International Journal of Computer Applications (0975 – 8887)
Volume 179 – No.23, February 2018

Achieving Energy Efficiency using Green Internet of Things through Incorporation of Machine Learning Architecture

Srishti Sharma
Research Scholar
Computer Engineering Department
LDRP-ITR, Gandhinagar

Hiren B. Patel, PhD
Head of Department
Computer Engineering Department
LDRP-ITR, Gandhinagar

Bela Shrimali
Assistant Professor
Computer Engineering Department
LDRP-ITR, Gandhinagar

ABSTRACT
Global energy consumption hikes and natural resource depletion calls for fine-grained energy consumption on necessity basis. Our work focuses on the implementation of the concept of Green Internet of Things (Green IoT); using Internet of Things based architecture to induce autonomous sleep cycles in publically shared everyday usage appliances such as water coolers, coffee maker machines, vending machines, information kiosks etc. that are very commonly located in places such as schools, colleges, offices, tourism spots, airports, railways stations etc. where saving energy is usually not thought of. The approach presented here uses this IoT-based architecture to have the appliance report its usage pattern. The objective is to obtain the future usage forecast of the appliance made on the basis of the current usage patterns using the Machine Learning Architecture comprising of a Machine Learning Algorithm. The predicted usage data is then used to induce autonomous sleep cycles in the water cooler, for it to function as efficiently as possible, with least energy consumption. A water cooler system prototype is implemented using controller boards and sensors forming the IoT Architecture; the real time usage readings obtained from the prototype are used for predicting the future usage using ARIMA Machine Learning Algorithm, implemented using Python; and this forecast is then used for controlling the operation of the water cooler system.

General Terms
Internet of Things, Green IoT, Energy Efficiency, Machine Learning Architecture, ARIMA

Keywords
Internet of Things; Green IoT; Machine Learning; ARIMA; MQTT protocol; Energy Optimization-publically shared daily usage appliances.

1. INTRODUCTION
The increasing energy consumption is a major cause of greenhouse effect, causing ozone layer depletion. The statistical review of world energy-2017 report suggests that electricity consumption hike over the year 2005 to 2015 was 2.8% while that in the year 2016 alone has been 2.2% [1]. Fine-grained access to electricity is the need of the hour. Energy conservation is usually taken care of for the appliances located at personal places but there is a need for energy conservation of the appliance located at public places (shared appliances) [2]. For shared appliances, there is need of energy optimization for the appliances to work as efficiently as possible, on necessity basis. **Energy optimization** is predictive, organized and systematic coordination of use of energy to cover requirements while taking account of ecological and economic aims. The term thus describes actions for the purpose of efficient energy handling [3].

Internet of Things is a novel concept characterized by heterogeneous technologies, covering various aspects of modern wireless telecommunication. The proliferation of spatially distributed, uniquely identified "things" with sensing and actuating capabilities, that are seamlessly integrated with the environment around us, interacting and cooperating with each other, giving a common operating picture to achieve some common goal is what creates Internet of Things.

Fig 1: Internet of Things

The heterogeneous technologies that Internet of Things works in co-ordination with are Wireless Communication Technologies, Machine Learning, Big Data, Cloud Storage and Prediction and Forecasting. The extension of Internet into the physical environment by embedding electronics into everyday physical objects, making the digital and physical entities linked by means of appropriate communication technologies, offering a whole new set of services [4].

Adding Network Technology and Information Technology to an Embedded System, enabling the Embedded System to report the data gathered over a network to a remote storage such as Cloud; for the data to further be used for the following purposes- to display on a desktop/mobile application, to notify the occurrence of an action/event, to signal a wearable / working device to perform some action is what makes Embedded System - Internet of Things.

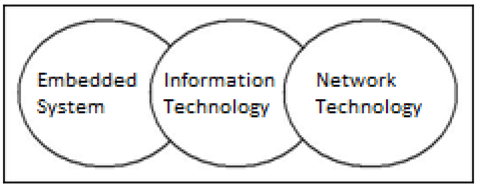

Fig 2: Internet of Things component Technologies

Green IoT stands for Green Internet of Things. It is a paradigm wherein the Internet of Things Architecture based procedures facilitate reducing the energy consumption of the existing appliances/applications so as to reduce the greenhouse effect. It is all about embedding context awareness or intelligence into the current appliances/devices using open hardware electronics such as sensors, Arduino Board, NodeMCU, and Raspberry Pi Board so as to transform these devices into eco-aware systems that facilitate metering their own energy consumption.

