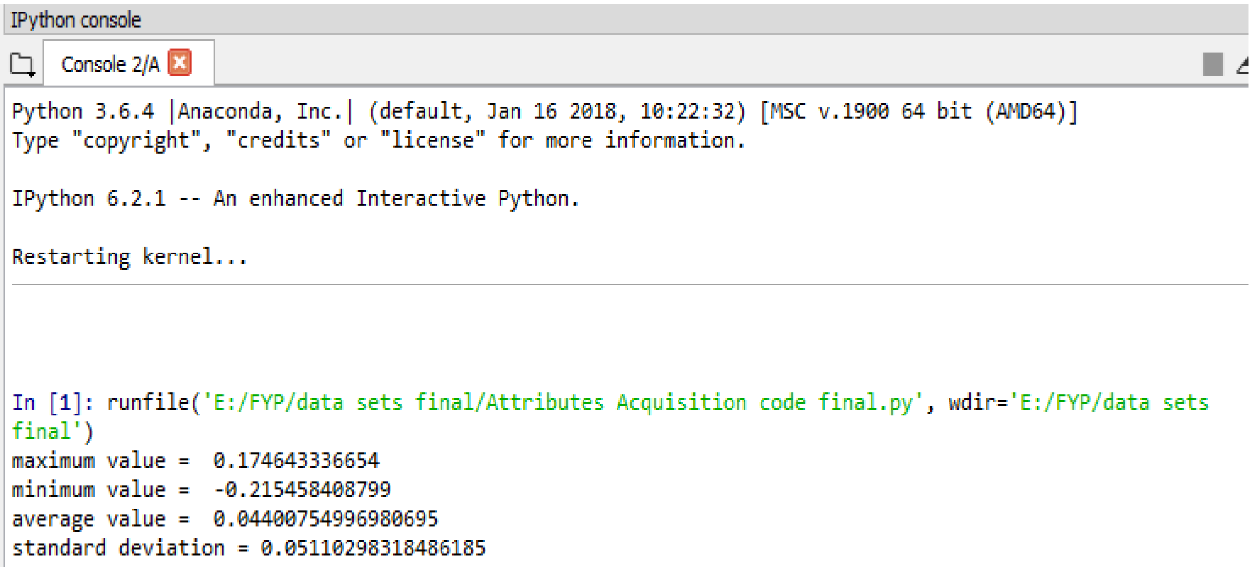
5.5.1.2 Bulb:



```
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('E:/FYP/data sets final/Attributes Acquisition code final.py', wdir='E:/FYP/data sets
final')
maximum value =  0.101727122551
minimum value =  -0.142542194695
average value =  0.07880193416398883
standard deviation = 0.041416122454179
```

