Machine Learning for Smart Energy Monitoring of Home Appliances Using IoT

Rozeha A. Rashid, Leon Chin, M.A. Sarijari, Rubita Sudirman
School of Electrical Engineering
Universiti Teknologi Malaysia
Johor Bahru, 81310, Johor Malaysia
leonchinmpm@gmail.com,[rozeha| madib| rubita]@utm.my

Teruji Ide
Department of Electrical and Electronic Eng.
National Institute of Technology, Kagoshima College
Japan
t-ide@ kagoshima-ct.ac.jp

***Abstract*— Inefficient energy use has been a major issue globally. In Malaysia, the statistics reveal that residential energy consumption has been on a steady increase due to the growing population as well as a lack of awareness within households regarding proper energy utilization that causes significant amount of energy wastage. The emergence of Internet-of-Things (IoT) is a consequence and convergence of several key technologies such as real-time analytics, machine learning, sensors and embedded systems. The application of intelligence in IoT, known as Cognitive IoT (CIoT), will enable decision making based on historical data, and automatically train, learn and troubleshoot future issues. This project proposes a smart energy monitoring system for home appliances incorporating CIoT which consists of three parts. Firstly, a Raspberry Pi-based smart plug serving as the gateway, that is able to read current data from individual home appliances, load the trained model from training server and test the verified data using the model. Secondly, Google Colab as the training server will be used to store the training data set and building the Tensorflow-based Long Short-term Memory (LSTM) model. This recurrent neural network model will forecast electricity bill and notify users if abnormal energy consumption of individual home appliances is detected. Thirdly, a dashboard using Matplotlib library where users may monitor the real-time energy consumption. The completed LSTM model demonstrates a high accuracy of more than 80%, where a low mean squared error with train score = 0.1798 and test score = 0.1229 is determined. Furthermore, the R^2 test shows a high level of goodness of fit with train score = 0.844 and test score = 0.835.**

Keywords*— *Machine Learning, Smart Energy Monitoring, Long Short-Term Memory, Tensorflow, Keras

I. INTRODUCTION

Inefficient energy use has been a major issue globally, as a large percentage of current energy sources are generated using fossil fuels. Statistics provided by Tenaga Nasional Berhad show that 94% of electricity within the country is generated using fossil fuels, and the figure will likely remain unchanged in the next decade [1], while the greenhouse gas emission in Malaysia has contributed up to 40% of carbon gases, according to Rozana [2]. Simultaneously, energy consumption within Malaysian households have seen a steady increase, which according to the Malaysian Energy Commission, residential energy consumption has increased 38% from 1,937 ktoe (2010) to 2,678 ktoe (2016) [3] and is expected to increase by another 50% by the year 2030 [4]. By inferring from the said figures, two major factors can be attributed to the steady increase of residential energy consumption. Firstly, the rapid increase of urban population within Malaysia, from 17 million (2005) to 24 million (2018) [5], which consumes more electricity compared to rural households. Secondly, inefficient energy usage habits within Malaysian households such as keeping unused appliances turned on, lowering air-conditioner temperature to 16°C etc. results in energy wastage.

Internet of Things (IoT) can be defined as a connection which consists of sensor, actuators and other Internet-enabled devices to the Internet [6]. This feature allows the "Thing", which is the device itself, to connect to the Internet, process and manipulate data in real-time as well as to actuate the data in the form of intelligible machine actions. However, the capability of IoT alone is limited, in terms of machine intelligence. As the world inches towards the 4th Industrial Revolution, Cognitive IoT (CIoT), referring to the application of intelligence in IoT, uses a new computing paradigm called Cognitive Computing which is essentially the convergence of IoT with machine learning. The element of machine learning in IoT is important to transform it to be capable of behaving decisively on the basis of past data and events and automatically train, learn and troubleshoot future issues up to some good extent [7].