This metered energy consumption is then made use of to control the energy consumption so as to have optimized energy consumption on necessity and usage basis. This procedure of metering the energy consumption is rightly termed as "Smart Metering". Smart Metering could be done in 2 ways as per the current applicability:

i. Making use of a device like a smart socket to measure and control the energy consumption of the everyday usage appliances and

ii. Making these devices eco-aware by embedding the open hardware electronics into them to have them monitor and control their energy consumption on their own.

The proposed work makes use of the 2nd technique in order to avoid the need for an additional device in the system; also the reason being that the devices such as web sockets are expensive for everyday usage appliances.

Machine Learning Architecture is model focusing on making devices/appliances develop the ability of learning without explicitly being programmed [5]. Overcoming the legacy of strictly static program instructions, machine learning architectures facilitate data driven decision making, also called prediction [2]. The machine learning architecture in combination with Green IoT is used for enforcing optimized energy consumption in the current usage appliances. In the proposed work, the data gathered by the IoT Architecture, stored on cloud is fed into a machine learning algorithm to obtain the usage forecast for the coming week.

Over the years, the most popular and widely used forecasting models have been ARIMA (Autoregressive Integrated Moving Average) and ANN (Artificial Neural Networks) [6]. ARIMA is well known for its accuracy and flexibility in representing several different types of data. The drawback identified in ARIMA is its inefficiency in nonlinear time series modeling. ANN overcomes this limitation of ARIMA, but provides inaccurate forecast for purely linear time series [6]. Also, ANN is known to have "blackbox" nature, imposing large computation overhead and has proneness to overfitting [7]. The slow convergence rate of the backpropagation algorithm [8] too is bottlenecks for the performance of ANN. Experimental results of [9] suggest that in some cases, ARIMA gives greater improvement over persistence than ANN. Results in [3] prove that ARIMA performs better than ANN in terms of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Scaled Error (MASE) and Mean Absolute Percentage Error (MAPE) performance metrics in their specific case.

Current dataset and forecast expected being similar to the dataset and forecast shown in [3], it is concluded that ARIMA will be the suitable most model for implementation in the current work.

ARIMA Machine Learning Algorithm is a combination of differencing with the auto regression and moving average gives rise to a non-seasonal ARIMA model. ARIMA stands for AutoRegressive Integrated Moving Average model. The mathematical representation of ARIMA Model is as follows:

$$y_t^{'} = c + \emptyset_1\, y_{t-p}^{'} + \theta_1\, e_{t-1} + \cdots + \theta_q\, e_{t-q} + e_t$$

where $y_t^{'}$ is the series obtained after performing differencing. The "predictors" used on the right hand side include both lagged values of y_t and lagged errors [10]. This is called an **ARIMA (*p*, *d*, *q*) model**, where

p = order of the autoregressive part
d= degree of first differencing involved
q=order of moving average part

An **auto regressive (AR(p))** component refers to the use of past values in the regression equation for the series *Y*. The auto-regressive parameter *p* specifies the number of lags used in the model. For example, AR(2) or, equivalently, ARIMA(2,0,0), is represented as

$$y_t = c + \emptyset_1\, y_{t-1} + \emptyset_2\, y_{t-2} + e_t$$

where $\emptyset_1 \emptyset_2$ are parameters of the model [10].

A **moving average (MA(q))** component represents the error of the model as a combination of previous error terms e_t. The order *q* determines the number of terms to include in the model

$$y_t^{'} = c + \emptyset_1\, e_{t-1} + \emptyset_2\, e_{t-2} + \cdots + \theta_q\, e_{t-q} + e_t$$

Differencing, autoregressive, and moving average components make up a non-seasonal ARIMA model which can be written as a linear equation:

$$y_t^{'} = c + \emptyset_1\, y_{d\ t-1} + \emptyset_p\, y_{d\ t-p} + .. + \theta_1\, e_{t-1} + \theta_q\, e_{t-q} + e_t$$

where y_d is Y differenced d times and c is a constant [10].

Message Queue Telemetry Transport (MQTT) Protocol is a messaging protocol based on the publish-subscribe architecture. It is designed so as to establish connections with distant or remote locations where there is a requirement of a small code snippet or provided network bandwidth is limited [11]. The MQTT protocol finds a wide usage in the applications developed in the Internet of Things domain where most of the applications make use of cloud to have their data stored and accessed globally/remotely. The Hypertext Transfer Protocol has so far been the most used protocol in sending data to remote locations. MQTT stands out to be better than the HTTP protocol due to a certain reasons.

