INTELLIGENT PV MONITORING AND POWER EFFICIENCY PREDICTION USING ARTIFICIAL NEURAL NETWORK

NASIR NASIR-AMEEN 1 and RAHEENAT MOHAMMED 2

**Abstract:** In this paper, we reviewed solar energy as a solution to supply current and future energy needs. The inverter is a major component of these photovoltaic (PV) systems either autonomous or grid connected and It affects the overall performance of the PV system. This paper aims to examine the use of Artificial Neural Network (ANN) in the power output efficiency prediction of a photovoltaic (PV) system. An IoT data logger device was developed to collect intelligently and monitor in real-time the voltage and current flowing through the system. A mathematical equation is used to estimate the generated power. The initial system results show that the data logger device was efficient in transmitting the solar power usage details to the mobile application and the prediction results of a photovoltaic system show that the proposed approach has great accuracy and efficiency for forecasting the power output of the photovoltaic system.

**Keywords:** Photovoltaic system; Inverter efficiency; Power Monitoring; Microcontrollers; Internet of things (IoT); Artificial neural network;

1. Introduction

The fast evolution of renewable energy technologies during the last several years has led to the installation of many systems all over the world. However, most of these technologies have not yet achieved full-development. In the particular case of photovoltaic systems, detailed knowledge of the meteorological data for the location where the system will be installed is desirable, as well as full monitoring of the PV system performance. To charge a system storage battery, an intelligent PV system can use both electrical and solar energy. This can also be utilized to create electricity when one or both of the energy sources are unavailable. Solar panels generate electricity exclusively during daytime hours, with peak output around midday. This electricity is fluctuating and unrelated to household usage. To bridge the gap between what is created and what is required when PV systems are not producing solar electricity, it is critical to store energy for later use and to properly control energy storage and consumption.

1.1 Internet of Things

The Internet of Things (IoT) is a term that refers to physical objects (or groups of such objects) that have sensors, processing power, software, and other technologies that allow them to connect and share data with other devices and systems over the Internet or other communications networks. The Internet of things has been called a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable. [1]

Internet of things (IoT) is playing a crucial role in the daily life of humans by enabling the connectivity of many physical devices through the internet where the devices are intelligently linked together enabling new kinds of communication between things and people, and between things themselves to exchange the data for monitoring and controlling the devices from anywhere around the globe using the internet connection. Additionally, communication between machines or different devices is possible without human intervention using IoT applications. The idea behind the IoT principle is to connect the sensors and devices of a special system on a common network through wired or wireless nodes. In general, IoT based wireless systems are widely chosen to avoid associated risks with wired systems.[2]

* 1. Internet of Things and PV monitoring systems

The connected "smart house" is a good illustration of IoT in action. Internet-enabled thermostats, doorbells, smoke detectors and security alarms create a linked hub where data is shared between physical devices and users can remotely operate the "things" in that hub (i.e., modifying temperature settings, unlocking doors, etc.) via a smartphone app or website. [3]

Smart grids exploit the capability of information and communication technologies (ICT) to improve the sustainability, quality performance and balance of energy production and demand previsions, whereas reducing resource consumption. ICT also help the smart grid to integrate renewable energies.[2]

The goal of PV monitoring systems is to deliver continuously clear information about numerous factors, particularly the energy potential, extracted energy, fault detection, historical analysis of the plant, and associated energy loss. Furthermore, the observed data can be used for preventive maintenance, early identification of alerts etc. Many classifications of PV monitoring systems are based on internet technology, data collecting systems employed and monitoring system approaches have been overviewed in depth by the authors..[4]

The research in [5]was one of the first attempts to produce low-cost gear for solar radiation monitoring and then for environmental monitoring. The system was built around an 8-bit microcontroller that controlled an ADC and saved data in a serial EEPROM until it was connected to a portable computer. Four analogue inputs were available with minimal uncertainty due to an upgrade stage. Because the data acquisition system was powered by a rechargeable battery, the data were collected and stored at 10-minute intervals, and the power consumption was reduced by keeping the microcontroller in a low-power mode between intervals.

More microcontroller-based designs can be found in the literature, but some of them use a low-resolution ADC connected to an amplifier stage, which defines every input to a specific sensor [6], [7], while others rely on a PC [7]–[9], commercial software[10], or do not follow IEC standards to manage accuracy or obtain data, which offset the achievement of low cost, portability, and low power consumption, among other benefits.

Authors have shown that ANN and its associated models, such as FFBP, are popular techniques. An artificial neural network (ANN) is a model that uses interconnected neurons to generate output by processing data from the input. The weight of the neurons and the input data for which the neurons must be educated in the system determine the link between input and output.

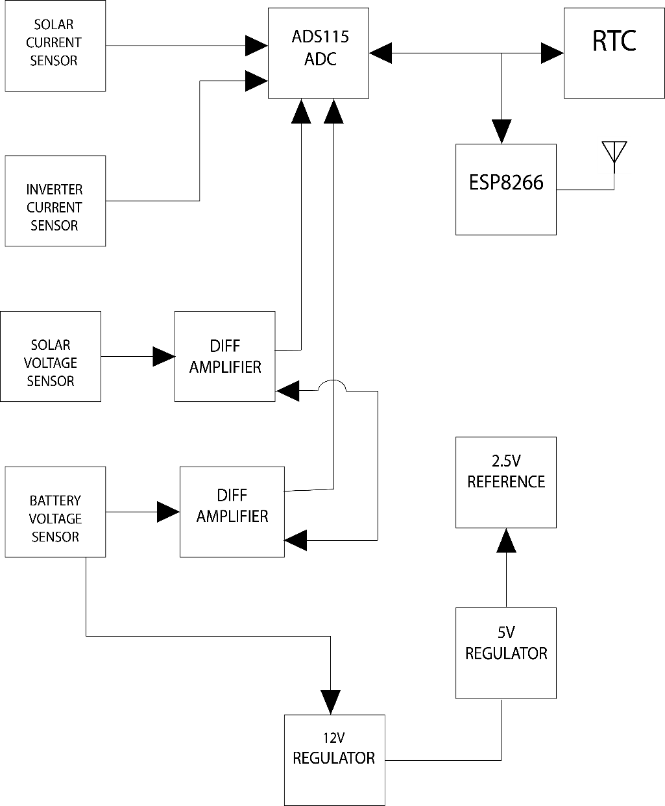
The authors in[11] used FFBP to anticipate temperature and found that it has the right capacity for complicated modeling of relationships between numerous factors. Because of its short training period and quick reaction, the author in [12] claims that FFBP is the finest neural network model for real-time forecasting.