One of the solutions to a growing urban population and household energy consumption is a smart energy monitoring and management system. Commercially available products nowadays are all able to connect wirelessly to the Internet via an Internet of Things (IoT) framework, as well as enabling users to monitor energy consumption of individual home appliances real-time, displaying historical energy consumption and current accumulated energy bill. These features can help visualize these data better to the average household and in theory, help them manage their energy more efficiently. However, most existing energy monitoring systems lack the element of CIoT with machine learning that enables the manipulation of data from users' energy habits. Such examples include a ubiquitous home control and monitoring system through an Android-based application [8] and a smart home energy management system based on ZigBee and PLC [9]. The application of machine learning within the context of a smart energy monitoring system enables the system to make predictions such as forecasting energy bill etc. or automated decisions such as switching on/off appliances etc. based on historical energy habits and user preferences.

This paper will discuss the prototyping of a smart energy monitoring system based on CIoT, which will be built using a Raspberry Pi, serving as a gateway, to sense any input current data, store and display the real-time energy consumption on the Matplotlib-based Python program. Simultaneously, the Pi would also predict energy consumption bill based on the trained Long Short-Term Memory (LSTM) model and display it on the dashboard.

The paper will be organised as follows. Section II reviews several related works that have been done and are compared alongside with this project. Section III describes the methodology of the project in detail. Section IV will show the implementation result of the system, while section V will be a conclusion to this paper.

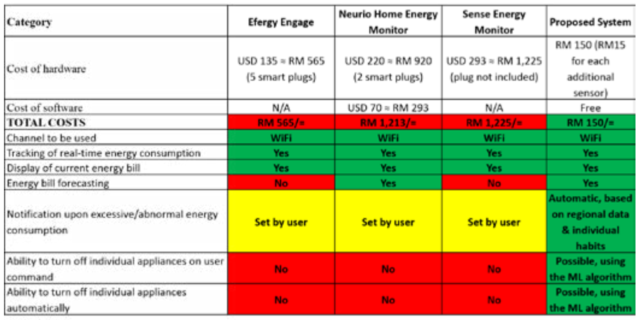
II. LITERATURE REVIEW

This segment highlights a few related projects and research regarding the topic of HEMS and the IoT technology. These works are guidelines and form the basis of the proposed system.

In the project completed by [8], the system allows home appliances control and monitoring through an Android-based application, where the gateway (Arduino Uno) is connected to appliances via Ethernet. The gateway acts as a micro-web server to manage, control and monitor system components, that enables hardware interface modules to successfully execute their assigned task using actuators and to report server with triggered events via sensors. This system relies heavily on wired Ethernet LAN connection and is lacking in terms of data transmission through WiFi. On the other hand, the proposed system by [9] considers both energy consumption and generation based on ZigBee and PLC-based renewable energy gateway (REG). The home server gathers both the energy consumption data through ZigBee and energy generation data through the REG. In terms of hardware implementation, the system is considerably complex due to the hardware implementation for energy generation analytics.

To tackle the issue of data transmission through WiFi and reducing the complexity of the system, [10] suggests a Home Energy Management System (HEMS) that uses Arduino Uno and multiple smart plugs to monitor energy in the house. Gateway can communicate with smart plugs to on/off devices based on user command on application. However, there is a lack of machine learning algorithm to implement automated decisions. Similarly, the project by [11] is based on both WIFI connection and LAN (Local Area Network) preconfigured in the house, consisting of wireless energy monitoring nodes and a Linux based single-board computer, used as a server, which can be accessed from the Internet, albeit the system is limited to only monitoring energy usage and is unable to conduct any form of decisions or data projection.

Commercially speaking, there is also a relative scarcity of machine learning algorithm among existing energy monitoring products which are capable of conducting forecasting. Fig. 1 below briefly compares the existing products with the proposed system in this paper.