Figure 5-20: Attributes of Bulb

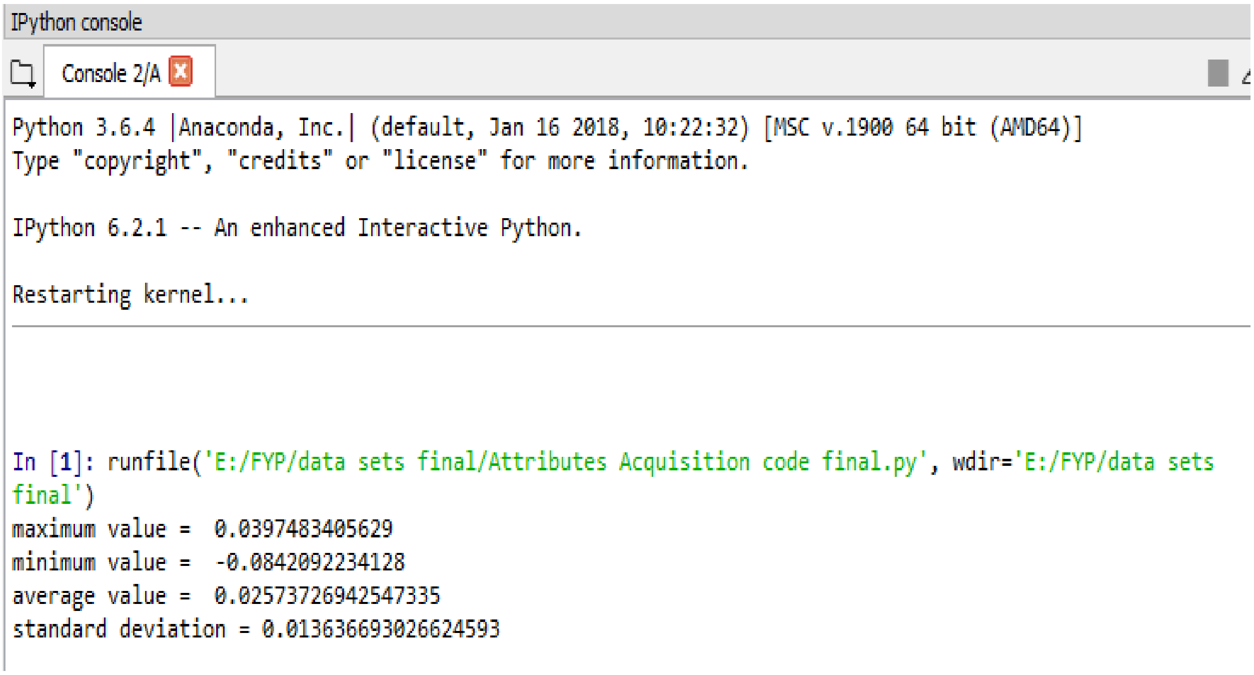
5.5.1.3 Laptop Charger:



```
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('E:/FYP/data sets final/Attributes Acquisition code final.py', wdir='E:/FYP/data sets
final')
maximum value =  0.174643336654
minimum value =  -0.215458408799
average value =  0.04400754996980695
standard deviation = 0.05110298318486185
```

Figure 5-21: Attributes of Blower

5.5.1.4 Mobile Charger:



```
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('E:/FYP/data sets final/Attributes Acquisition code final.py', wdir='E:/FYP/data sets
final')
maximum value =  0.0397483405629
minimum value =  -0.0842092234128
average value =  0.02573726942547335
standard deviation = 0.013636693026624593
```

Figure 5-22: Attributes of Mobile Charger

5.6 Device Detection Code:

The final step in our project is the detection of electrical appliance using ML algorithm. The device is trained on acquired data sets and then differentiated and detected on the basis of their attributes.

The code for device detection is as follows:

```
import pandas as pd
from pandas.plotting import scatter_matrix
import numpy as np
import matplotlib.pyplot as plt


def predict(num):
if num==0:
print('Blower')
elif num==1 :
print('bulb')
elif num==2:
print('laptop charger')
elif num==3:
print('mobile charger')


dataset = pd.read_csv("E:/FYP/data sets final/Final_datasets.csv")


print(dataset.shape)
print(dataset.head(10))
print(dataset.describe())
print(dataset.groupby('Class').size())
dataset.hist()
plt.show()
```

```
X = dataset.drop('Class', axis=1)
y = dataset['Class']

scatter_matrix(X)
plt.show()

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.60)

from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

import csv
with open('F.csv', 'r') as ifile:
reader = csv.reader(ifile, delimiter=",")

rownum = 0
a = []
for row in reader:
a.append (row)
```

```
rownum += 1

# making list from CSV coloumn
arr=[]
first_values_as_ints = [float(line[0]) for line in a]
for x in first_values_as_ints:
if x<1:
x=x*(-1)
arr.append(x)

# maximum and minimum values
mx=max(arr)
mi=min(arr)
l=len(arr)

#total sum and average
i=0
total=0.0
for i in range (0,l):
if arr[i]<0:
arr[i]=(arr[i])*(-1)
total=total+arr[i]
i+=1

avg=total//l
sd=np.std(arr)
d1=mx
```

```
d2=mi
d3=avg
d4=sd
data = [[d1, d2, d3, d4]]
print('The predicted device for the given CSV is ')
predict(classifier.predict(data))
```

5.6.1 Code Explanation:

Predict(num) is a function which is used to show the results of the value of Decision tree classifier. Here we have four devices each with a corresponding target value as:

Blower=0

Bulb=1

Laptop Charger=2

Mobile Charger=3

5.6.1.1 dataset.shape():

It generates the total number of rows and columns i.e. the number of attributes and the number of samples in the dataset file

5.6.1.2 dataset.head():

It generates some chunks of rows from first and last of the data set to show how dataset looks like.

