SmartOFF

Power Supply Management of Appliances for Energy Conservation

Group: **Bhushan Sonawane 111511679**
Nishant Borude 111447198
Ishupreet Singh 111424300

Objective:

The idea is to build a system by combining IoT and Machine Learning methods for accurate prediction of a particular appliance usage from residential data, and cut off the power supply to that appliance when not is use. This will help reduce energy wastage from all devices that consume some power in standby mode for most of the day when not in use.

Introduction:

Previous Work Done:

1. **Plugwise**: A smart gadget which consists of 9 plugs called Circle, which can remotely manage the devices connected through it. It can switch the devices on or off from the App and also turn off the device for specific time slots specified by the user.
   Difference: Doesn't automatically detect the usage patterns.
   Link: https://www.plugwise.com/products/home-stretch-basic

2. **Belkin Energy Saving Switch**: This is a switch which can cut off the power entirely supplied to a device connected through it.
   Difference: Manually switch the device on or off.
   Link: http://www.belkin.com/us/p/P-F7C009/

3. **IGo Green**: An energy saving switch board, which cuts off the power supply to the connected devices when they are fully charged.
   Difference: Can be used to control the energy supplied to battery powered devices such as phones and laptops.
   Link: https://www.youtube.com/watch?v=KwOuv-2_mDc

4. **Power Reduction for Smart Homes in an Internet of Things Framework**: A smart meter approach by applying algorithms to set threshold values for smart appliances.
   Difference: Doesn't cut off power completely and works only for smart devices.
   Link: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7535225

Statistics:

We referred the data obtained from Lawrence Berkeley National Laboratory website which states the standby power consumption for some daily use household appliances as follows:

Appliance	**Standby Power (Watts)**
Laptop Plugged In and powered off	8.9
Set Top Box	44.63
Microwave	3.08

This might not seem a lot of power consumption for individual appliance, but if we consider all such appliances plugged in for many residences, then the total energy wastage for standby appliances is significant.

Approach:

Project is split into hardware (IoT) and software (Machine Learning) part as follows:

1. **Internet of Things**: The end goal is to create a system which doesn't need much human intervention for successful functioning. We propose following two approaches:
    a. Locally attach a smart device to the appliance which will be connected to a central grid.
    b. Centralize everything and manage the devices from one smart grid.

   The aim in both the approaches is the same, i.e. cut off power supply to the appliance when not in use.
2. **Machine Learning**: In order to make the devices smart, we use machine learning to analyze the behavior of the consumer. To analyze behavior, here, means to recognize the patterns of the user's usage of a particular appliance, such as, the time ranges when the devices are used the most or the time ranges when the devices are in standby mode for most of the times and so on. Thus the system will know when the device is most used and when it is not. So, we can cut the power supply. We plan to make the models easily adaptable for new usage patterns. Hence, we'll be incorporating Transfer Learning.

Internet of Things

How does the AI model control the physical switch?

Hardware Used:

We used the following hardware components in the project:

1. ESP8266 Microcontroller: It has an ATMega processor with inbuilt WiFi support.
2. Relay module: This module was used to interface between the microcontroller and the mains switch.
3. Power Adapter: It was used to supply power to the ESP8266 and the relay module

4. Jumper Cables were used to make connection between all the above mentioned components.

The connections between all these components were not permanent, hence they were very fragile. This is why we did not use them for the demo during the presentation.

Architecture:

The overall architecture is divided into two parts:

1. Hardware part: This part deals with the connections between the various hardware model that are mentioned above. As shown in the figure below, pin 4 of the ESP8266 is connected to input pin of the relay and the power from the 5V DC power adapter is supplied to both the microcontroller and the relay module.

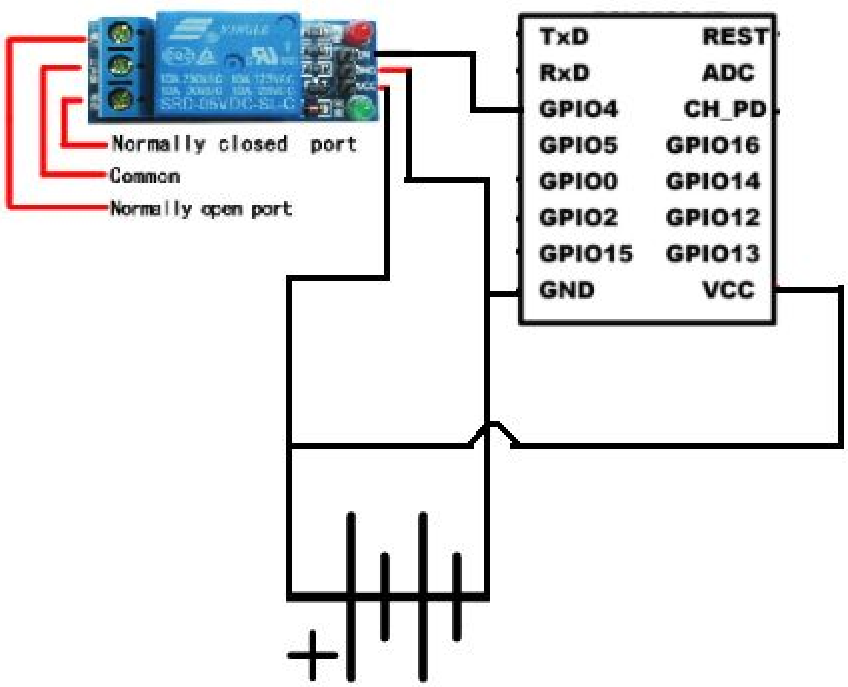


Figure 1: Circuit Diagram (Image Credits: [1](),[2]())

2. Software Part: This is the part that involved the maximum amount of research. The aim of this part was to communicate the signal being sent from the artificial intelligence to the microcontroller in a secure manner such that the microcontroller is able to control the switch.

The software has three major components which are as follows:

1. Machine Learning Component: This component has been explained in the previous sections.
2. Android App Component: This component was responsible for giving the user an interface to control his appliances remotely through his smartphone.

3. Microcontroller software component: This component was responsible for communicating with the Android app and the AI model.