Category	Efergy Engage	Neurio Home Energy Monitor	Sense Energy Monitor	Proposed System
Cost of hardware	USD 135 = RM 565 (5 smart plugs)	USD 220 = RM 920 (2 smart plugs)	USD 293 = RM 1,225 (plug not included)	RM 150 (RM15 for each additional sensor)
Cost of software	N/A	USD 70 = RM 293	N/A	Free
TOTAL COSTS	RM 565/=	RM 1,213/=	RM 1,225/=	RM 150/=
Channel to be used	WiFi	WiFi	WiFi	WiFi
Tracking of real-time energy consumption	Yes	Yes	Yes	Yes
Display of current energy bill	Yes	Yes	Yes	Yes
Energy bill forecasting	No	Yes	No	Yes
Notification upon excessive/abnormal energy consumption	Set by user	Set by user	Set by user	Automatic, based on regional data & individual habits
Ability to turn off individual appliances on user command	No	No	No	Possible, using the ML algorithm
Ability to turn off individual appliances automatically	No	No	No	Possible, using the ML algorithm

Fig. 1. Comparison with existing commercial products

III. METHODOLOGY

A. *System Architecture*

According to [12], IoT refers to the paradigm in which computing and networking capabilities are embedded in any kind of conceivable object, where the said object becomes the intermediary between the physical world and the digital world through the equipment of sensors and actuators. The most basic IoT architecture is a three-layer architecture that, as employed in this project, can be divided into 3 layers: physical sensing layer, IoT middleware (network layer) and the application layer. In the case of this project, the classification of each components to each layer is as shown in Fig. 2 below.

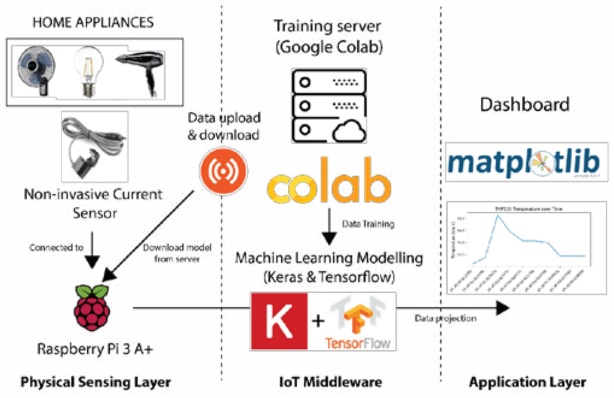


Fig. 2. System architecture of the Smart Energy Monitoring System

The physical sensing layer comprises of a non-invasive current sensor that is connected to the gateway, Raspberry Pi 3 A+, which is based on an existing project [13]. The circuit diagram is as shown below in Fig. 3 below, where the analogue signal from the current sensor using the MCP 3008 analog to digital converter. The non-invasive current sensor is essential for the hardware setup as any sensors that require serial connection to the appliance can be hazardous due to relatively high current involved. This is done by attaching the sensor to the live wire of the appliance, as shown in Fig. 4.

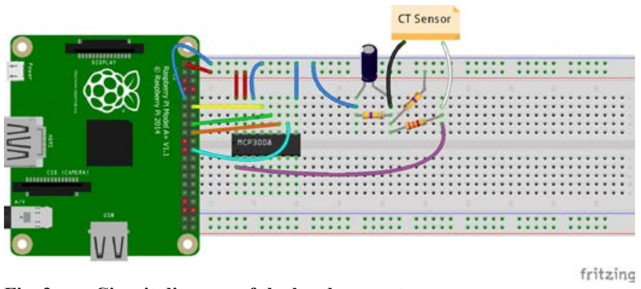


Fig. 3. Circuit diagram of the hardware setup

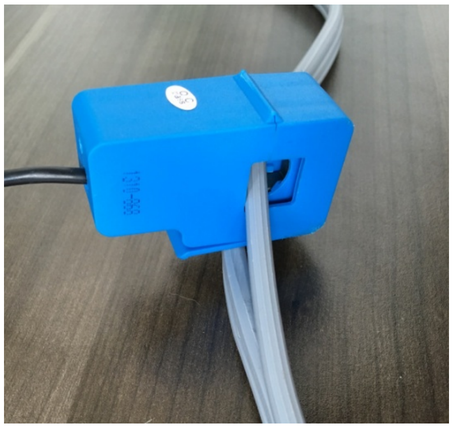


Fig. 4. Non-invasive current sensor attached to live wire

The output from the current sensor is also attached with a 3.3k burden resistor such that voltage and current reading can be carried out. The obtained data is logged and stored onto the Pi.