5.6.1.3 dataset.describe():

It generates descriptive statistics that summarize the central tendency, dispersion and shape of a dataset’s distribution, excluding NaN value the result’s index will include count, mean, std, min, max as well as lower, 50 and upper percentiles. By default, the lower percentile is 25 and the upper percentile is 75. The 50 percentile is the same as the median.

5.6.1.4 dataset.groupby('Class').size():

This command generates the number of samples taken for each type of class.

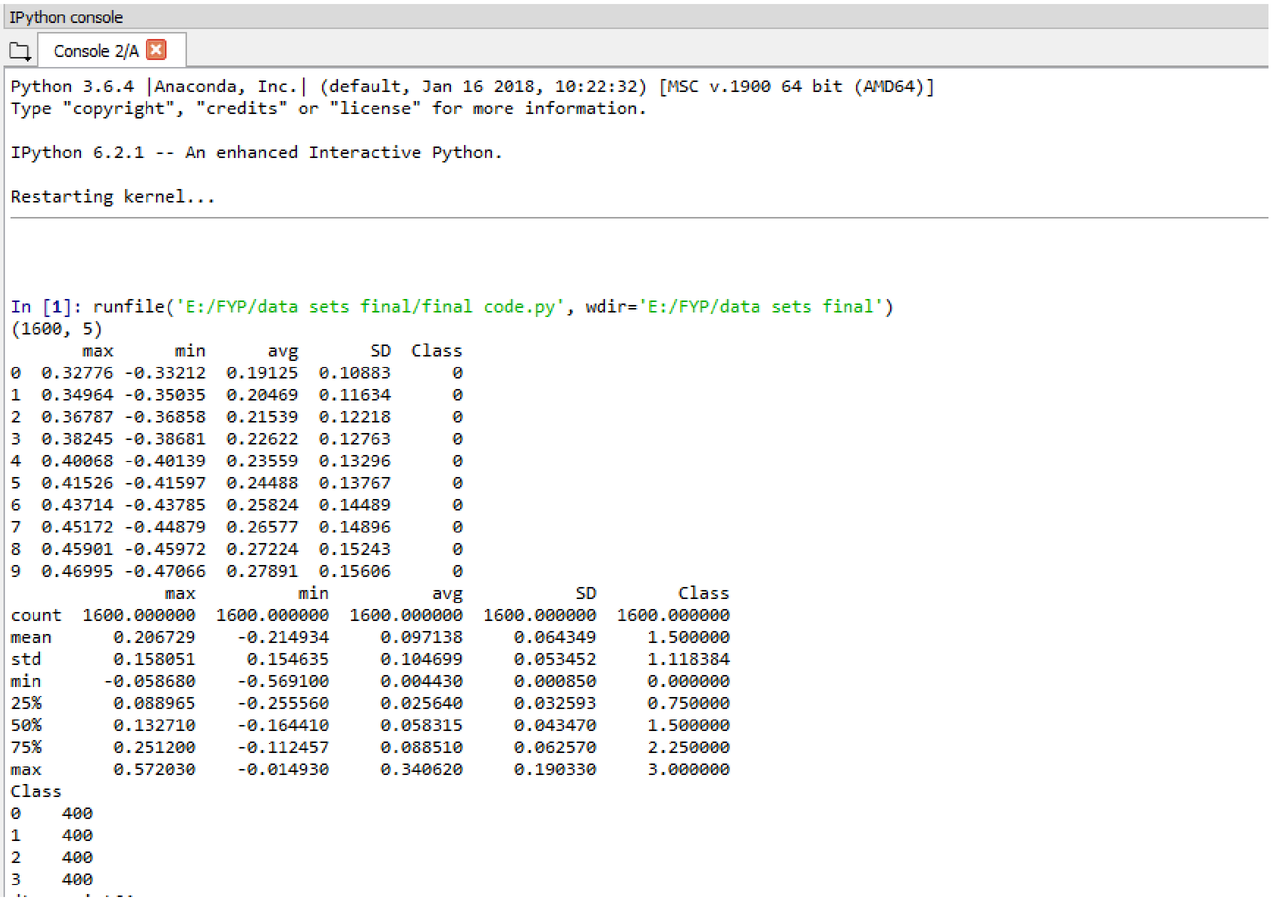
5.6.1.5 scatter_matrix(X):

This command is used to check the correlation between the variables and it also determine whether the correlation is positive or negative.