While HTTP is a request-response architecture protocol, MQTT is a publish subscribe mechanism based protocol. The publish-subscribe mechanism reduces the overhead involved in the request sent by the subscriber each time data required. As and when the data produced, the publisher publishes the data which is made available to the subscriber once the subscriber subscribes for the service [11]. MQTT protocol runs over TCP/IP protocol which is a reliable protocol compared to the UDP protocol on which Hypertext transfer protocol runs. The MQTT protocol is secured than Hypertext transfer protocol. MQTT allows one-to-one and one-to-many delivery whereas HTTP allows only one to one delivery of data [11].

The paper is structured as follows. Section 2 gives an overview of the related work carried out in this domain. Section 3 represents the proposed architecture, proposed technique, how can smart metering be done and how can the gathered data be processed using machine learning algorithm ARIMA. In Section 4, the implementation of a prototype of the proposed architecture as well as the application of the proposed technique over the prototype has been described. Section 5 includes the experimental results along with the analysis and discussion regarding the results. Section 6 concludes the paper and indicates the future direction for research.

2. RELATED WORK

Increasing energy consumption awareness amongst people as a basic measure for reducing the energy wastage is a global need today. There are measures taken by people for reducing the energy consumption of appliances in their home environments but there still is a very little contribution made for conservation of energy of an appliance shared by more number of people such as information kiosk, vending machine, currency exchange machine, water cooler etc. that are found at work places or other public places. In [3], the authors have worked upon reducing energy consumption or increasing energy awareness amongst people for appliances used at public places. The outcome of their research is a RESTful architecture letting internet connected device reduces energy wastage. The technique proposed here lets an everyday shared electrical appliance (coffee maker, beamer projectors, kettles, portable fans) report its usage patterns to Cloud Server where data is transformed into time-series and then processed to obtain appliances' next week usage forecast using ARIMA Model.

The test bed used by them is coffee machines situated in a corporate office, each connected to an Arduino Mega controller Board that reports the usage data to cloud where it is stored in CouchDB.

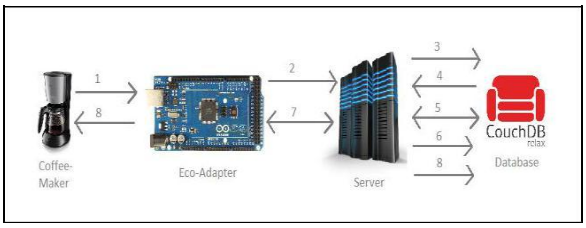

Fig 3: Proposed Method Paper 1 [3]

The energy event of the coffee maker machine is reported to the controller board. The controller board sends this energy event data along with the time stamp to the cloud server using HTTP protocol. This data is stored using CouchDB on cloud.

Every Sunday, 23 days of such data is fetched by the script running on the server. Machine Learning Algorithm ARIMA is applied over this data and prediction for the coming week (Monday to Saturday) is generated. The predicted data is then converted into a bit vector.

This bit vector is stored back in the database. The bit vector is then sent to the controller board using HTTP protocol on a get request from the controller. The eco-adapter then controls the coffee maker machine using the bit vector, turning the machine in ON state on encountering a '1' in the bit vector and in the STANDBY state when encountering '0' in the bit vector.

Work depicted in [12] also focuses on reducing the energy wastage with energy consumption awareness with a residence energy control system as the research outcome. The difference between [3] and [12] is that the usage pattern data is gathered using a smart socket in [12] instead of the IoT Architecture used in [3].

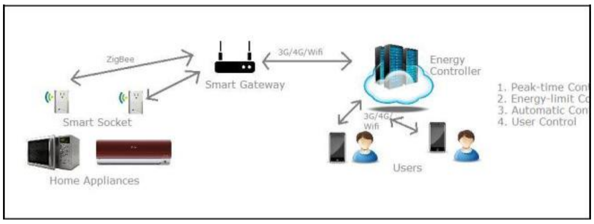

Fig 3: Proposed Method Paper 2 [12]

The use of a smart socket is made so as to record the usage pattern of appliances. Back propagation neural network algorithm is used to establish energy usage model based on history of smart socket. The experimentation has been carried out on home appliances like air conditioner, television are plugged into smart socket that sends data to Cloud server via home gateway using technologies such as ZigBee, Wi-Fi etc.