For comparison and analysis of difficulties linked to solar panels, some researchers have used various forms of ANN. Applications for forecasting the power production of PV systems fall into two categories: the first is a prediction model based on solar radiation intensity. First, use the solar radiation model to compute the projected value of solar radiation, then apply the PV output formula to calculate the PV system's power production[13]. Another method is to estimate the PV system's power output directly. Although the prediction model based on solar radiation intensity has been deemed an efficient strategy in practice, solving complex differential equations necessitates a large amount of meteorological and geographical data. [14]

1. Materials and Methods

Since the data needs to be collected, processed, stored and analyzed in an IoT setup, a low-cost data pipeline for monitoring the electrical and environmental parameters in a photovoltaic station is designed for this work.

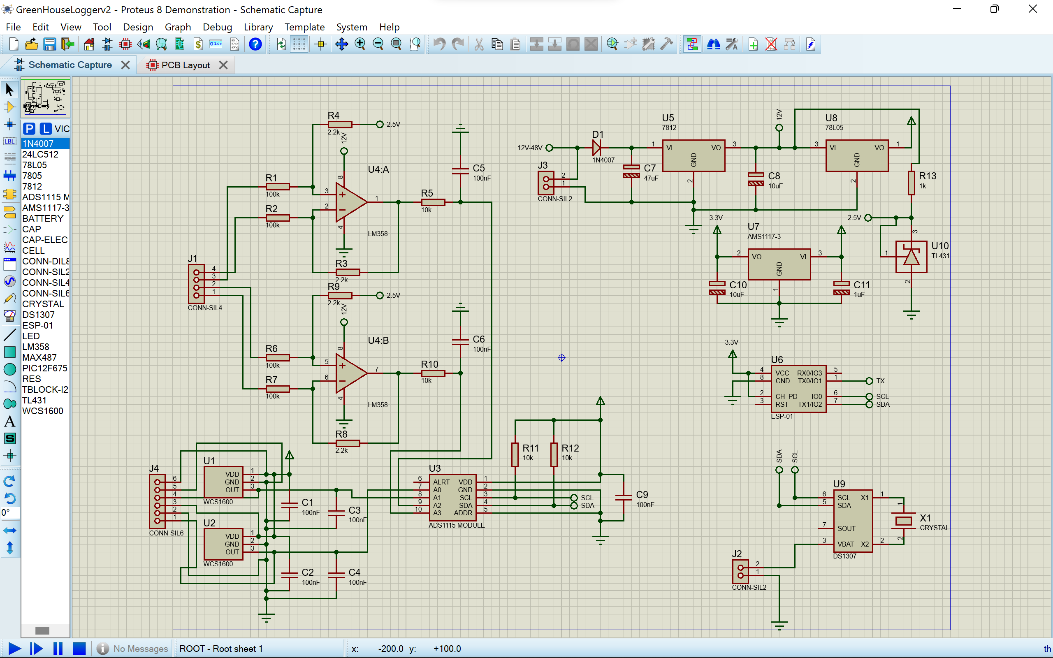
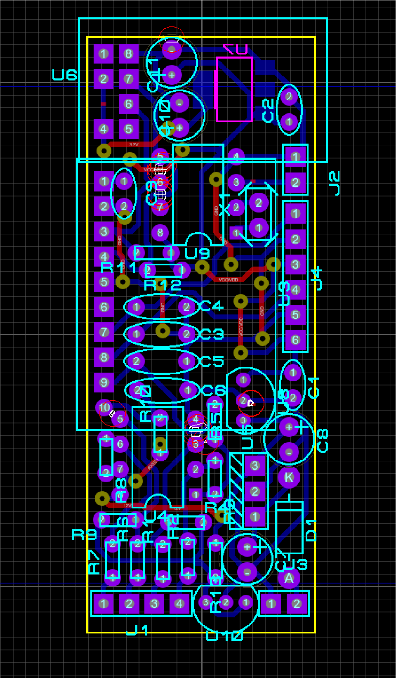
The ESP8266 module board serves as the microcontroller in the proposed monitoring system, acquiring and processing data from multiple sensors before transmitting the processed data to a mobile application via built-in Wi-Fi. There are two stages to data communication: The initial stage is inter-integrated circuit protocol (I2C) communication between sensors and the controller, followed by Wi-Fi protocol communication between the controller and the mobile application. Figure 2.1 shows the block diagram of the proposed IoT solution for monitoring photovoltaic systems:

****

***Figure 2.1 Block Diagram of Data Logger***

* 1. Hardware and Software Design of the Data Logger
     1. Datalogger Hardware design

The Proteus Design Set is a proprietary software tool suite that is primarily used to automate electrical design. It was used to develop schematics and electronic prints for this project. The developed data logger prototype was built on a programmable PWM solar charge controller. It includes components such as the sensors, voltage regulators, differential amplifiers, an ADC, an RTC and ESP8266 module integrated with a programmable charge controller. The experimental setup of the system hardware software is illustrated in Figure 3. 1. The developed experimental prototype consists mainly of the PV system including the battery, sensors and the dual core ESP32 controller

*Figure 2.2 : a). Circuit Diagram Of Proposed Data Logger System b) PCB Diagram Of Proposed Data Logger System*

2.1.2 Datalogger Software design

This process aimed to program the Data Logger’s microcontroller with instructions on how to interpret the data sent to the ADC, compute the parameters from that data and log the computed parameters; it was also during this phase that the instructions for IoT interactions between the data logger and mobile application were written.

NodeMCU is an open-source platform based on ESP8266 which can connect objects and enable data transfer using the Wi-Fi protocol. It includes firmware that runs on the ESP8266 which was used to program the instructions for the data logger. The program was written in the C++ programming language.