5.6.1.6 confusion_matrix(y_test, y_pred):

It is used to determine how many times the classifier predicted wrongly for test dataset.

5.6.2 Results:



```
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('E:/FYP/data sets final/final code.py', wdir='E:/FYP/data sets final')
(1600, 5)
       max      min      avg       SD  Class
0  0.32776 -0.33212  0.19125  0.10883      0
1  0.34964 -0.35035  0.20469  0.11634      0
2  0.36787 -0.36858  0.21539  0.12218      0
3  0.38245 -0.38681  0.22622  0.12763      0
4  0.40068 -0.40139  0.23559  0.13296      0
5  0.41526 -0.41597  0.24488  0.13767      0
6  0.43714 -0.43785  0.25824  0.14489      0
7  0.45172 -0.44879  0.26577  0.14896      0
8  0.45901 -0.45972  0.27224  0.15243      0
9  0.46995 -0.47066  0.27891  0.15606      0
               max          min          avg           SD        Class
count  1600.000000  1600.000000  1600.000000  1600.000000  1600.000000
mean      0.206729    -0.214934     0.097138     0.064349     1.500000
std       0.158051     0.154635     0.104699     0.053452     1.118384
min      -0.058680    -0.569100     0.004430     0.000850     0.000000
25%       0.088965    -0.255560     0.025640     0.032593     0.750000
50%       0.132710    -0.164410     0.058315     0.043470     1.500000
75%       0.251200    -0.112457     0.088510     0.062570     2.250000
max       0.572030    -0.014930     0.340620     0.190330     3.000000
Class
0    400
1    400
2    400
3    400
```

Figure 5-23: Results of Device Detection (a)