Within the IoT middleware layer, Google Colab is employed for the purpose of data set training due to its versatility and cloud solutions. In the context of machine learning, it provides the speed optimization by allocating GPU assets from Google servers to an otherwise limited hardware on the programmer end, which is vital to the memory-mongering machine learning algorithm. Google Drive provides a cloud drive platform to store this data set, subsequently loading and training it using the Colab online notebook. Subsequently, the trained model is loaded into the Pi and verified using the obtained current data.

B. Data Set Description & Implementation of Machine Learning Algorithm

The entirety of the training data set is retrieved from UCI Machine Learning Repository, namely the *Individual household electric power consumption Data Set* [14]. The data set is a time-series data, spanning over 2 million minutes in 1-minute intervals, measured from a single household in France. The data set contains several attributes, namely the date, time, global active and reactive power, voltage, current and active power readings from 3 different submeters. Each submeter reading corresponds to a maximum of 3 home appliances. In the context of this paper, the 3 submeters will be replaced with 3 different home appliances for simplicity.

LSTM stems from Sepp Hochreiter's 1991 analysis of a problem regarding learning to store information over extended time intervals via recurrent backpropagation is time-consuming, mainly attributed to insufficient, decaying error backflow [15]. To solve this, memory cells and gate units are introduced to the existing recurrent neural network, with memory cells controlling storage of data while the gate units block or pass on data based on its strength and import as shown in Fig. 5. A certain number of LSTM cells can be implemented within an LSTM layer and subsequently the whole layer may function as a parametric function, transforming an input and a previous hidden state into a new hidden state. In the context of LSTM, it may be useful to analogize each layer as a single entity or function.

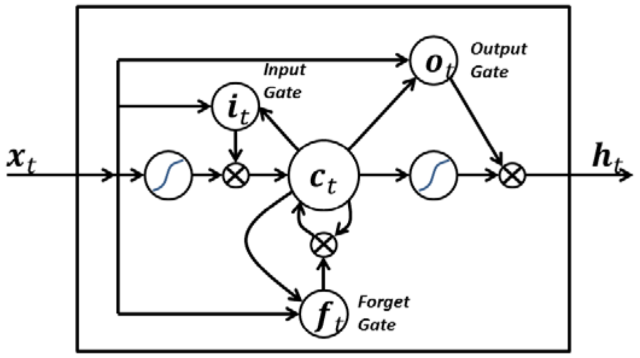


Fig. 5. Components of an LSTM cell

The machine learning method implemented for this project is deep learning as compared to a simple, traditional regression analysis. This is due to an abundance of resources on natural time-series data analysis. In this paper, implementation of the Keras documentation is demonstrated, which is an open-source neural-network library capable of integration with Tensorflow library. It was first developed for the usage of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and subsequently supported in Tensorflow's core library by Google's Tensorflow team in 2017 [16]. As outlined in the Keras documentation [17], the guiding principles of the Keras library are user friendliness, modularity, easy extensibility and works with Python. In the case of user friendliness, Keras offers minimal number of user actions required, a consistent and simple API while providing clear and actionable feedback upon user error. Neural layers (including Convolutional

Neural Network (CNN) and Recurrent Neural Network (RNN) layers), cost functions, optimizers and other functions are also modular and easily extendable from previous layers. Last but not least, Keras also runs on Python. The difference between the two libraries are significant. While Keras functions as a high-level model definition interface, Tensorflow mainly is used as the backend for the Keras models. Additionally, the modularity of the Keras model is demonstrated in Fig. 6 below, where a 3-layer LSTM RNN is implemented.