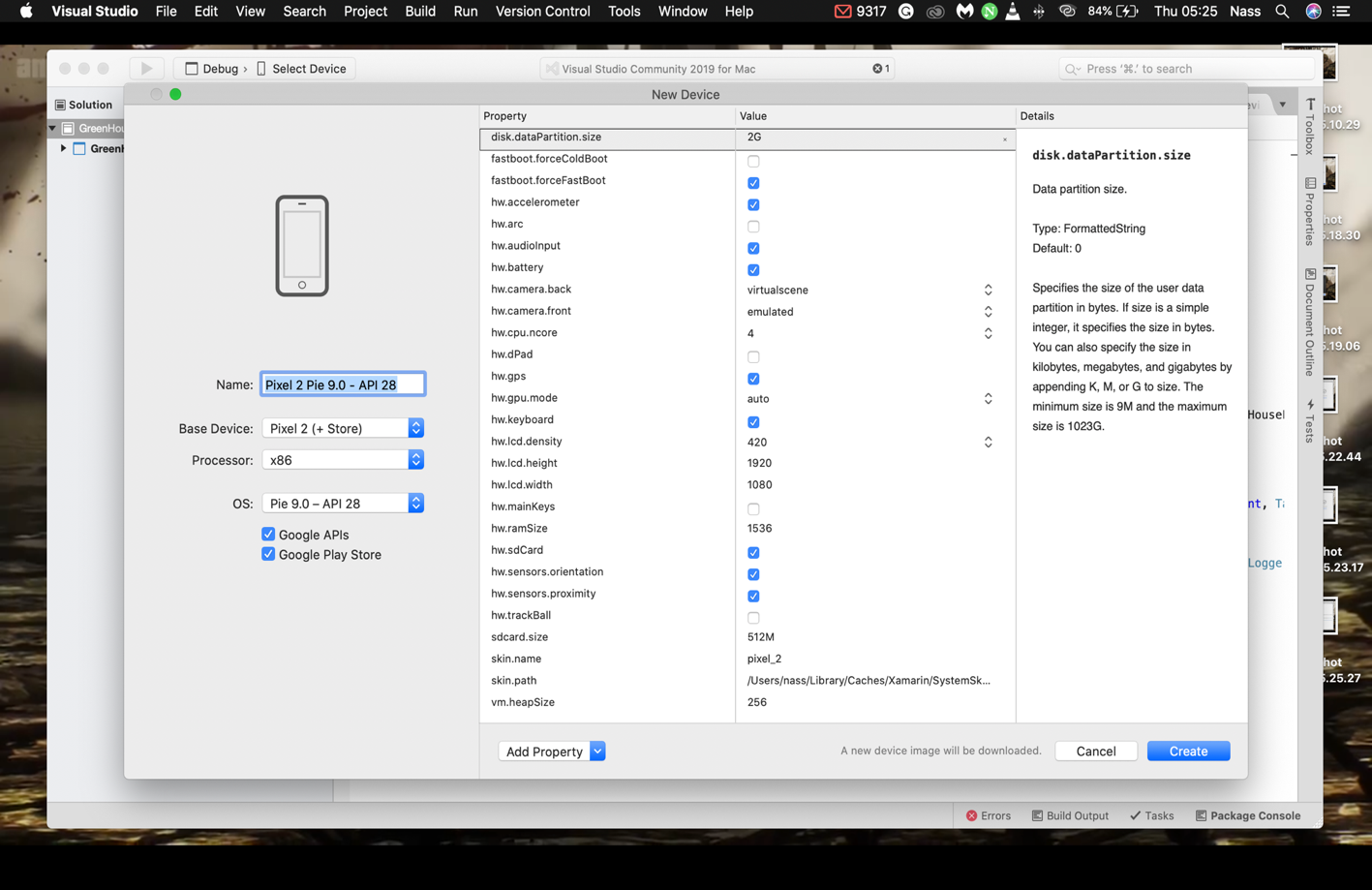
* 1. Mobile Application Development

To ease the process of retrieving the data stored on the data logger, an android application was developed called the “GreenHouse Logger”. The mobile application development process was done using Xamarin.Forms. Xamarin.Forms is a feature of Xamarin, the popular mobile development framework that extends the .NET developer platform with tools and libraries for building mobile apps. Xamarin.Forms uses MVVM. MVVM is a design technique employed to divide the user-interface (view), data (model), and application logic (view model).

* + 1. Deployment and testing

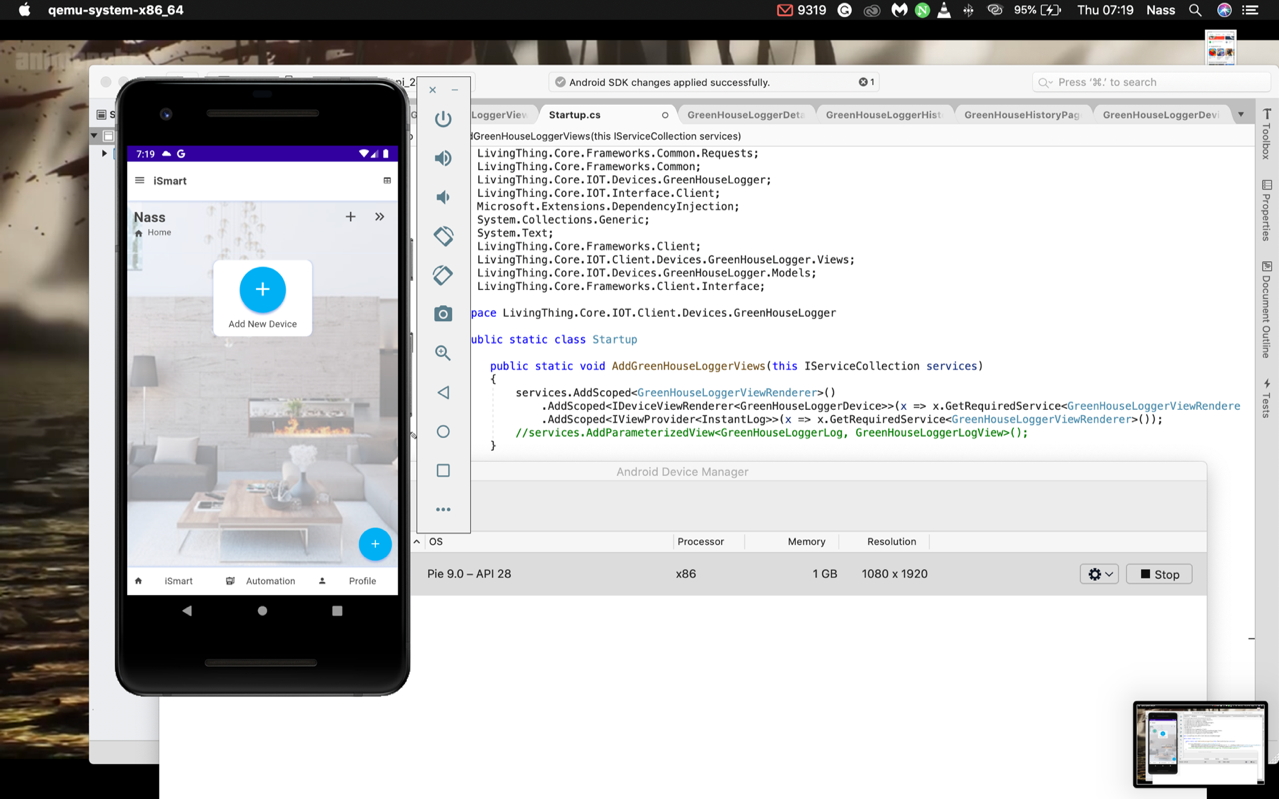
To debug the mobile application we configured Visual Studio to use a virtual machine called an android emulator to run the mobile application.

To configure a new android emulator for testing and the debugging , the following steps were followed. I clicked on the "Select Device Manager" in the menu bar of VS code. I was prompted to verify If I would like to create an emulator. I then selected “Add a Virtual Device”; This opens the Android Device Manager. Select + New Device to start the creation process. The options are automatically populated for a base emulator. If required, change any options and then selected “Create”.