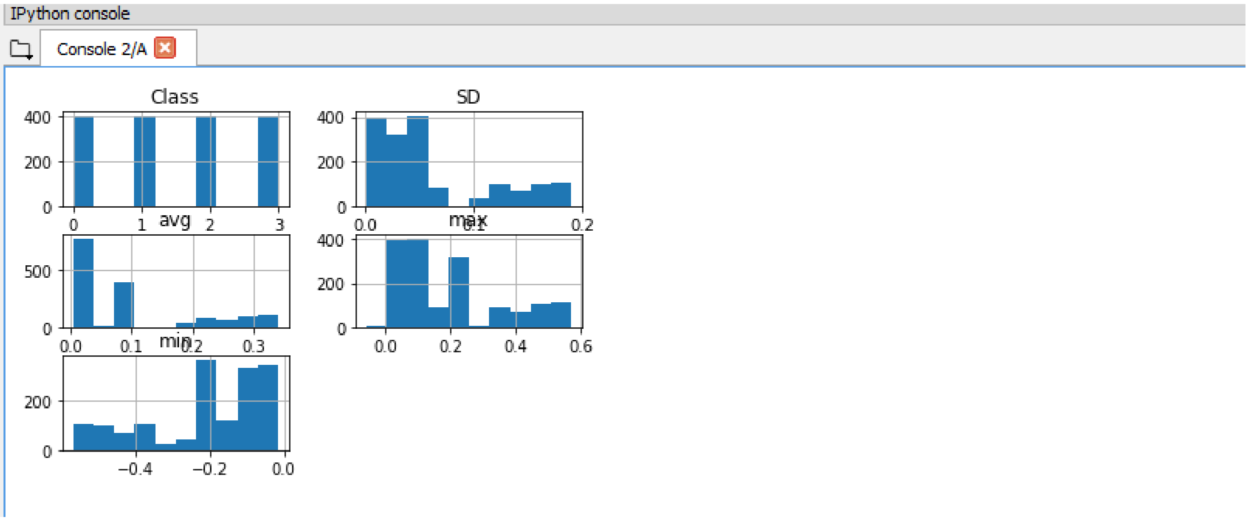
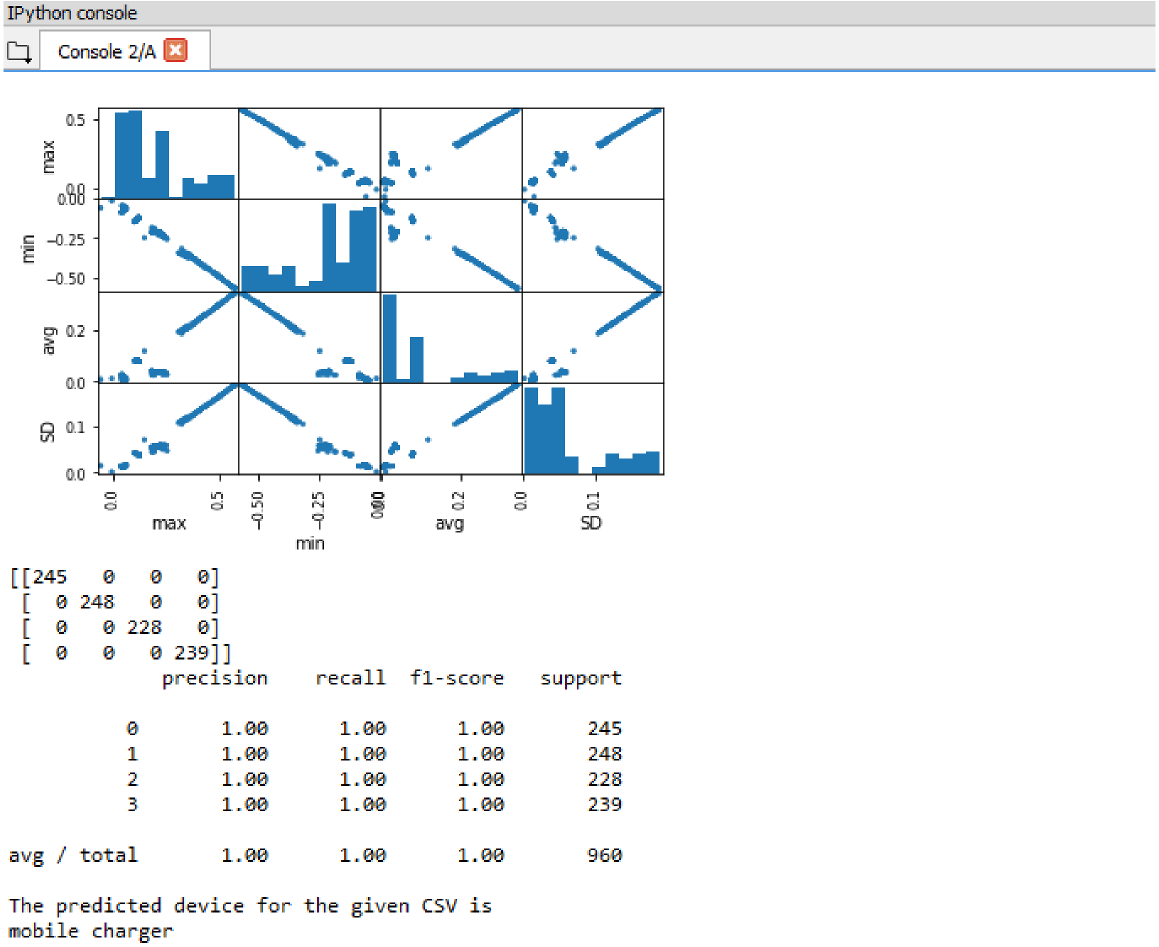


Figure 5-24: results of Device Detection (b)



```
[[245   0   0   0]
 [  0 248   0   0]
 [  0   0 228   0]
 [  0   0   0 239]]
             precision    recall  f1-score   support

          0       1.00      1.00      1.00       245
          1       1.00      1.00      1.00       248
          2       1.00      1.00      1.00       228
          3       1.00      1.00      1.00       239

avg / total       1.00      1.00      1.00       960

The predicted device for the given CSV is
mobile charger
```

Figure 5-25: Results of Device Detection (c)

5.7 Graphic User Interface:

A graphic user interface is designed on MATLAB for providing wireless access to the Smart Switch Board. This GUI provides remote monitoring of the device status and the power consumed by the electrical appliance connected to the board. The NodeMcu connects with Thingspeak through internet provided by user, Thingspeak is and online free cloud platform used for the wireless communication between the GUI and the NodeMcu.