Software Design

Embedded Software: We implemented 2 modes of operation in the microcontroller which are defined below:

a) Configuration mode: This mode is used to configure the WiFi credentials of the home network and the symmetric key used for the encrypted communication. In this mode only one client is allowed to connect to the microcontroller, this is to prevent the scenario when a malicious attacker is listening into the communication that is happening between the Android app and the microcontroller. If an attacker is able to listen to the communication when we transfer the symmetric key that is generated by Android app to the microcontroller, then the attacker can decrypt all the future communication that happens in the network.

When the microcontroller switches to configuration mode, only the management app is allowed to connect to it. The app generates the symmetric key and sends it to the microcontroller which uses it for all future communication. The next figure explains the handshake that happens in the configuration mode.

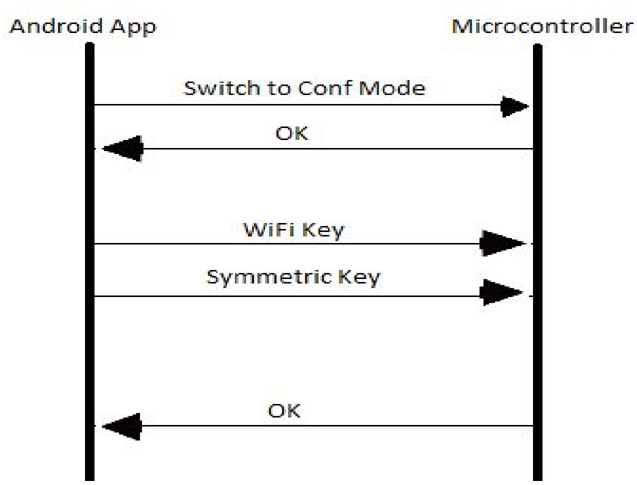


Figure 2: Configuration Mode Handshake

b) Operation Mode: This mode of operation deals with the general functioning of the project. In this mode, the AI model or the Android app sends the commands to the microcontroller to turn the switch on or off.

At first, the app or the AI model sends a UDP message at the broadcast IP address asking for the IP addresses of all the microcontrollers present in the network. Now the model or the app sends the command to the IP of the microcontroller which is connected the switch.

Before the sending the command, the app or the AI model encrypts the command and adds an initialization vector to the ciphertext so that no two ciphertexts are the same. This feature acts as an additional layer of security.

We also developed a protocol to facilitate communication between the microcontrollers. Every microcontroller also broadcasts its own WiFi network. In our protocol, if a microcontroller is not in the range of the home WiFi network then it would connect to nearest microcontroller and would use it as a hop to send the message across.

c) Android App: We developed an Android app to give the user option to control the appliances remotely through the App. This app was used to configure the home WiFi credentials and the symmetric key in the microcontroller during the configuration mode. In the operation mode, this app was used to send control signals to the microcontroller.

Machine Learning:

Data

No	Dataset name	Info	Link
1.	DRED	● Disaggregated data ● Waiting for permission	● http://www.st.ewi.tudelft.nl/~akshay/dred/
2.	UCI	● Individual household data for 4 years ● No individual appliances data but category of Kitchen, Laundry Room, Heater and AC.	● http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption
3.	REFIT	● 21 households data ● Individual appliances data ● Data over the year 2013-14	● https://pure.strath.ac.uk/portal/en/datasets/refit-electrical-load-measurements(31da3ece-f902-4e95-a093-e0a9536983c4).html
4.	UK-DALE	● 5-10 household data ● Total data available of 4-5 years ● Individual appliances data	● http://data.ukedc.rl.ac.uk/simplebrowse/edc/efficiency/residential/EnergyConsumption/Domestic/UK-DALE-2017/UK-DALE-FULL-disaggregated ● http://jack-kelly.com/data/
5.	AMPD	● Not available per individual appliances but category of Kitchen, BedRooms, .. ● 2 Years of data	● http://ampds.org/ ● https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/MXB7VO

6.	COOLL	● Individual appliances current and voltage measurement in June 2016 ●	● https://coolldataset.github.io/
7.	REDD	● Around 30 houses data ● Individual appliances data for current and voltage of several weeks ● Waiting for username and password for data	● http://redd.csail.mit.edu/
8.	BLUEd	● Data per appliances	● http://portoalegre.andrew.cmu.edu:88/BLUED/
9.	iAWE	● Per appliance	● http://iawe.github.io/
10.	PLAID	● 11 Different appliance type ● 55 household data collected during summer 2013	● http://plaidplug.com/

We choose UK-DALE data for further analysis because -

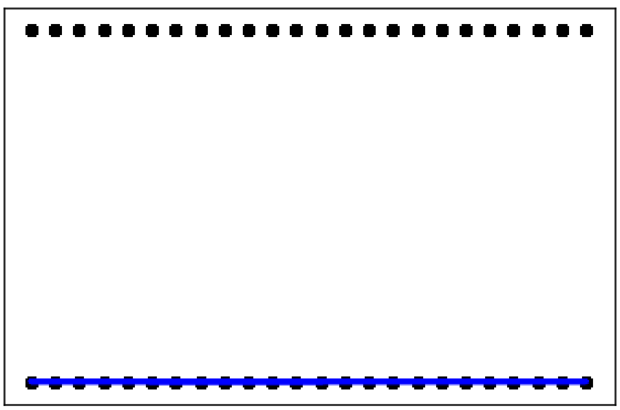
1. Consist of of 4-5 years for an individual home per appliance.
2. Data intervaled on every 6 seconds.
3. We can easily find threshold to cut usage into for stand-by mode and in-use.