*Figure 2.3 :Screenshot illustrating the emulator creation process*

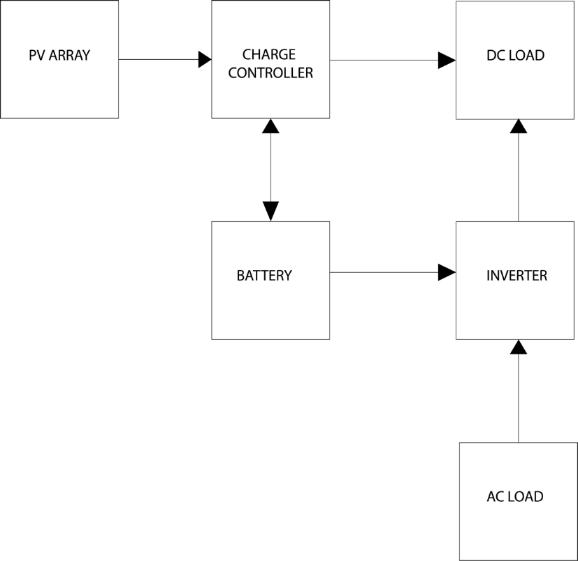
* + 1. Deploying and testing the app

Once the emulator has been created, I clicked “Play” and the Android emulator launched. From the menu, I selected Run > Start Debugging. The application was then built, deployed, and ran on the emulator.

*Figure 2. 4: Screenshot illustrating the debugging process*

2.3 Integrating the Data Logger with a PV system

The data logger was configured as a charge controller for a PV system by setting various parameters such as the High Voltage and Low Voltage disconnects, the float voltage and Discharge reconnect. The configuration process and Initial connection with the Solar Inverter were carried out by an solar installation expert following the block diagram in the figure 2.5 below.



*Figure 2. 5: Block Diagram of the Data Logger Integrated with a Solar Inverter*

2.4 Data Gathering

Data collection was the first step in preparing the ANN Model's input data. Hence, the data logger we built was used to collect data from three homes with Solar Inverter installation in Ado-Ekiti. A total of *1770* samples of data were collected from three homes between April 1st and April 20th, 2021. The PV systems in two of those homes had similar specifications to the inverter we proposed to develop, hence only the data from these houses were used for training the model. The components of the three PV systems and their specifications are listed in Table 2.1

From the data collected, the efficiency was calculated using the equation shown below:

(1)

Where Inverter power = (2)

Total DC power = (3)

Solar Power = (4)

Battery Current = (5)

Battery Power = (6)

Table 2. 1: The components of the three PV systems and their specifications

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| House | Solar Panel Rated Power | Solar Panel Type | Battery Capacity | Inverter Rating | Inverter Voltage | Inverter  Type |
| 1. | 150 W | Monocrystalline | 150 Ah | 2.5KVA | 12 V | PSW |
| 2. | 150 W | Monocrystalline | 100 Ah | 2.5KVA | 12 V | PSW |
| 3. | 100 W | Monocrystalline | 40 Ah | 1.5KVA | 12 V | PSW |

* 1. Data cleaning and pre-processing

Data cleaning is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect,inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. New columns containing the output power, input power, and power efficiency were added, whereas columns containing data not relevant to the model were removed. The Data cleaning process for this project was done using the pandas library in python .

Pre-processing data is also known as normalization, it makes the training process less sensitive to the scale of the features. This results in getting better coefficients after training, the process of making features more suitable for training by rescaling is called feature scaling.[15] The normalization for this project was done before the training process using the sci-kit-learn-preprocessing module.

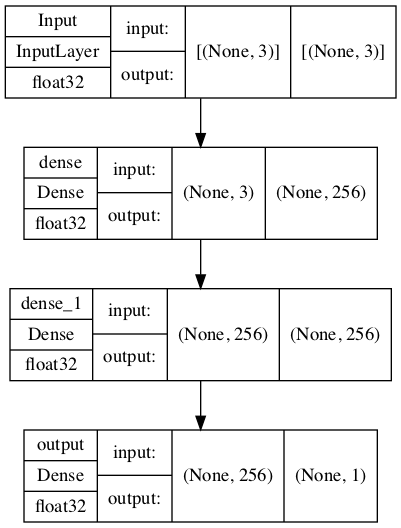
2.6 Prediction Model Development

For this study, an ANN-based technique for prediction the power production efficiency of PV systems was built. We employed a feed-forward backpropagation (FFBP) neural network to estimate the power output of the PV system directly by reference to the historical data of the PV system.

2.6.1 Creating the network

Creating a network involves defining network parameters , such as the number of neurons for the hidden layer, transfer function, training function, weight and bias learning function, and performance functions. During the training phase, parameters such the number of training epochs, learning rate, and the number of hidden layers could be modified depending on the error performance of the ANN output.

**2.6.2 Flowchart and architecture of the model**



*Figure 2.6 : a). Flowchart of the ANN Model. b) Architecture of the ANN Model*

2.6.3 Input layer

From observations, the power output or production of the PV system is zero from 18:45-to 05:59, hence the data recorded for the study period was trimmed to data samples from 07:30 to 17:30. The parameters selected for the Input Layer are the Solar Power, Battery Power and Load Power. Only the first 15 rows were selected, these rows are displayed in Table 2.2 below.

Table 2. 2:Input Data for Prediction Model

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Solar Power** | **Load Power** | **Battery Power** |
|  | 13.8539557 | 39.9821701 | 27.4842717 |
|  | 14.7519102 | 40.8178368 | 27.5015241 |
|  | 15.171035 | 41.1846886 | 27.4952736 |
|  | 17.5626571 | 37.4239883 | 23.7138299 |
|  | 4.09354767 | 43.1012188 | 39.4104065 |
|  | 18.6095287 | 42.4564195 | 25.6173049 |
|  | 19.6624259 | 42.6975097 | 24.8718902 |
|  | 19.5048193 | 41.958136 | 24.2259722 |
|  | 18.8551378 | 31.9870434 | 14.8446021 |
|  | 19.0775554 | 40.3541635 | 23.0242962 |

2.6.4 Hidden layer

The network has 2 hidden layers with 256 hidden neurons in the feed-forward neural network. A trial-and-error strategy was utilized to identify the proper number of hidden neurons for this model. The first hidden layer, the dense\_1 consists of 256 neurons and the ‘relu’ activation function. The second hidden layer, dense\_2 consists of 512 neurons and the ‘relu’ activation function.

2.6.5 Output layer

The output layer has just one parameter, which is the power efficiency. Only the first 15 rows of the training data were included in Table 2.3 below. The output layer contains the training data for the parameter we are trying to predict.