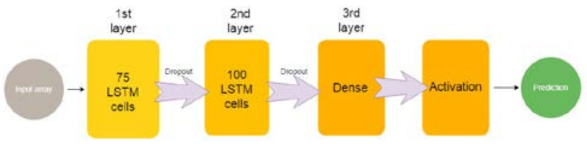


Fig. 6. Implemented LSTM layers

In the 1st and 2nd layer of the LSTM model, each layer contains LSTM cells which behave similar to neurons in an artificial neural network, and with each cell providing feedback to each other. The number of cells for each layer however is a hyperparameter, heavily relying on trial and error for optimization. In between layers, dropout method is implemented, in which random data is omitted in the model training to prevent overfitting [18]. For the purpose of this system, a factor of 0.2 is chosen (20% of data is dropped before being passed on to the next layer). For the 3rd layer, the dense layer is simply a regular layer of n neurons in a neural network. Each neuron receives input from all the neurons in the previous layer, outputting an n dimensional vector. In this architecture, n defines the number of future minutes of data for the model to predict, providing modularity and flexibility to the machine learning model. This is significant to the system architecture as energy forecasting in terms of global active power can be done for different number of days e.g. 1 day, 1 week or 1 month depending on user demand. Lastly, a simple linear activation function is implemented to convert an input signal of a node to an output signal.

C. Dashboard Integration

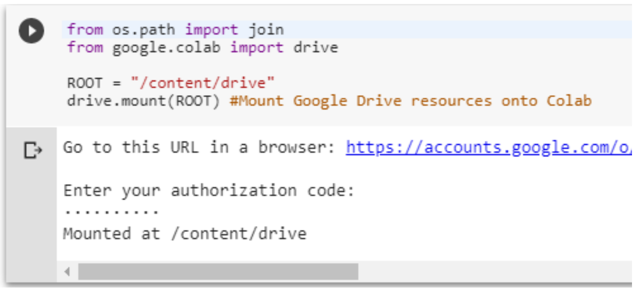
For a smart energy monitoring system to function as an integrated solution, a real-time data stream and monitoring is essential to provide accessibility to users to track their energy consumption and adjusting it accordingly. In this paper, the library used that is used to visualize this data is Matplotlib.

Matplotlib is a Python 2D plotting library which is commonly used for data visualization. The possibilities are endless as the library includes resources for plots, histograms, power spectra and many more with simply a few lines of coding. In the implementation of the live data stream, a Python program which includes the *FuncAnimation* method from the *matplotlib.animation* module is written. This allows the otherwise static graph to perform updates automatically at regular intervals and appending timestamps onto the plotted data. Subsequently, the data obtained from the hardware are formatted to match the attributes of the *Individual household electric power consumption Data Set*. For the purpose of the data stream, the data is refreshed according to the time interval as programmed in the data stream module.

IV. IMPLEMENTATION RESULTS

The implementation results show the developed components of the smart energy monitoring system: machine learning algorithm trained on Google Colab & dashboard module.

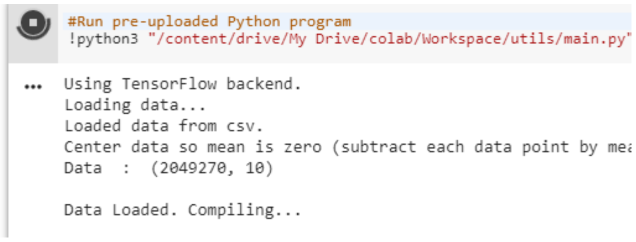
Simultaneously, the machine learning model is trained on Google Colab to make use of the GPU runtime for optimal calculation speeds. This is shown in Fig. 7, 8 & 9, where the resources on Drive is mounted onto Colab first before executing the training. The subsequent .h5 model file is stored onto Google Drive on the same root directory */content*.



```
from os.path import join
from google.colab import drive

ROOT = "/content/drive"
drive.mount(ROOT) #Mount Google Drive resources onto Colab
```

```
Go to this URL in a browser: https://accounts.google.com/o

Enter your authorization code:
··········
Mounted at /content/drive
```

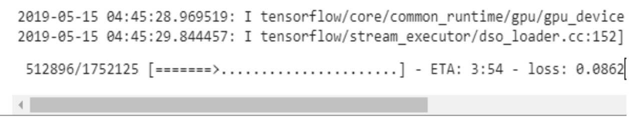
Fig. 7. Mounting Google Drive resources onto Colab



```
#Run pre-uploaded Python program
!python3 "/content/drive/My Drive/colab/Workspace/utils/main.py"
```

```
Using TensorFlow backend.
Loading data...
Loaded data from csv.
Center data so mean is zero (subtract each data point by mea
Data   :  (2049270, 10)

Data Loaded. Compiling...
```