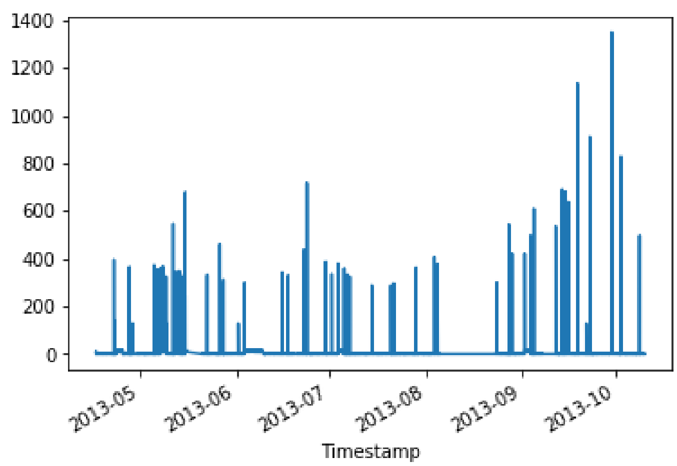
Approach

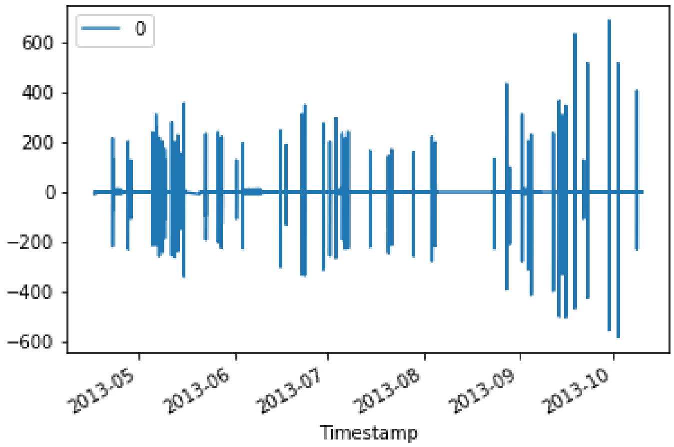
1. **Linear Regression**
    a. We applied linear regression as a basic model to analyse the data. We used the timestamp and preprocessed the data in order to get the features such as year, month, date, day, hour, minute and second.
    b. These are used as the features for training the data with the Usage of the appliance as parameter.
    c. However, this is not an appropriate measure to calculate the usage as the parameters are not uniform with respect to anything and won't produce meaningful results.
    d. With different combinations of features used for training, we got an MSE of 567.35 on the test data
2. **Supervised Learning**
    a. Next method we applied was converting the time series data into supervised learning data.
    b. The preprocessing phase includes considering all the possible values of Usage of a particular appliance and then mapping the values to 0 and 1.

c. The value is 1 denoting that the system in ON if the Usage value is greater than some threshold value.
d. This threshold is decided by the standby value consumption by the appliance.
e. Methods used:
    i. **Naive Bayes:** First model we applied using the sklearn package for supervised learning is Naive Bayes. We used GaussianNB() model for training over full dataset.
    Results: MSE = 0.11
    ii. **Logistic Regression:** We applied the same methods used above for this classification problem. We used LogisticRegression() model from sklearn
    Results: MSE = 0.01

Sample classification data using Naive Bayes which uniformly separates 1s and 0s from the data.



3. **ARIMA**
    a. Next we applied an algorithm specifically designed for analysing time series data.
    b. First, we preprocessed the data into the specific format for ARIMA.
    c. The timestamps are converted into appropriate date format such as YY-MM-DD HH-MM-SS as a string and then given to the model to train for the Usage.
    d. 

e. Above is the data plot for distribution of Usage over a period of 5 months for a 6 second separated observation for Treadmill.
f. On training over the complete data, we got the residual running plot as follows:
    i. 

g. The training accuracy was really good and we managed to get a testing accuracy of 0.003 The train plot is as follows:
h. The drawback of ARIMA: ARIMA works well when the data has some patterns in the distribution.
i. ARIMA exploits these patterns and makes predictions based on the inertia of the distribution. So, in this case ARIMA does well but not the best.
j. We need something which used memory based approach and that's where LSTM comes to the rescue.