5.7.1 Graphic User Interface MATLAB code:

The MATLAB code for graphic user interface is as follows:

```
function gui12

f = figure('Visible', 'off','color',[0.4 0.4 0.4],'Position',[360,500,1050,600]);


hsttext = uicontrol('Style','text','BackgroundColor','[0.4 0.4
0.4]','ForegroundColor','green','FontWeight','bold','FontSize',25,'Position',[300,400,450,
130],'String','SMART SENSE SWITCH BOARD');


huitext = uicontrol('Style','text','BackgroundColor','[0.4 0.4 0.4]','String','Please Push
Following Buttons to Turn on Real Time Monitering
','ForegroundColor','red','FontSize',15,'FontWeight','bold','Position',[300,380,450,50]);

hsttext11 = uicontrol('Parent',f,'Style','push','BackgroundColor',[0 0.5 1],
'Position',[210,205,200,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');

hsttext13 = uicontrol('Parent',f,'Style','push','BackgroundColor',[0 0.5 1],
'Position',[210,150,200,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');


hsttext15 = uicontrol('Parent',f,'Style','push','BackgroundColor',[0 0.5 1],
'Position',[660,205,200,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');

hsttext16 = uicontrol('Parent',f,'Style','push','BackgroundColor',[0 0.5 1],
'Position',[660,150,200,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');


hsttext21 = uicontrol('Parent',f,'Style','edit','BackgroundColor',[0 0.5 1],
'Position',[420,205,50,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');

hsttext23 = uicontrol('Parent',f,'Style','edit','BackgroundColor',[0 0.5 1],
'Position',[420,150,50,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');
```