Fig. 8. Data preprocessing sequence



```
2019-05-15 04:45:28.969519: I tensorflow/core/common_runtime/gpu/gpu_device
2019-05-15 04:45:29.844457: I tensorflow/stream_executor/dso_loader.cc:152]

 512896/1752125 [=======>.....................] - ETA: 3:54 - loss: 0.0862
```

Fig. 9. Executing data set training

To verify whether the model is working, the data has been split into training & test set in a 90:10 ratio and subsequently the tested data is plotted. In this prediction, if *t* is defined as the current time, *t*-10 minutes of historical data will be considered for the data prediction of *t*+5 minutes. The comparison between raw data and the prediction is as shown in Fig. 10 below, with the blue line as raw data and orange line for the prediction.

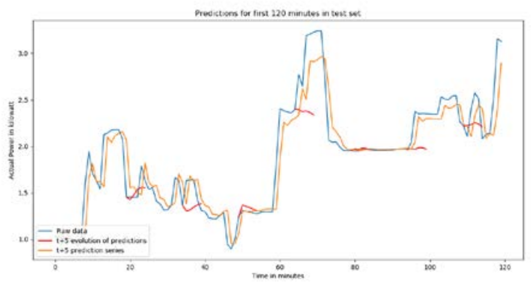
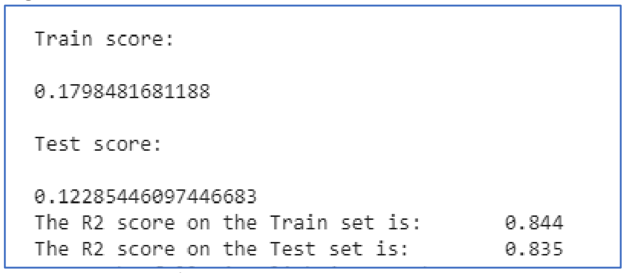


Fig. 10. LSTM energy consumption forecast

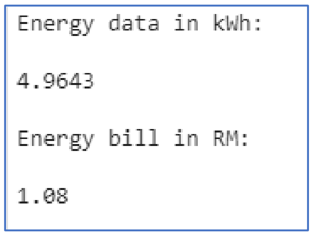
Additionally, a mean squared error (MSE) and R^2 test is implemented to verify the accuracy of the model and test goodness of fit of the model. By calculating the MSE scores for both the training data and test data, it is found that the train score = 0.1798 while the test score = 0.1229. The significance of this values is that there is only a small deviation (the closer to zero the better) from the original data set. Subsequently, the R^2 score for the train set and test set are 0.844 and 0.835 respectively, which can be interpreted as an 84.4% and 83.5% accuracy for the LSTM model. The values are as shown in Fig. 11.



```
Train score:

0.1798481681188

Test score:

0.12285446097446683
The R2 score on the Train set is:        0.844
The R2 score on the Test set is:         0.835
```

Fig. 11. MSE & R^2 scores

With the trained model in place, predicted energy consumption can be estimated from the model for the selected time period. After obtaining the predicted energy consumption, the energy bill can be calculated based on the tariff rates set by the utility company. Fig. 12 shows an example of utility bill estimation for a selected time period of 180 minutes.



```
Energy data in kWh:

4.9643

Energy bill in RM:

1.08
```

Fig. 12. Predicted energy consumption & calculated energy bill

V. CONCLUSION

Proper energy utilisation is an enormous hurdle to overcome, especially in this era where population is expected to increase rapidly, and global warming is already showing its effect. With the development of a smart home energy monitoring system which integrates machine learning to help households to manage their energy consumption, it is hoped that proper energy consumption habits can be instilled. In a nutshell, this project will help propel the utilisation of machine learning within the field of home energy management system and to create a more cost-effective solution for consumers.

REFERENCES

[1] Shahrul Nizam Mohammad, R. Z., Wahid Omar, Muhd Zaimi Abd Majid, Abd Latif Saleh, Mushairry Mustafar, Rosli Mohammad Zin, Noor Azland Jainuddin. 2014. Potential of Solar Farm Development at UTM Campus for Generating Green Energy. *Applied Mechanics and Materials*. 479–480: 553–558.