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                D.Usage   No. Observations:              2089139
Model:                 ARIMA(5, 1, 1)   Log Likelihood            -5373186.725
Method:                       css-mle   S.D. of innovations              3.168
Date:                Mon, 04 Dec 2017   AIC                       10746389.450
Time:                        13:31:42   BIC                       10746489.868
Sample:                    04-16-2013   HQIC                      10746416.294
                         - 10-10-2013
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const         -4.564e-06      0.002     -0.003      0.998      -0.004       0.004
ar.L1.D.Usage     0.0495      0.067      0.738      0.461      -0.082       0.181
ar.L2.D.Usage    -0.0167      0.011     -1.542      0.123      -0.038       0.005
ar.L3.D.Usage    -0.0038      0.003     -1.084      0.279      -0.011       0.003
ar.L4.D.Usage    -0.0117      0.001     -9.733      0.000      -0.014      -0.009
ar.L5.D.Usage     0.0159      0.001     13.478      0.000       0.014       0.018
ma.L1.D.Usage    -0.2110      0.067     -3.147      0.002      -0.342      -0.080
                                    Roots
=============================================================================
                 Real           Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1          -1.7361             -1.4080j            2.2353           -0.3915
AR.2          -1.7361             +1.4080j            2.2353            0.3915
AR.3           0.8675             -2.0834j            2.2568           -0.1872
AR.4           0.8675             +2.0834j            2.2568            0.1872
AR.5           2.4770             -0.0000j            2.4770           -0.0000
MA.1           4.7400             +0.0000j            4.7400            0.0000
-----------------------------------------------------------------------------
```

4. LSTM

LSTM has been proved to be outperforming other machine learning models for time series data. LSTM is well-suited as device being on or standby-mode can be determined from past data.

A. LSTM model details

a. Parameters

Month, Day, Hour, Minutes, Seconds

Max normalized

b. 50 LSTM cells

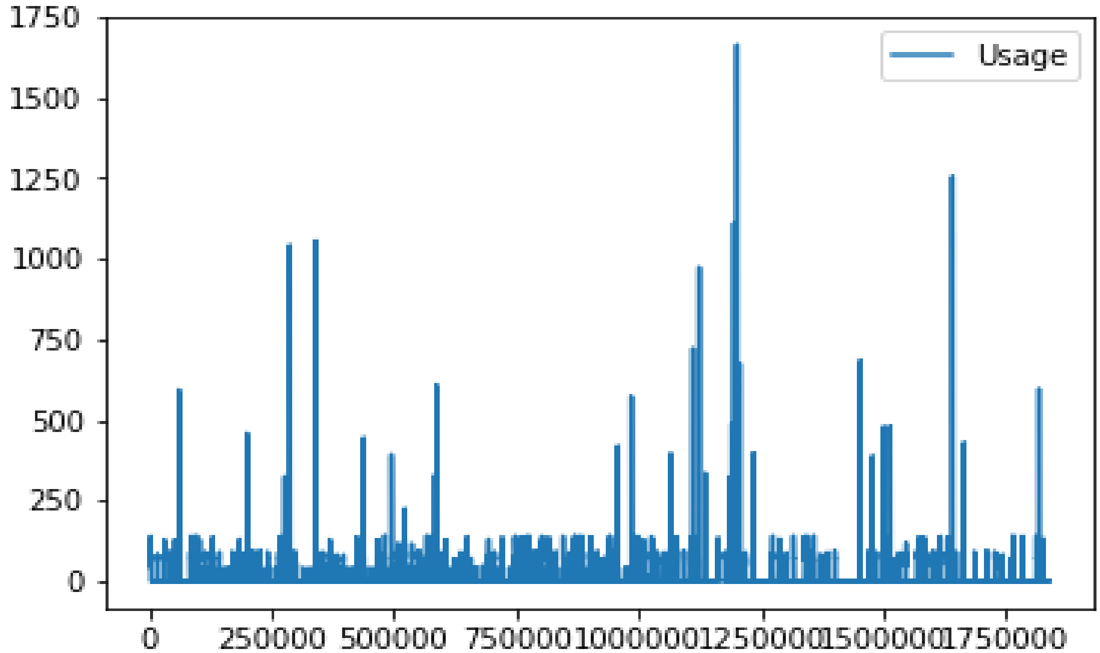
c. 1 Dense layer

d. 5-10 epoch

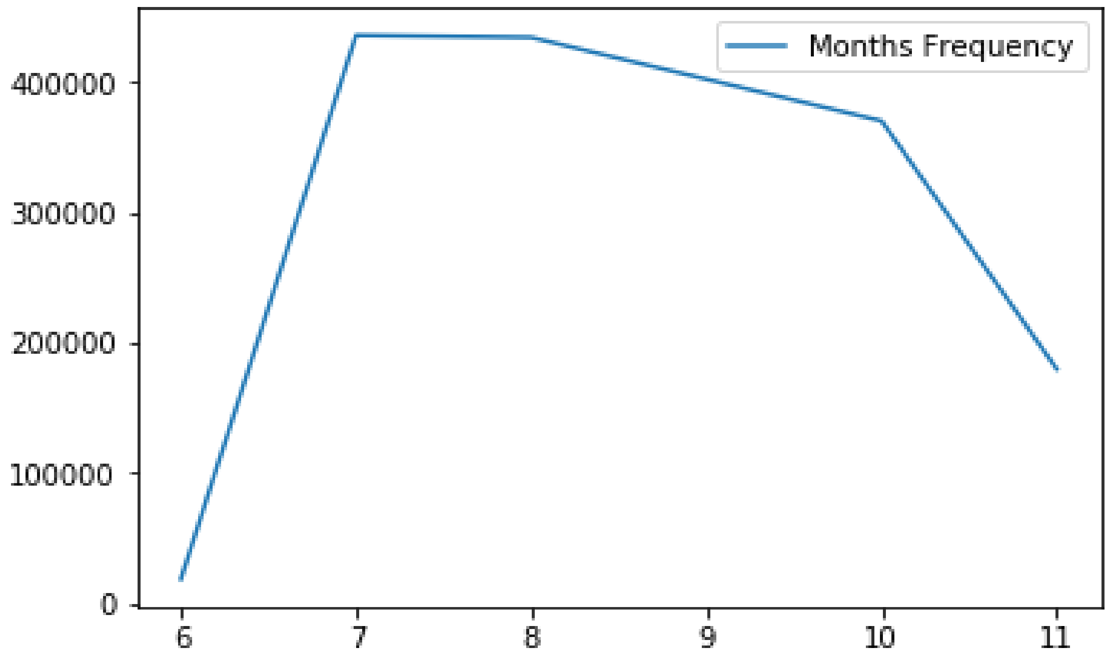
B. **Training**

a. TV data

i. Following is the TV data usage over the period of six months

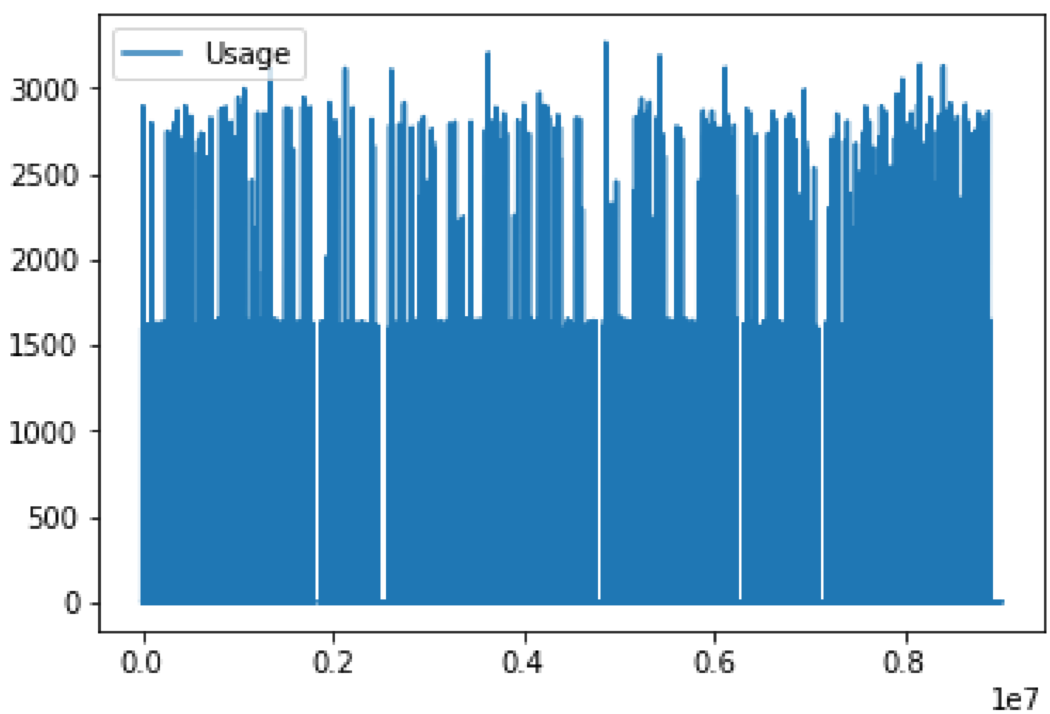


ii. Following is the monthly frequency of TV data

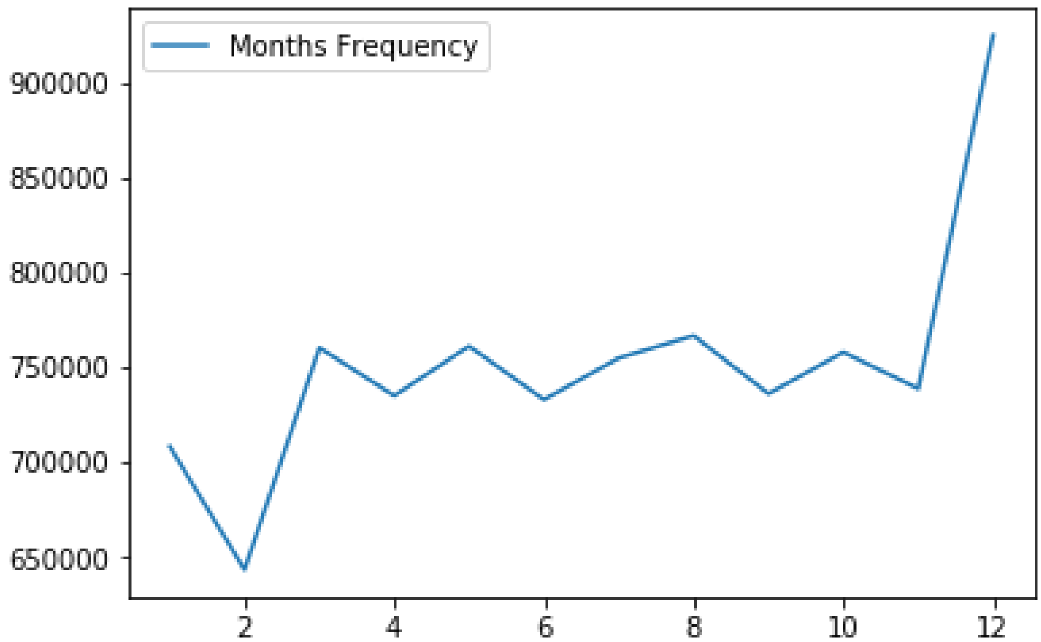


b. Microwave data

i. Following is the Microwave data over the period of twelve months

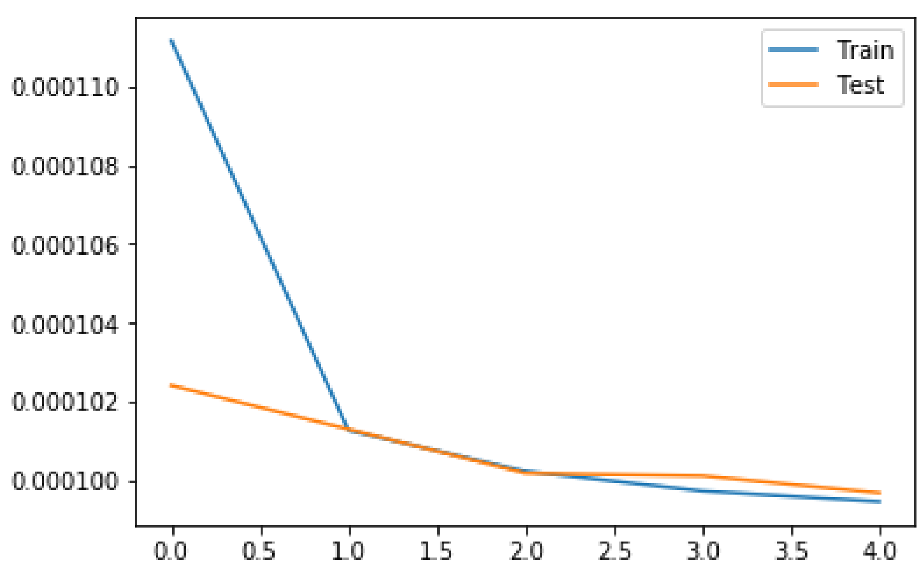


ii. Following is the monthly frequency of Microwave data

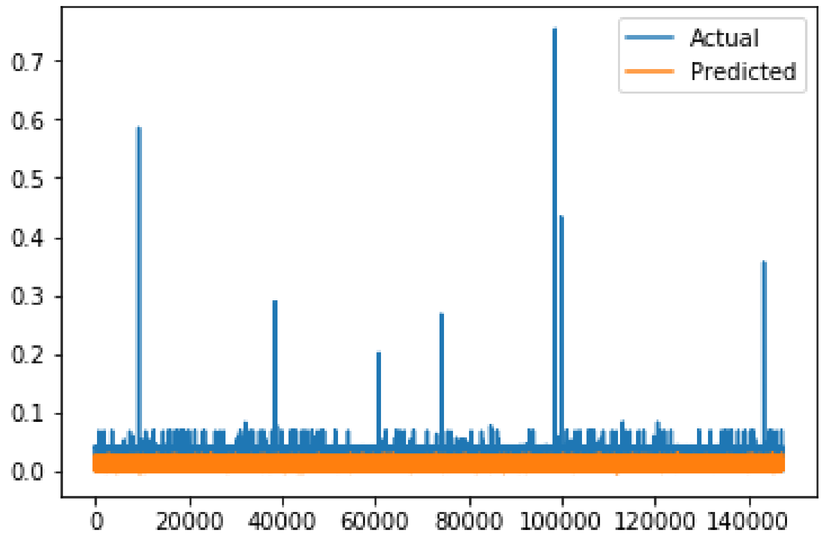


C. Analysis of TV data