The research work in [13] suggests Optimization of the energy consumption pattern using a decision support system. The researcher has made use of metaheuristic forecast system and dynamic optimization algorithm for prediction using each of ARIMA, ANN and SVM. The research bed used is a smart grid installed at a residential building.

In [14], a Sparse coding based model for forecasting individual household electricity loads has been built using ARIMA and Holt-Winters Smoothing over a data set of 5000 households in a joint project with electric power board of Chattanooga, for the period from September 2011 to August 2013.

A predictive model for enhancing Environmentally Conscious Manufacturing (ECM) making use of ARIMA model for prediction of energy consumption and GHG emission (time series data) has been suggested in [15]. The research was carried out in an Indian pig Iron manufacturing organization with the objective of obtaining the energy consumption prediction/forecast.

3. PROPOSED METHOD

The proposed method is for implementation over a water cooler located in public places such as schools and colleges. The proposed method for implementation has four major steps:

1. Sensing (Sensing is required for gathering the usage data from the appliance)
2. Consumption Awareness (Storage and Analysis of the sensed data)
3. Future Usage Prediction (Feeding the stored data into a machine learning algorithm for future usage forecast)
4. Optimized Energy Consumption (Controlling the appliance, putting it in ON/OFF states as per the usage forecast)

Each of these steps can be formulated as a layer in the system architecture. Each of this layer formulated from the steps of implementation have been mentioned below in sequence:

1. Hardware Layer
2. Data Storage Layer
3. Processing Layer
4. Utility Layer

1. Hardware Layer.
The Hardware Layer consists of the following hardware components: Accelerometer sensor and Controller Board. The Accelerometer is connected to the water tap so as to detect motion that shall give the time stamps at which the tap was turned on. The Controller Board acts as an eco adapter that is a mediator between the appliance and the network. It sends the data sensed by the accelerometer sensor over cloud using the Wifi support. It is the foremost layer and constitutes step 1 in the implementation of the proposed method. MQTT Protocol is made use of to send the data.

2. Data Storage Layer
The Data Storage Layer stores the data sent by the controller board in a database over cloud.

3. Processing Layer
The Processing layer runs ARIMA Machine learning algorithm (implemented in combination of R and Python) over the data stored in the database with objective of obtaining the future usage forecast. A bit vector is generated with entries 1 corresponding to the timestamps that have been predicted tobe the ones when water shall be consumed (as per the forecast) and 0 for the rest.

4. Utility Layer
The Utility Layer sends this bit vector back to the controller board which makes use of this bit vector to turn the relay switch ON/OFF so as to control the power supply (working) of the appliance. MQTT Protocol is made use of for the purpose.

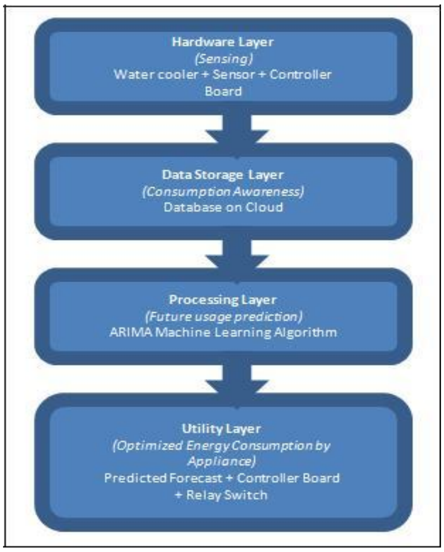

Fig 4: Layered Architecture

The stepwise proposed procedure:

1. The Accelerometer sensor gives a certain reading for its x, y and z co-ordinates when the tap is still. This reading varies when the tap is turned on. A constant reading ‘x’ and a reading obtained on turning the tap on ‘b’ distinguish between the 2 states namely tap in idle state and tap turned on.
2. These readings of the accelerometer sensor are sent to cloud by the controller board using MQTT Protocol and Wi-Fi support. They are stored along with the time-stamp in a database on cloud.
3. These readings stored in the database are then processed using the ARIMA Machine Learning algorithm, implemented in R and python.