[2] R. Zakaria, K. S. F., R. Mohamad Zin, J. Yang, Samaneh Zolfagharian. 2012. Potential Retrofitting of Existing Campus Buildings to Green Buildings. *Applied Mechanics and Materials.* 178–181: 42–45.

[3] Malaysia Energy Information Hub. (2019). Statistics - Malaysia Energy Information Hub. [online] Available at: https://meih.st.gov.my/statistics;jsessionid=285BC564DEFA781C600C990836878F40?p_auth=NJE4RKhD&p_p_id=Eng_Statistic_WAR_STOASPublicPortlet&p_p_lifecycle=1&p_p_state=maximized&p_p_mode=view&p_p_col_id=column-1&p_p_col_pos=1&p_p_col_count=2&_Eng_Statistic_WAR_STOASPublicPortlet_execution=e1s1&_Eng_Statistic_WAR_STOASPublicPortlet__eventId=ViewStatistic3&categoryId=4&flowId=7 [Accessed 1 Jan. 2019].

[4] S Hassan, J & Mohamad zin, Rosli & Abd Majid, Muhd.Zaimi & Balubaid, Saeed & Hainin, Mohd Rosli. (2014). "Building Energy Consumption in Malaysia: An Overview." *Jurnal Teknologi*. 707. 2180-3722. 10.11113/jt.v70.3574.

[5] Worldometers.info. (2019). Malaysia Population (2019) - Worldometers. [online] Available at: http://www.worldometers.info/world-population/malaysia-population/ [Accessed 1 Jan. 2019].

[6] Serbulent Tozlu, Murat Senel, Wei Mao, Abtin Keshavarzian , Robert Bosch LLC , "Wi-Fi Enabled Sensors for Internet of Things : A Pratical Approach," IEEE Communications Magazine, vol. 50, no. 6th June 2012 , pp. 134-143, 2012.

[7] Dutta Pramanik, Pijush & Pal, Saurabh & Choudhury, Prasenjit. (2018). Beyond Automation: The Cognitive IoT. Artificial Intelligence Brings Sense to the Internet of Things. 10.1007/978-3-319-70688-7_1.

[8] Rajeev Piyare. "Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone" *International Journal of Internet of Things*, 2013

[9] Jinsoo Han, Chang-Sic Choi, Wan-Ki Park, Ilwoo Lee, and Sang-Ha Kim. "Smart home energy management system including renewable energy based on ZigBee and PLC" *IEEE Transactions on Consumer Electronics*, Vol. 60, No. 2, May 2014

[10] Sofia Ait Bellah. "Smart Home Energy Management System" Al Akhawayn University In Ifrane School Of Science & Engineering, 2015

[11] Evtim Peytchev, Mihail Lyaskov, Kostadin Popovski, Grisha Spasov. "Home Energy Monitoring System based on Open Source Software and Hardware" International Conference on Computer Systems and Technologies - CompSysTech'16

[12] Peña-López, Itu Internet Report 2005: The Internet of Things, 2005.

[13] Element14. (2018). Remote Home Monitoring with Raspberry Pi and Hologram Nova - Full Instructions. [online] Available at: https://www.element14.com/community/community/raspberry-pi/raspberrypi_projects/blog/2018/04/26/remote-home-monitoring-with-raspberry-pi-and-hologram-nova [Accessed 12 Apr. 2019].

[14] Hebrail, G. and Berard, A. (2012). UCI Machine Learning Repository: Individual household electric power consumption Data Set. [online] Archive.ics.uci.edu. Available at:

http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption [Accessed 12 Apr. 2019].

[15] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. *Neural computation*. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[16] "Module: tf.keras | TensorFlow", TensorFlow, 2019. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras. [Accessed: 22- Apr- 2019]

[17] "Home - Keras Documentation", Keras.io, 2019. [Online]. Available: https://keras.io/. [Accessed: 22- Apr- 2019]

[18] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15.1 (2014): 1929-1958.