1. We splitted data into 60 - 20 - 20 as Train - Validation - Test data
2. Following is the train - validation error during training



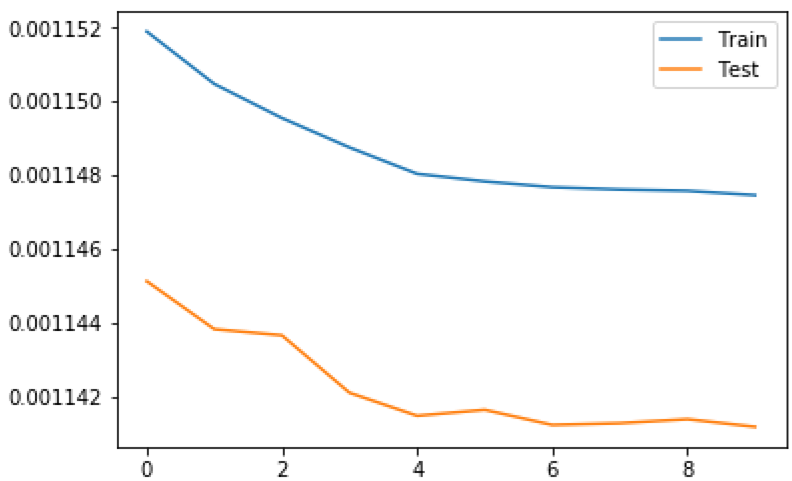
3. Following is the prediction vs actual usage value on test data



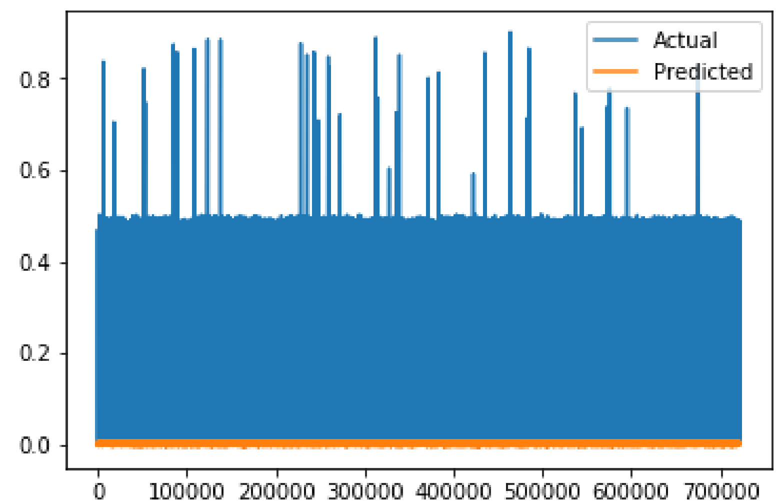
D. Analysis of Microwave data

We splitted data into 60 - 20 - 20 as Train - Validation - Test data

1. Following is the train - validation error during training



2. Following is the prediction vs actual usage value on test data



E. Accuracy

a. TV (Training) (MSE) -

i. Trained on 1104303 samples validated on 588962 samples

ii. Train loss: 9.9464e-05

iii. Validation loss: 9.9685e-05

b. TV (Test) (RMSE)

i. 0.000107

c. Microwave (Training) (MSE) -

i. Trained on 5411282 samples validated on 2886017 samples

ii. Train loss: 0.0011

iii. Validation loss: 0.0011

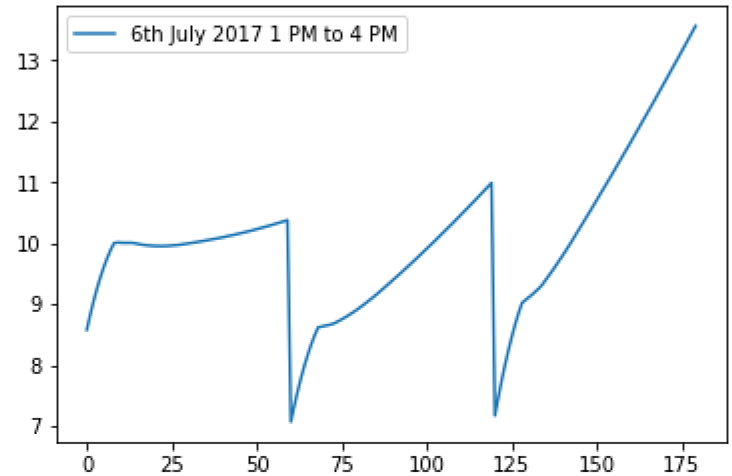
d. Microwave (Test) (RMSE) -

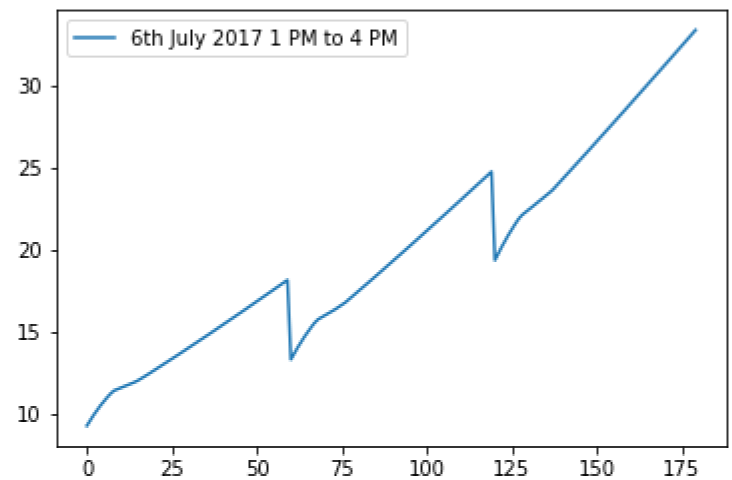
i. 0.0011371

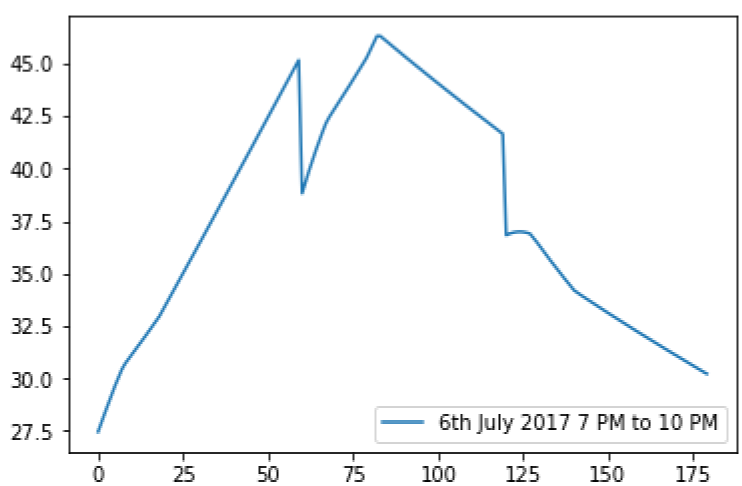
F. Predicting on random data

Now, trained model is saved and can be reused any time for finding if device is on or in standby mode. Following are some random time we are predicting if device is on or not. Below, we are predicting usage for next three hours and will cut-off the device on it's standby mode threshold.

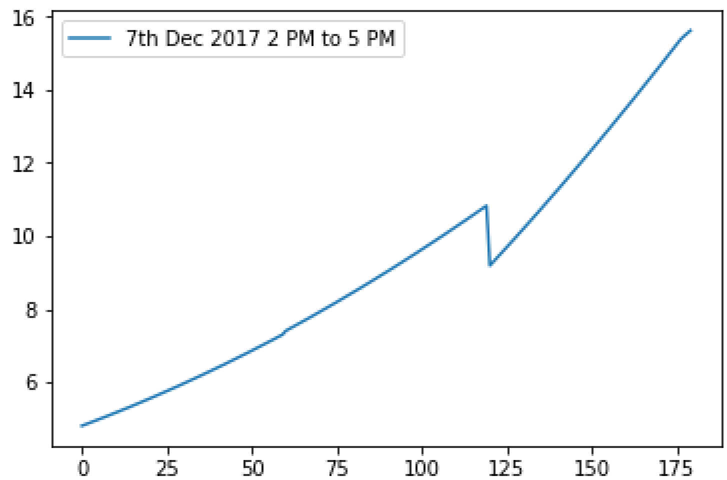
a. TV