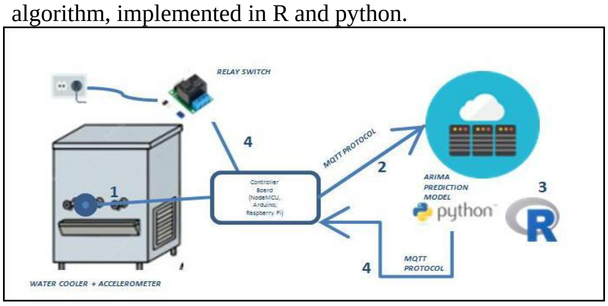

Fig 5: Proposed System Architecture

4. The output of this machine learning algorithm is the future usage forecast of the water cooler.

5. This output is converted into a bit vector assigning ‘1’ to all time stamps that are predicted to be the ones wherein water shall be consumed and ‘0’ to the rest of the time stamps.
6. This bit vector is sent back to the controller board using the MQTT Protocol.
7. The controller board uses the bit vector to control the relay switch that works on inputs 0 and 1 supplied to it from the bit vector on predicted time stamp basis.
8. The relay switch controls the power supply of the water cooler having it to work as efficiently as possible, on necessity basis.

The ARIMA Machine Learning Algorithm Implementation using R and Python is as under

1. The data stored in the database is first passed as an argument to auto.arima() method in RStudio, so as to obtain the order of p, d and q for the dataset

2. The dataset is then read in the form of a csv file in the Python IDE

3. The dataset is partitioned into 2 different datasets: training dataset and testing dataset

    a. Training dataset: The training dataset is used to make the model learn. It can be the entire dataset or a subset of the dataset (as per requirement)
    b. Testing dataset: The testing dataset is used to measure the correctness of the predicted forecast. The size of the testing dataset should be equal to the size of the expected prediction set.

4. Method ARIMA() is called with training dataset and the order of p, d and q as arguments.

5. The prediction is stored in a list called prediction

6. This prediction list is then mapped to a bit vector with each time stamp that is predicted to be the one where water is consumed be assigned value '1' and the rest have value '0'.
7. The testing dataset too is mapped to a bit vector in similar way.

8. The 2 bit vectors are then compared to measure the correctness of the prediction.

9. The predicted bit vector is sent to the controller board.

4. IMPLEMENTATION

The proposed technique has been implemented on a water cooler system prototype. The equipments used in the making of the prototype include: Water tap, Accelerometer Sensor, NodeMCU Board, and Relay Switch. The Accelerometer sensor is fixed upon the water tap and is connected to the NodeMCU using the jumper cables. There is a Wi-Fi support available for the NodeMCU Board so as to send the data read by the sensor to the server. A web service is made so as to have the ARIMA Machine Learning Algorithm run over the data gathered for obtaining the future predictions. A relay switch is connected to the NodeMCU which is operated using the future usage forecast showing the turning on and turning off of the water cooler machine so as to save power.

Objective of experiment: To get prediction of the water cooler tap usage on the basis of the current usage data, gathered using an accelerometer sensor.

- The expected prediction is supposed to be able to be mapped to 2 states namely 0 and 1 where: 1 shall be depicting the time interval wherein the tap was turned on and 0 shall depict the time interval wherein the tap wasn't turned on. The necessity of mapping the prediction to these 0/1 state is that these predicted state values shall be used to control the operating of the water cooler based on the usage prediction, in the future.
- The time stamp considered shall be of 5 minutes interval each so as to avoid huge amount of data and considering the fact that 5 minute time stamp would suffice accurate prediction making.
- Since the accelerometer gives its readings in the form of x-axis, y-axis and z-axis values and the expected prediction is in the form of Boolean states (0/1), there was a need for preprocessing the value read by the accelerometer to have a prediction made that could be mapped to 2 distinguishable states.

Thus, the experiment was carried out on the "Activity Recognition from Single Chest-Mounted Accelerometer" Data Set downloaded from UCI Machine Learning Repository

(https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer) [16].

The format of the data gathered was such that each record contained 5 values namely: time, x-axis, y-axis, z-axis and label. The description of these fields are as follows: time- time stamp value of the continuous readings received from the sensor, x-axis, y-axis and z-axis values- are the x, y and z axes readings of the accelerometer and label depicts the state(activity) recognized from these x, y and z axes readings. The total number of states/activities recognized from these readings is 7.