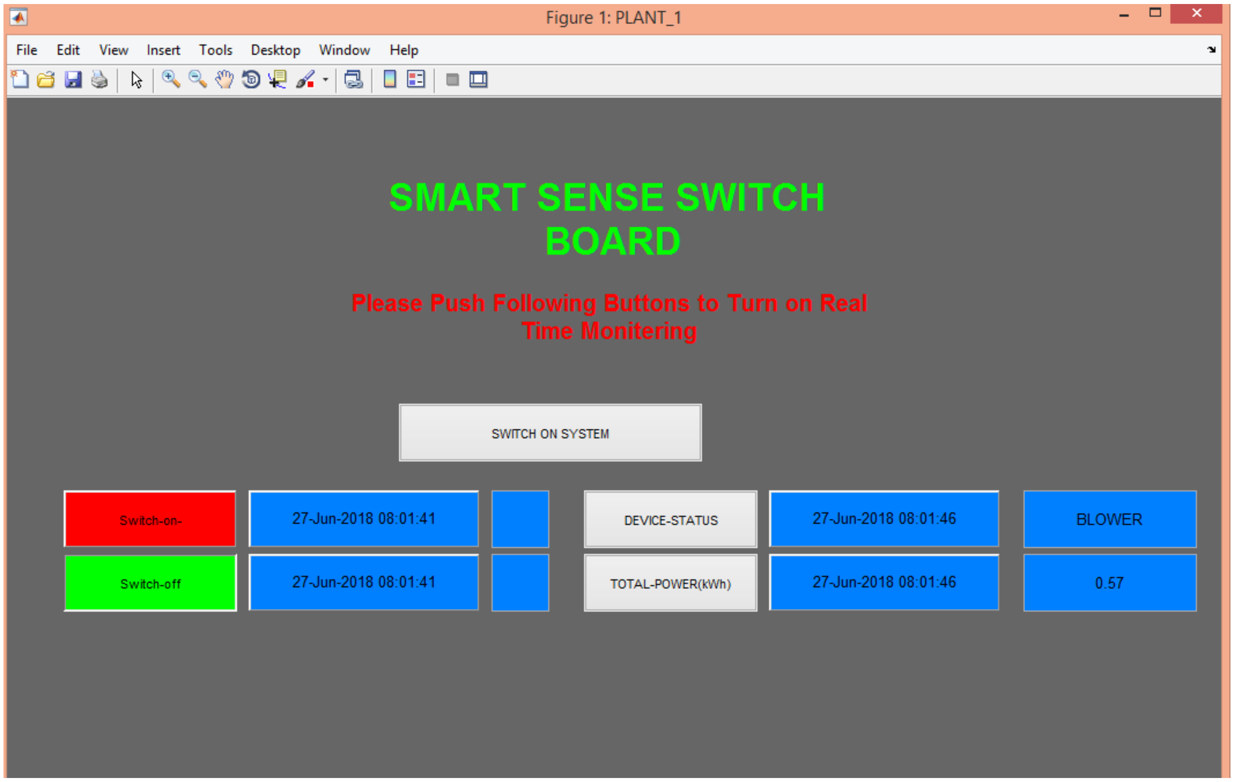
```
hsttext25 = uicontrol('Parent',f,'Style','edit','BackgroundColor',[0 0.5 1],
'Position',[880,205,150,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');

hsttext26 = uicontrol('Parent',f,'Style','edit','BackgroundColor',[0 0.5 1],
'Position',[880,150,150,50],'ForegroundColor',[0 0 0],'FontSize',10,'Visible','on');


set(f,'Name','PLANT_1')

movegui(f,'center')

h1button = uicontrol('Parent',f,'Style','togglebutton','String','SWITCH ON
SYSTEM','Position',[340,280,262,50],'Callback',@callbackfn2);

h2button = uicontrol('Parent',f,'Style','togglebutton','String','SWITCH-DEVICE-1',
'Position',[50,205,150,50]);

h3button = uicontrol('Parent',f,'Style','togglebutton','String','SWITCH-DEVICE-2',
'Position',[50,150,150,50]);


h5button = uicontrol('Parent',f,'Style','togglebutton','String','DEVICE-STATUS',
'Position',[500,205,150,50]);

h6button = uicontrol('Parent',f,'Style','togglebutton','String','TOTAL-POWER(kWh)',
'Position',[500,150,150,50]);


h7button =
uicontrol('Parent',f,'style','togglebutton','BackgroundColor','green','Position',[465,205,2
0,50],'Visible','off');

h8button =
uicontrol('Parent',f,'style','togglebutton','BackgroundColor','green','Position',[465,150,2
0,50],'Visible','off');

h9button =
uicontrol('Parent',f,'style','togglebutton','BackgroundColor','green','Position',[465,95,20,
50],'Visible','off');


set(f,'Visible','on')
```

```
function callbackfn2(hObject,eventdata)



while(get(h1button,'Value')==1)
 us1=get(h2button,'Value');
us2=get(h3button,'Value');

if us1==1
   th1=1;
   set(h2button,'BackgroundColor','green','String','Switch-off');
   set(hsttext11,'string',datestr(now));
else
   th1=0;
   set(h2button,'BackgroundColor','red','String','Switch-on-');
   set(hsttext11,'string',datestr(now));
end
if us2==1
    set(h3button,'BackgroundColor','red','String','Switch-on');
   set(hsttext13,'string',datestr(now));
   th2=1;

else
    set(h3button,'BackgroundColor','green','String','Switch-off');
   set(hsttext13,'string',datestr(now));
   th2=0;
end
```

```
thingSpeakWrite(463129,'Fields',[1,2],'Values',{th1,th2},'WriteKey','WIO0MIOULDEFL9NK');
data1 = thingSpeakRead(477068,'Fields',[1],'NumPoints',1);
data2 = thingSpeakRead(477068,'Fields',[2],'NumPoints',1);
data2=data2/100;
disp(data1);
disp(data2);
if data1==1
   set(hsttext25,'String','NO-dev');
elseif data1==2
   set(hsttext25,'String','MOBILE CHARGER');
elseif data1==3
   set(hsttext25,'String','LAPTOP CHARGER');
elseif data1==4
   set(hsttext25,'String','BULB');
elseif data1==5
    set(hsttext25,'String','BLOWER');
end
 set(hsttext15,'string',datestr(now));
 set(hsttext16,'string',datestr(now));
%%set(hsttext25,'String',data1);
set(hsttext26,'String',data2);
pause(12);


end


end
```