Table 2. 3: Expected Output Data

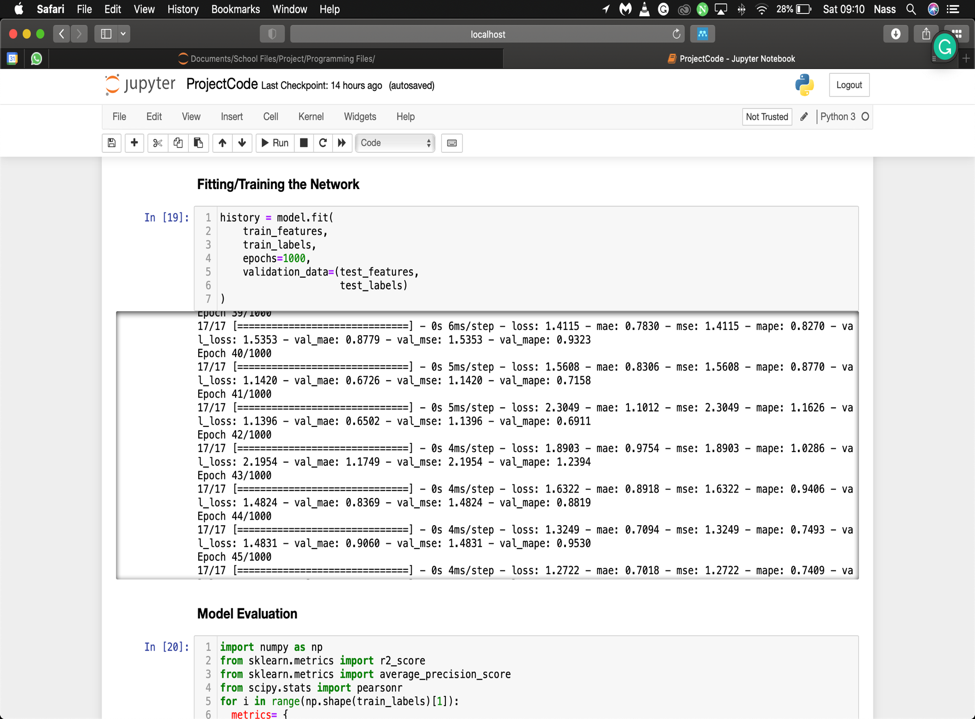
|  |  |
| --- | --- |
| NO | Power Efficiency |
|  | 92.117036 |
|  | 96.3517894 |
|  | 97.3643506 |
|  | 97.1373593 |
|  | 94.9177752 |
|  | 95.9129313 |
|  | 92.3611008 |
|  | 96.8140107 |
|  | 99.2176453 |
|  | 99.3245579 |

2.6.5 Training the network

After data normalization, the next step was to train the input data using the Adam optimizer. The Adaptive Moment Estimation (Adam) Adam optimizer involves a combination of two gradient descent methodologies:

* Momentum: This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the ‘exponentially weighted average’ of the gradients.
* Root Mean Square Propagation (RMSP): Root means square prop or RMSprop, it is an adaptive learning algorithm that takes the ‘exponential moving average’.[16]

Before the training process, the weight and bias values were set into initial values. Then 702 samples of data will be randomly divided by the model’s training algorithm into training, testing and validation data. The algorithm selected the data randomly from the data set during the training process, so every instance of the training process will get a varied MSE value that depends on 80% of the input data chosen. The code illustrated in Figure 3.6.3 below was used to Initiate the Network Training Process.



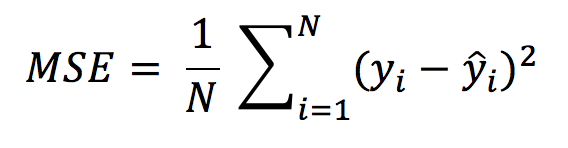
*Figure 2. 7: Screenshot illustrating the code to initiate network training process*

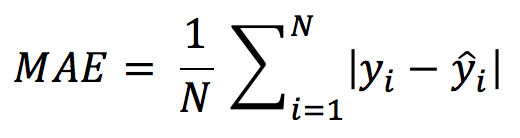
2.6.7 Assessing the Performance of the Prediction model

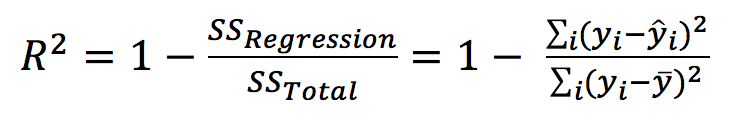
Then, the next stage was analyzing the performance of the constructed ANN Model. To analyze the ANN model employed in this research, we took use of numerous error statistics. The mean squared error (MSE) informs you how near a regression line is to a set of points. MSE is determined by the sum of the square of prediction error which is real output minus predicted output and then divide by the number of data points. It offers you an absolute number on how much your expected outcomes depart from the actual amount. You cannot understand numerous insights from one single result but it gives you an actual number to compare against other model results and help you select the best regression model. In the context of machine learning, mean absolute error (MAE), refers to the extent of difference between the prediction of observation and the true value of that observation.[17]

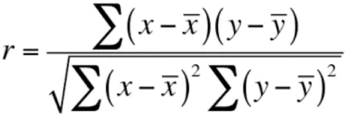
R Square is derived by the sum of the squared prediction error divided by the square's entire sum, which substitutes the calculated forecast with the mean. R Square value is between 0 to 1 and a greater value implies a better fit between prediction and actual value. R Square is better suited to communicate the model to other people because you can explain the number as a proportion of the output variability. [17]

All the errors formula expressed in a percentage as defined as follow:

 (2.1)

 (2.2)

 (2.3)

 (2.4)

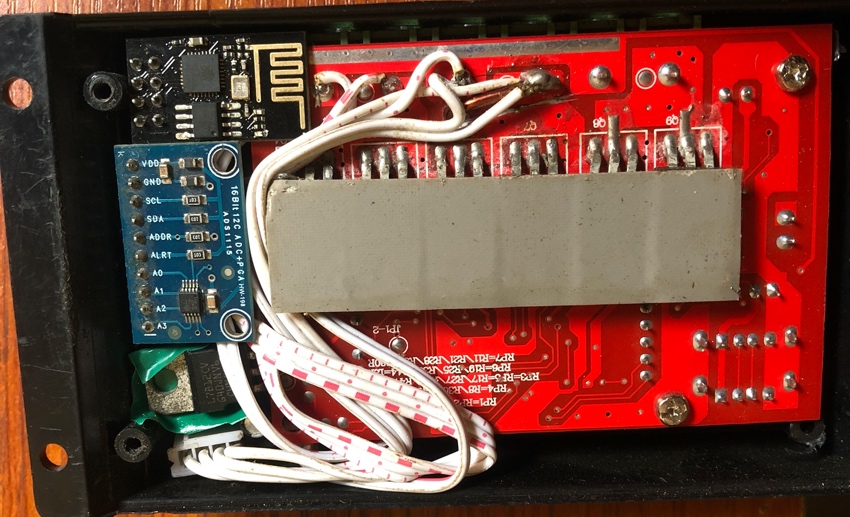
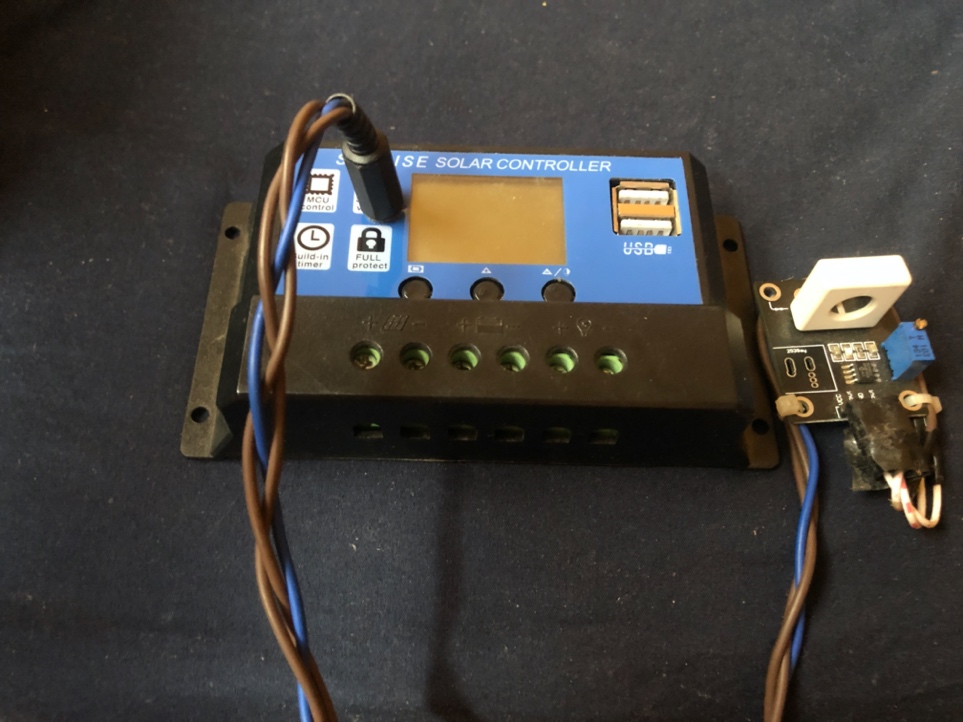
1. Results and Discussion