Table 1. Data Format – Activity Recognition from Single Chest-Mounted Accelerometer

Time	x-axis	y-axis	z-axis	State/label
0	1502	2215	2153	1
1	1667	2072	2047	1
2	1914	2373	2049	2
3	1903	2378	2051	2

The format needed for the experiment was to contain fields namely: time and a summarized reading (x, y and z axes) of the sensor. Also, since the objective of the experiment was to obtain prediction in the form of 2 states namely: 0/1, the experiment required only 2(Boolean) form states/activities to be recognized instead of 7. The sample dataset was preprocessed so as to bring it to the format needed for our experiment.

The dataset preprocessing steps included:

- Removal of the records belonging to states other than state 1 and 2(since experiment needed only 2 states recognition).

- Removal of the records belonging to states other than state 1 and 2(since experiment needed only 2 states recognition).

- Taking state 1 to be our out state 0 (tap not turned on) and state 2 to be our state 1 (tap turned on).

- Since the experiment required only 1 summarized reading instead of 3 different readings for x, y and z axes, a summarized reading was computed using the following formula [17]:

  Value=$\sqrt{(x-axis)^2+(y-axis)^2+(z-axis)^2}$

- Data was considered for 3 days – 20th, 21st and 22nd September, 2017 with 110 timestamps per day in the form 8:00, 8:05, 8:10,….,17:05 as being the college working hours. So the total number of records was 330.

Table 2. Activity Recognition from Single Chest-Mounted Accelerometer – Preprocessed as per required format

Time	Value
9/20/2017 8:00:00 AM	3434.769
9/20/2017 8:05:00 AM	3355.932
9/20/2017 8:10:00 AM	3673.272
9/20/2017 8:15:00 AM	3671.906

Here, the table depicts that on 20th September, 2017: The tap wasn't turned on in the time slot between 8:00 AM and 8:10 AM and was turned on in between 8:10 AM and 8:20 AM (in reference to table 1- state 1 and 2).

An observation made on the basis of the value computed using the 3 axes readings was that the computed value for records depicting state 1 in Table 1 was <3565.429 and that for records depicting state 2 in Table 2 was >3565.429.

The threshold value 3565.429 was obtained on calculating the median for the range of magnitude values obtained from the dataset.

From the above made observation the median of the range of magnitude values obtained comes out to be 3565.429, it is clear that there comes out to be a threshold value distinguishing between 2 states. *The threshold value will depend upon the dataset.* Here, the threshold value is 3565.429. So, the prediction values falling in the range above 3565.429 shall have bit 1 assigned for that interval and those falling below 3565.429 shall be assigned bit 0 in that interval.

For prediction of the future usage pattern, the algorithm ARIMA (Autoregressive Integrated Moving Average) has been made use of. The predicted values are mapped to a bit vector as in Table 3.

The expected values (training dataset values) are also mapped to a bit vector in the similar way. The 2 bit vectors are both having 110 entries each. The bit vector is used to turn the relay switch on and off so as to control the power supply of the water cooler system, the objective being reduction in wastage of power in idle hours.

Table 3. Predicted values to bit vector mapping

Time	Predicted Value	State	Bit Value
0	3586.301	>3565.429	1
1	3328.774	<3565.429	0
110	3679.288	>3565.429	1

The tools used for implementation are Arduino IDE, R Studio and Anaconda. The sensor data reading and sending it to the server is done using code written in Arduino IDE. R Studio is used for obtaining the order values of the p, q and r for the dataset gathered from the sensor data readings. Python IDE is used for the ARIMA implementation and Anaconda is used for generation of the future usage forecast.

Table 4. Observation drawn from Table 1 and Table 2

Table 1 (sample dataset)					Table 2 (preprocessed formatted dataset)		State
Time	x-axis	y-axis	z-axis	label/state	Time	value	-
0	1502	2215	2153	1	9/20/2017 8:00:00 AM	3434.769	<3565.429
1	1667	2072	2047	1	9/20/2017 8:05:00 AM	3355.932	<3565.429
2	1888	2374	2053	2	9/20/2017 8:15:00 AM	3662.681	>3565.429

3	1905	2375	2049	2	9/20/2017 8:20:2017 AM	3669.884	>3565.429
4	1611	1957	1906	1	9/20/2017 8:25:00 AM	3171.436	<3565.429
5	1601	1939	1831	1	9/20/2017 8:30:00 AM	3110.544	<3565.429
6	1910	2381	2044	2	9/20/2017 8:35:00 AM	3673.581	>3565.429
7	1921	2377	2048	2	9/20/2017 8:40:00 AM	3678.95	>3565.429
8	2000	1965	1879	1	9/20/2017 8:45:00 AM	3375.184	<3565.429
9	1917	2386	2047	2	9/20/2017 8:50:00 AM	3682.132	>3565.429

5. RESULTS AND DISCUSSION

The under mentioned linear graph-1 shows the bit vector values mapped from the usage prediction plotted against time.