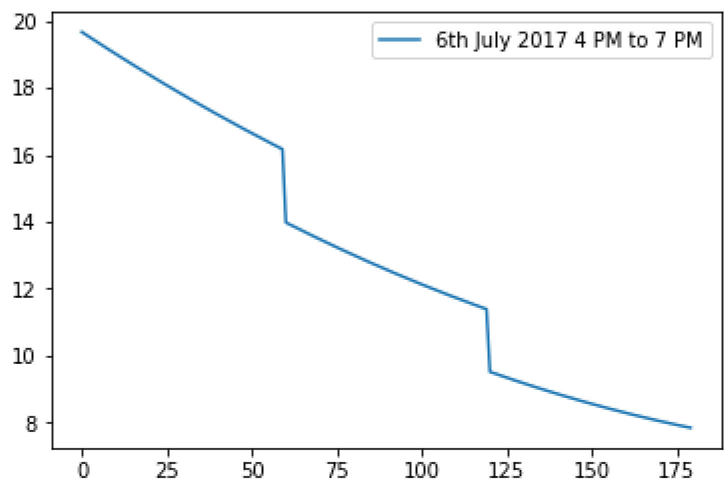

6th July 2017 1 PM to 4 PM
13
12
11
10
9
8
7
0
25
50
75
100
125
150
175

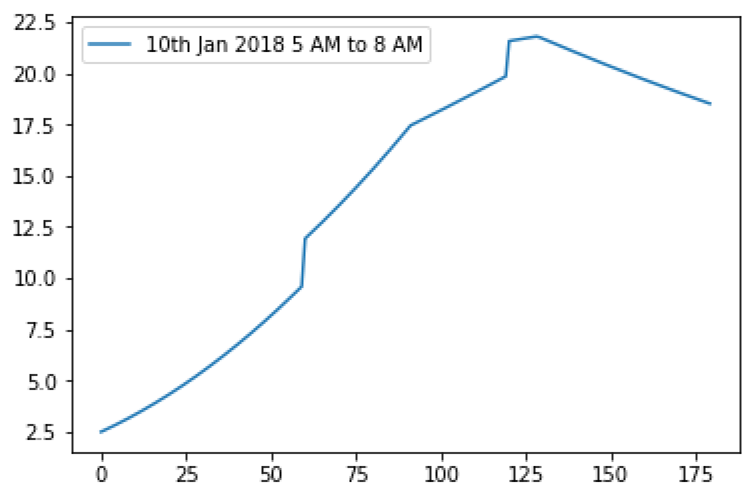

6th July 2017 1 PM to 4 PM
30
25
20
15
10
0
25
50
75
100
125
150
175


45.0
42.5
40.0
37.5
35.0
32.5
30.0
27.5
6th July 2017 7 PM to 10 PM
0
25
50
75
100
125
150
175

b. Microwave


7th Dec 2017 2 PM to 5 PM
16
14
12
10
8
6
0
25
50
75
100
125
150
175


6th July 2017 4 PM to 7 PM
20
18
16
14
12
10
8
0
25
50
75
100
125
150
175


10th Jan 2018 5 AM to 8 AM
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0
25
50
75
100
125
150
175

Interacting with IoT Device

1. Server

a. **Overview:**

   i. Desktop application with synchronizes with IoT device.

   ii. Uses saved trained model for predicting usage.

   iii. Two modes

      1. Instant sync: Synchronizes with current timestamp with device
      2. 3-Hour autopilot:
         a. Predicts usage for next three hours
         b. Find mode change i.e. from standby to on or on to standby and logs.
         c. Communicates with device on logged time frame.

b. **Implementation:**

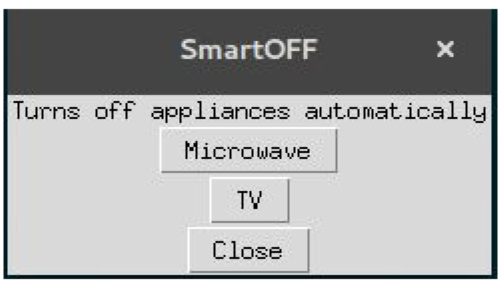
   i. User select device to control

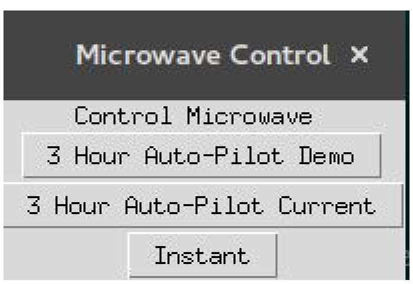
   ii. Loads respective trained model

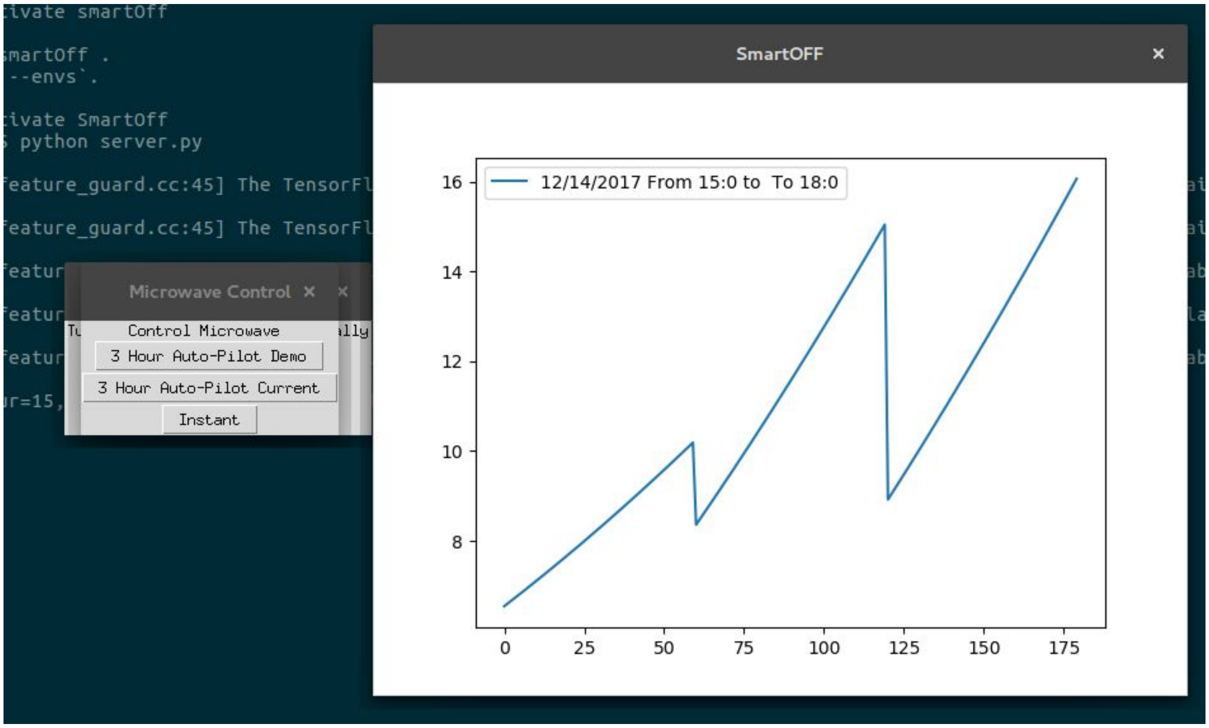
   iii. Predicts on selected model on given mode

   iv. Using tinydb for storing metadata in json file.

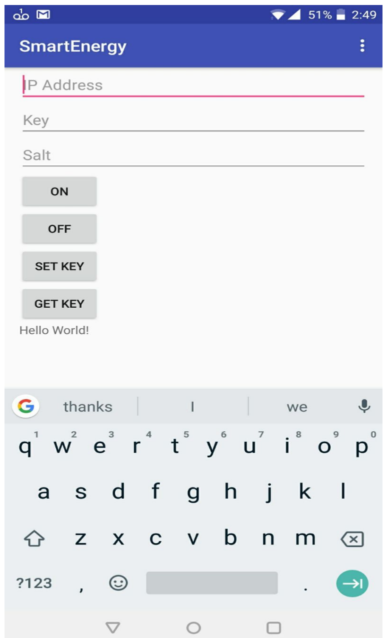