In this section, the results of the study are presented and discussed with reference to the aim and objectives of the project, which was the design and implementation of a smart data logger and prediction of the power efficiency of a PV system.

* 1. **Datalogger**

Owing to its high computation performances and simple programming, the ESP8226 controller was chosen to acquire, process, and transmit the gathered data in real-time. Besides its high calculation capabilities, this controller incorporates a built-in Wi-Fi technology, which means that our application will not require additional ethernet or Wi-Fi shields to ensure network connection and the transmission of data.

The crucial data, such as related PV panel and battery system voltages, current, state of charge of batteries, etc., are collected and stored are stored in the NVRAM. Data stored in the NVRAM is transmitted from the NodeMCU synchronised with the mobile application which is connected through WiFi. This allows the user to easily choose what kind of data and which period he wishes to see. For this project, a mobile application named “GreenHouseLogger” was developed as shown in Figure 3.3

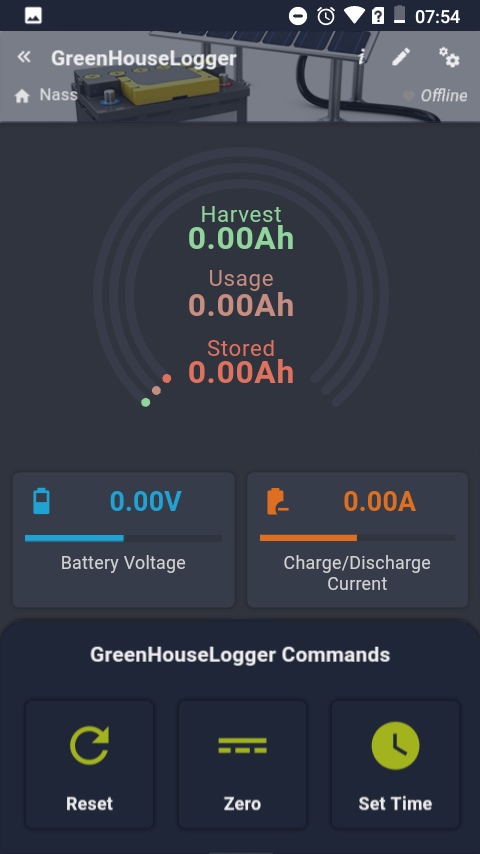
 

***Figure 3.1*** *(****a****) is an image of the internal part of the Data Logger device (****b****) is an image of the external part of the Data Logger device*

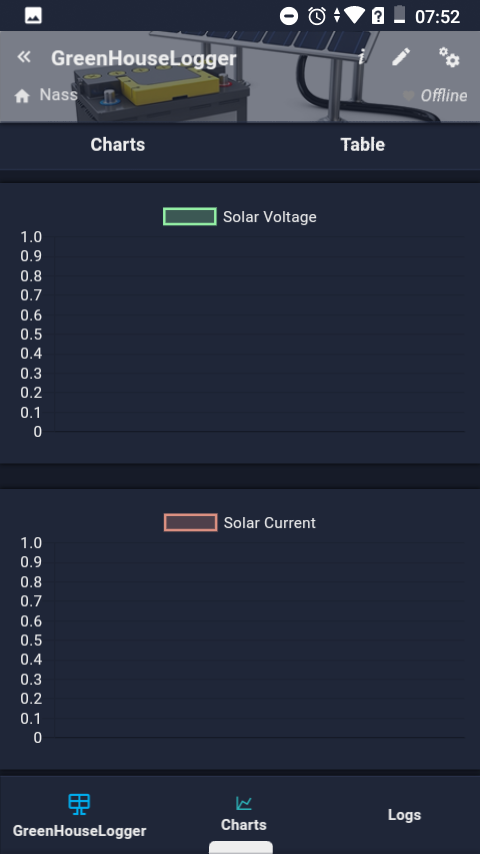
* 1. **MOBILE APPLICATION**

In summary, the proposed IoT application can continuously supervise and maintain the PV system in a safe state and high working performance. The crucial data, such as related PV panel and battery system voltages, current, state of charge of batteries, etc., are logged and transmitted from the data logger synchronised with the mobile application which is connected through WiFi.

The data may be presented on the main dashboard as the device’s state as shown in figure 4.2, showing several parameters or it may be viewed as charts or as a table as shown in the figure 4.3 below. This allows the user to easily choose what kind of data , which period and the format he/she wishes to see. The screenshot in figure 4.2 also shows the commands that can be executed on the data logger device through the mobile application.