5.7.2 GUI Result:

Following result is obtained after compiling the MATLAB code:



SMART SENSE SWITCH BOARD

Please Push Following Buttons to Turn on Real Time Monitering

Switch-on-	27-Jun-2018 08:01:41		DEVICE-STATUS	27-Jun-2018 08:01:46	BLOWER
Switch-off	27-Jun-2018 08:01:41		TOTAL-POWER(kWh)	27-Jun-2018 08:01:46	0.57

Figure 5-26: Graphic User Interface

Chapter 6: Conclusion and Future Scope

Chapter 6: Conclusion and Future Scope

6.1 Conclusion:

Thus, as a conclusion we were able to make a device that was supposed to learn the behavior of the electric appliances considering the external and internal parameters. Basically, the voltage and current were recorded as an effort to distinguish the unique electrical signature of the electrical appliance and thus by the routine the correlation between the factors was calculated and in this manner by the help of machine learning a specific appliance was identified.

This is a device that may revolutionized a new concept in the field of home automation in which the user is provided an ease to first get a specific routine learnt and that after the learning of specific routine of the user, the device provides an ease to the user that afterwards from now onwards the user would not be using the ordinary and outdated switch boards for their appliances. Our device that will provide functionality of a normal switch board of will now act as the remote and pertaining to all the information recorded in the form of voltage and current values that is basically it corresponds to a specific value of these parameters and thus the machine is used to predict the connected appliance through the correlation maintained and take a decision of maintaining temperature accordingly.

6.2 Future Scope:

6.2.1 Domestic Market:

The device has a great scope in almost all the domestic market where the use of Switch Boards is necessary.

6.2.2 Energy Efficiency in Products:

The people concerned with the wastage of energy may buy the product that is a great research concept practically implemented so It can be served as an essential product that must be used to save energy.

6.2.3 Multipoint sensing:

In our project we are using single point sensing that is a single device is detected at a time but in future this project can be enhanced for multipoint sensing (sensing more than one device at a time) and it can be trained for several devices by maintaining data sets of desired devices.

6.2.4 Road towards Cost Management:

All the domestic users concerned with the cost management of the household products should use this product as this is one of those concepts that are a research-based concept but when practically implemented on such cheap and effective platform has many positive outcomes. It can serve to be cost effective. Thus, it can be served as an energy saving device and thus a necessary to be installed item in every household, offices and corporate sectors where the wastage of energy is frequent, and the saving of energy is required.

6.2.5 Enhancement of Relationship with Home automation startups:

All the companies that make smart automation systems for homes which cover the use of smart appliances are only made to provide ease to the user and the ease to the domestic purposes so those companies can be dealt for more effective ideas and formulation of much more efficiency in the product .Partnership can be done with **IOT** manufacturing companies on negotiable perspectives so that as this device servers to be one of the best smart IOT projects , devices can be made much more smart and user friendly that may ultimately serve the purpose of energy and cost efficiency through these devices.

Chapter 7: Bibliography

X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on, pages 253–264. IEEE, 2009.

J. Kelly and W. J. Knottenbelt. Metadata for energy disaggregation. CoRR, abs/1403.5946, 2014.

N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava. NILMTK: An open source toolkit for non-intrusive load monitoring. In Fifth International Conference on Future Energy Systems (ACM e-Energy), Cambridge, UK, 2014.

Y.-H. Lin and M.-S. Tsai. A novel feature extraction method for the development of nonintrusive load monitoring system based on BP-ANN. In 2010 International Symposium on Computer Communication Control and Automation (3CA), volume 2, pages 215–218. IEEE, 2010

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, Advances in Neural Information Processing Systems 27, pages 3104–3112. Curran Associates, Inc., 2014.