c. **Demo:**






SmartOFF
12/14/2017 From 15:0 to To 18:0
Microwave Control
Control Microwave
3 Hour Auto-Pilot Demo
3 Hour Auto-Pilot Current
Instant
python server.py

2. **Mobile application**
    a. **Overview:**
    Android application that allows user to manually control device.
    b. **Implementation:**
        i. Android Studio
        ii. Works for Android Kitkat onwards.
    c. **Demo:**



Source code:

Please find detailed source code here: https://github.com/bhushan23/SmartOff

Future plan of action

1. We can add a hardware component to the microcontroller which could monitor the actual energy consumption of the appliance and could feed live data into our model.
2. The communication mechanism between the microcontrollers can be made more robust. Making can extend out protocol such that theoretically infinite number of devices can be connected the internet.
3. Android app - adding autopilot mode in app.

Contribution

Task	Contributor
Paper reading	Ishupreet Singh, Nishant Borude, Bhushan Sonawane
Idea brainstorming	Ishupreet Singh, Nishant Borude, Bhushan Sonawane
Hardware implementation	Ishupreet Singh
Data analysis	Nishant Borude, Bhushan Sonawane
Linear regression	Bhushan Sonawane
Naive Bayes	Nishant Borude
Logistic regression	Nishant Borude
ARIMA	Nishant Borude, Bhushan Sonawane
LSTM	Bhushan Sonawane
Android application	Ishupreet Singh
Desktop server	Bhushan Sonawane