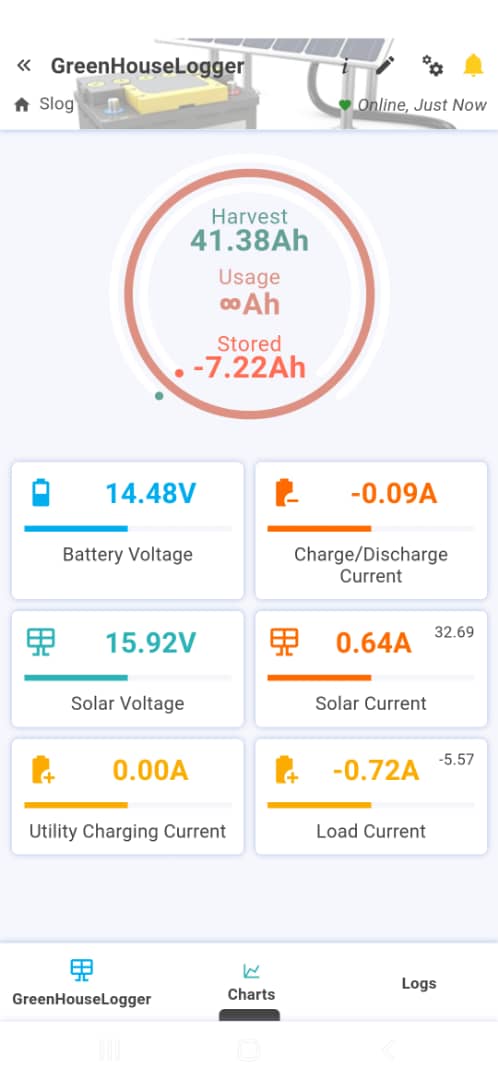
**Figure 3.2: Screenshot of Greenhouse Data Logger Dashboard**

****

**Figure 3.3: Screenshot of Greenhouse Data Logger Data View Page**

**3.3 Integration with Mobile Application**

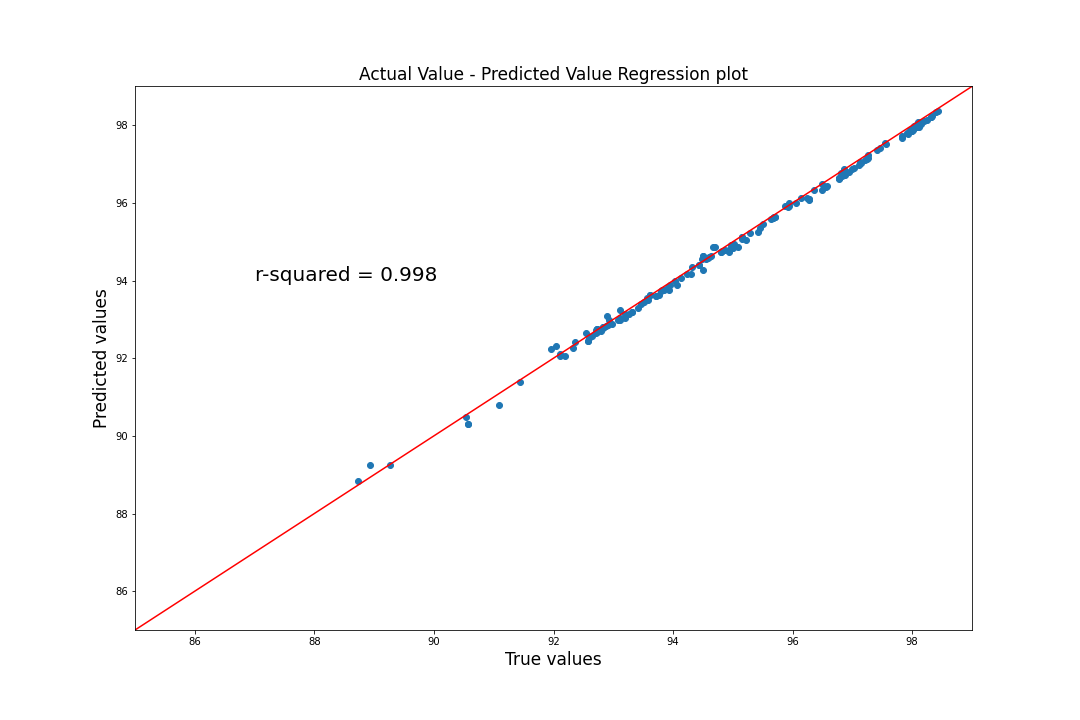
After installing the application on the mobile phone, the phone is connected to the data logger’s WiFi. This connection enables the mobile application to display a visualized format of the data recorded by the data logger hardware without a wired connection. The results show a mobile application was successfully developed to perform the desired function, figure 4.4 below illustrates the dashboard view of the data logger device. The dashboard view shows information about the PV system’s Total Harvested battery power , the amount of which has been used and that being stored. It also gives the voltage , current and charging/discharge current of the solar battery , solar module and inverter respectively.

****

**Figure 3..4: Mobile Application Interface After Integration with Solar Inverter**

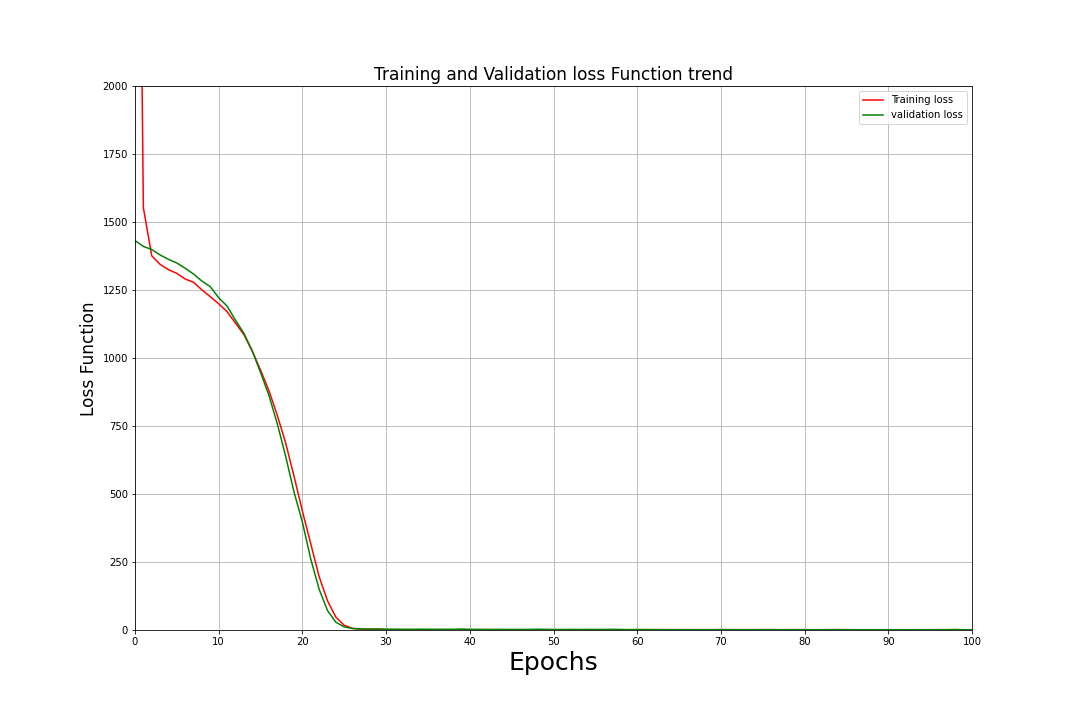
**3.4 Prediction Model**

When assessing the performance of the model, we produced a regression plot of the actual and predicted values, with the R-squared score of the model as illustrated in Figure 4.5. We observed that, the data points are concentrated on a straight line. While the best fit for testing was approaching 1 means a positive linear relationship . It indicates that our model is performing well. From my research on prediction analysis, stronger relationships produce a tighter clustering of data points.