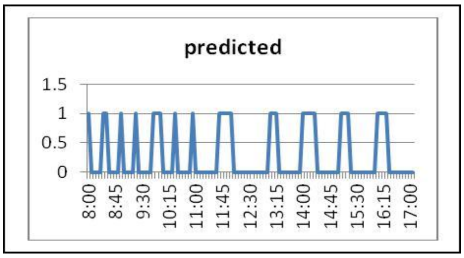

Fig 6: Predicted (bit) values vs. Time Stamp

Linear Graph depicted in Figure 6 shows bit vector mapped from testing dataset values plotted against time.

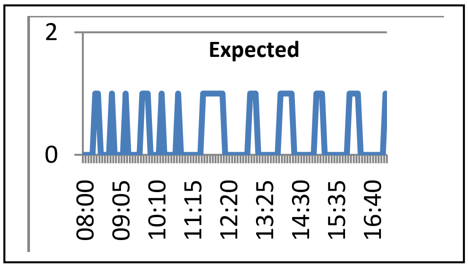

Fig 7: Actual/ Test (bit) values vs. Time Stamp

The figure 8 shows both expected and predicted valued bit vectors plotted against time.

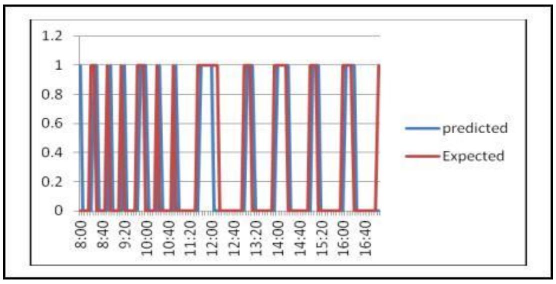

Fig 8: Actual (bit) and Predicted (bit) values vs. Time Stamp

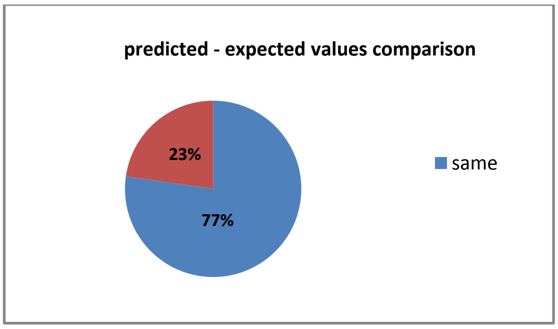

Fig 7: Comparison between Predicted and Test dataset values

The power consumption of a water cooler is 0.2 to 0.3 kWh per 24 hours [18]. Maximum power consumption of a water cooler depends on the power consumption of the compressor which makes it completely dependent upon compressor cycles per hour (CPH).The compressor in a water cooler currently has a CPH value ranging from 4 to 5. This means that the compressor starts up and shuts down for 4 to 5 times in an hour [19]. On applying the proposed method, the prediction that we obtained on the basis of the readings suggests that there are certain hours of the day wherein there is no requirement of these 4-5 compressor cycles. Table 5.1 shows the prediction data and the power supply status for one hour (8:00 AM to 9:00 AM).

Table 4. Prediction bit vector and Power Supply for one usage hour

Time Slot (hh:mm)	Prediction (m/sec^2)	Bitvector (0/1)	Power Supply
8:00	3617.790	1	ON
8:05	3145.945	0	OFF
8:10	3192.784	0	OFF
8:15	3220.797	0	OFF
8:20	3182.175	0	OFF
8:25	3422.124	0	OFF
8:30	3340.965	0	OFF
8:35	3258.327	0	OFF
8:40	3334.750	0	OFF
8:45	3257.279	0	OFF
8:50	3213.507	0	OFF
8:55	3586.805	1	ON
9:00	3748.180	1	ON

The table shows that there is a requirement of at maximum 2 compressor cycles in this hour. Similarly, there are more of such hours wherein the utility of the water cooler is less and henceforth the requirement of compressor cycles is also less. Lesser number of compressor cycles than 4-5 per hour shall effectively reduce a huge amount of power consumption when collectively taken into account.