**Figure 3. 5: A regression plot of the actual and predicted values**

The graph in Figure 4.6 illustrates the training and validation loss function trend. From the graph, we notice that the model’s loss stabilizes after the first 30 epochs.This is also another indication that the model’s performance was very good.



**Figure 3.6: Loss Function Trend**

4.1 Conclusion

This project explored the tools and techniques used in the monitoring systems of existing PV systems. A prototype was developed by my project partner to demonstrate a system for monitoring PV systems to provide real-time data to the consumer. The system design features an out-of-the-box hardware setup and easy IoT synchronization, this flexibility makes the system more suitable for applications in remote areas of developing countries.

The main achievement was the accuracy of the developed predictive model of this project. Accurate power planning requires power prediction for PV modules. In this study, the energy efficiency of solar panels was estimated using mathematical formulas. Then we trained the FFBP network with the estimated performance. The trained model showed excellent performance. The performance of the ANN model was evaluated using MSE, MAE, R-squared score and Pearson correlation coefficient which presented the accuracy and efficiency of the system itself.

Using this neural network method makes it easy and fast to accurately predict the power output efficiency of a PV system and solve the stated problem. The significance of this study is for PV panel system owners to control, estimate and plan their system usage and maintenance. We can conclude that the aim of the project was achieved.

**4.2 Limitations of Project**

This project has some significant success claims and some recognized limitations such as the issues of the mobile application is only available for Android phones; this limitation exists because iOS devices do not allow the installation of applications that are not available on the app store.

**4.3 Recommendation**

Observations from this project shed light on how key parameters could affect the efficiency of the forecasts made by a prediction model, it is to be taken into consideration the number of data samples used in training the model and the weather conditions in which the samples were taken. In that case, it is recommended that the data logger system be modified to capture solar irradiance and climate data. With this supplementary data, the model can be trained to accurately predict the power efficiency of the PV system on days with different weather conditions. We also suggest that tests for inverter efficiency should be conducted with properly sized arrays to ensure accurate computations.

**Acknowledgments:** To my project to our supervisor, Engr. Dr S. L. Gbadamosi , we would like to thank you for your patience, insights and guidance during the whole process of making this project a success. We have benefited greatly from your wealth of knowledge.

**Conflicts of Interest:** The authors declare no conflict of interest.

References

[1] “Internet of things - Wikipedia.” https://en.wikipedia.org/wiki/Internet\_of\_things (accessed Mar. 14, 2022).

[2] Y. Cheddadi, H. Cheddadi, F. Cheddadi, F. Errahimi, and N. Es-sbai, “Design and implementation of an intelligent low-cost IoT solution for energy monitoring of photovoltaic stations,” *SN Applied Sciences*, vol. 2, no. 7, Jul. 2020, doi: 10.1007/s42452-020-2997-4.

[3] “What Is The Internet of Things? A Complete Guide To IoT. | Built In.” https://builtin.com/internet-things (accessed Mar. 04, 2022).

[4] S. R. Madeti and S. N. Singh, “Monitoring system for photovoltaic plants: A review,” *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 1180–1207, Jan. 2017, doi: 10.1016/J.RSER.2016.09.088.

[5] R. Mukaro, X. F. Carelse, and L. Olumekor, “First performance analysis of a silicon-cell microcontroller-based solar radiation monitoring system,” *Solar Energy*, vol. 63, no. 5, pp. 313–321, Nov. 1998, doi: 10.1016/S0038-092X(98)00072-3.

[6] C. Kamunda, X. F. Carelse, M. Mathuthu, and A. Makarau, “Design and construction of microclimate monitoring system,” *Review of Scientific Instruments*, vol. 78, no. 8, 2007, doi: 10.1063/1.2766827.

[7] M. Benghanem, “Low cost management for photovoltaic systems in isolated site with new IV characterization model proposed,” *Energy Conversion and Management*, vol. 50, no. 3, pp. 748–755, Mar. 2009, doi: 10.1016/J.ENCONMAN.2008.09.048.

[8] M. Benghanem, A. H. Arab, and K. Mukadam, “Data acquisition system for photovoltaic water pumps,” *Renewable Energy*, vol. 17, no. 3, pp. 385–396, Jul. 1999, doi: 10.1016/S0960-1481(98)00126-8.

[9] M. Benghanem and A. Maafi, “Performance of stand-alone photovoltaic systems using measured meteorological data for Algiers,” *Renewable Energy*, vol. 13, no. 4, pp. 495–504, Apr. 1998, doi: 10.1016/S0960-1481(98)00013-5.

[10] M. Benghanem, “Measurement of meteorological data based on wireless data acquisition system monitoring,” *Applied Energy*, vol. 86, no. 12, pp. 2651–2660, Dec. 2009, doi: 10.1016/J.APENERGY.2009.03.026.

[11] C. Jyosthna *et al.*, “ANN Approach for Weather Prediction using Back Propagation,” *International Journal of Engineering Trends and Technology*, 2012, Accessed: Apr. 21, 2022. [Online]. Available: http://www.internationaljournalssrg.org

[12] K. Alawode, “A Comparison of Neural Network Models for Load Forecasting in Nigerian Power System Group Consensus Algorithms for Multi-agent Systems View project NSGA-III Reference Based approach for solving Multi-Objective Economic/Emission and Transmission Loss optimization. View project,” 2013. [Online]. Available: https://www.researchgate.net/publication/273575221

[13] E. Lorenz, J. Hurka, D. Heinemann, and H. G. Beyer, “Irradiance Forecasting for the Power Prediction of Grid-Connected Photovoltaic Systems,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 2, no. 1, pp. 2–10, 2009, doi: 10.1109/JSTARS.2009.2020300.

[14] A. Yona, T. Senjyu, A. Y. Saber, T. Funabashi, H. Sekine, and C. H. Kim, “Application of neural network to one-day-ahead 24 hours generating power forecasting for photovoltaic system,” *2007 International Conference on Intelligent Systems Applications to Power Systems, ISAP*, 2007, doi: 10.1109/ISAP.2007.4441657.

[15] “2 Easy Ways to Normalize data in Python - JournalDev.” https://www.journaldev.com/45109/normalize-data-in-python (accessed Mar. 06, 2022).

[16] “Intuition of Adam Optimizer - GeeksforGeeks.” https://www.geeksforgeeks.org/intuition-of-adam-optimizer/ (accessed Mar. 14, 2022).

[17] “3 Best metrics to evaluate Regression Model? | by Songhao Wu | Towards Data Science.” https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b (accessed May 07, 2022).