6. CONCLUSION AND FUTURE PROSPECTIVE

The publically shared appliances that are very commonly located in schools, colleges, offices and other public places, such as water cooler, coffee-machine, information kiosks, PNR Status check machine at railway station, Currency Exchange Machines and other vending machines can have the proposed method deployed upon them, since an attempt of using the potential of IoT and Machine Learning to optimize the energy consumption of these appliances would be a step ahead in reducing the greenhouse effect, ultimately benefitting the society. Further, the researcher wishes to implement a prototype for this proposed work and verify results using readings generated from the prototype. The ARIMA Machine Algorithm can be replaced by a hybrid Algorithm consisting of features of both ARIMA and Artificial Neural Network for more accurate predictions.

7. REFERENCES

[1] Annual World Energy Economic Statistical Report 2017 https://www.bp.com/content/dam/bp/en/corporate/pdf/energy-economic/statistical-review-2017/bp-statistical-review-of-world-energy-2017-full-report.pdf

[2] Daniela Ventura, Diego Casado-Mansilla, Juan Lopez-de-Armentia, Pablo Garaizar, Diego Loex-de-Ipina, and Vincenzo Catania, "Embedding intelligent eco-aware systems within everyday things to increase people's energy awareness ", Springer-Verlag Berlin Heidelberg, June 2015

[3] White Paper issued by Siemens on Energy Efficiency, Energy Optimization and Energy Management https://w3.siemens.com/mcms/process-control-systems/SiteCollectionDocuments/efiles/pcs7/support/marktstudien/WP_Energy-Man-PA_EN.pdf

[4] Paraphrased from: Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of R&D doi:10.1147/rd.33.0210

[5] Paraphrased:"Machine Learning and Pattern Recognition can be viewed as two facets of the same field"

[6] Ina Khandelwal, Ratnadip Adhikari, Ghanshyam Verma, "Time Series Forecasting using Hybrid ARIMA and ANN Models based on DWT Decomposition", International Conference on Intelligent Computing, Communication & Convergence, ScienceDirect 2015.

[7] Jack V. Tu, "Advantages and Disadvantages of using Artificial Neural Networks versus Logistic Regression for predicting Medical Outcomes", Elsevier 1996.

[8] Bogdan M. Wilamowski, Okyay Kaynak, Serdar Iplikci, M. Önder Efe, "An Algorithm for Fast Convergence in Training Neural Networks", IEEE 2001.

[9] Gao, G., Lo, K. and Fan, F.L. (2017) "Comparison of ARIMA and ANN Models Used in Electricity Price Forecasting for Power Market". Energy and Power Engineering, 9, 120-126. https://doi.org/10.4236/epe.2017.94B015, April 2017

[10] ARIMA Machine Learning Algorithm – https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-sata-science-tutorials.

[11] HTTP vs. MQTT Protocols – http://www.rfwireless-world.com/Terminology/MQTT-vs-HTTP.html

[12] Kun-Lin Tsai, Fang-yie-leu, and Ilsun you, "Residence Energy Control System Based on Wireless Smart Socket and IoT", IEEE Access, May 2016.

[13] Jui-Sheng Chou, Ngoc-Tri Nago, "Smart grid Data Analytics framework for increasing energy saving in residential buildings", Elsevier-Automation in Construction-January2016.

[14] Chun-Nam Yu, Piotr Mirowski ad Tin Kam Ho, "A Sparse Coding Approach to Household Electricity Demand Forecasting in Smart Grids", IEEE Transactions on smart grids, December 2015.

[15] Parag Sen, Mousumi Roy, Parimal Pal, "Application of ARIMA for forecasting energy consumption and GHG emission: A case study of an Indian pig iron manufacturing organization", Elsevier-Energy, December 2016

[16] Dataset downloaded from UCI Machine Learning Repositoryhttps://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer

[17] The magnitude value from the three axis readings http://www.instructables.com/id/Accelerometer-Gyro-Tutorial

[18] Compressor cycle rate per hour https://reductionrevolution.com.au/blogs/news-reviews/57786245-water-cooler-water-boiler-energy-consumption-revealed

[19] Compressor cycle rate per hour https://www.researchgate.net/post/What_is_the_compressor_cycle _rate_for_water_coolers