**Materials and methodology**

In this chapter, the description of the experimental set-up and components, used to realise the hardware for IOT based smart weather monitoring device, is established. Detailed descriptions of the components used along with their electrical parameters are stated. The device has a capability of feeding temperature and humidity data to an online server. These data are recorded at an interval of one twenty seconds and can be accessed anywhere by logging into the server through an account. The hardware can be directly connected to mains but also has a dedicated rechargeable lithium-ion battery system to operate in remote conditions in absence of a power source. For the sensing part we have chosen the DHT22 digital temperature and humidity sensor. The NodeMCU ESP8266 v3 microcontroller board is chosen for processing and uploading data to the server. TP4056 battery management module for charging the li-ion battery and AC to DC converter module for power supplying purposes.

**Basic Block diagram**

ESP8266

Microcontroller

Display

Temperature and Humidity sensor

Battery management system

Li ion Battery

AC/DC

Converter

Fig: Block diagram of the IOT based smart weather monitoring device.

The hardware basically comprises of six components, the microcontroller, display, sensor, AC to DC converter, Battery management system and a lithium-ion battery. The components used are listed in table (give number). Basic functioning and electrical parameters of all the components are discussed in the next sections.

Table (giv no): List of all components used along with their specifications and price.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl no | Name of component | Functioning | Price (INR) | Remarks |
| 1 | NodeMCU ESP8266 board | Microcontroller | 450 | Board with Integrated WiFi used for data acquisition and uploading |
| 2 | DHT 22 | Temperature and Humidity sensor | 388 | Digital sensor for acquiring temperature and humidity data |
| 3 | SSD1306 | I2C Oled Display | 325 | On board instantaneous data display |
| 4 | ICR18650-26F | Lithium-ion battery | 350 | Used for stand-alone operations |
| 5 | TP 4056 | Battery Management System | 148 | To charge the li-ion battery |
| 6 | AC-DC converter | AC-DC converter | 110 | To work as a power source |

**1. Microcontroller**

For the microcontroller, we have chosen the ESP8266 development board because of its inbuilt Wi-Fi capabilities. The board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor is capable of running at 80MHz to 160 MHz clock frequency and supports RTOS. The flash memory of the board is 4MB which can store programs and it has a RAM of 128 KB. Its high processing power with in-built Wi-Fi and deep sleep operating features make it ideal for IoT projects. The board has sixteen general purpose input output pins (GPIOs), with nine digital pins (D0 to D9) and one analog pin (A0) with an inbuilt 10-bit internal analog to digital converter as shown in fig 1.1. The board also supports UART, SPI, and I2C interface [1].



Fig 1.1: Basic pinout diagram of ESP8266 development board []

The operating voltage of the microcontroller is between 3.0 to 3.6V however it can be powered through its input pins (Vin and G) by applying a voltage anywhere between 5 to 12 volts. The pins D1 and D2 can be used for I2C connections where the D1 acts as Serial Data Line (SDA) and D2 acts as Serial Clock (SCL). The power consumption, if operated in 3.3V is between 185mW to 561 mW. The board supports 802.11 b/g/n Wi-Fi protocol with a frequency range within 2.4 to 2.5 GHz [2][2].

**2. Sensor**

For sensing temperature and humidity, we have used the DHT 22 sensing module. The sensor is polymer capacitor based has a working voltage of 3.3 to 6 volts. It has a temperature sensing range of 40 to 80 and relative humidity sensing range of 0 to 100%. Table 2.1 lists the technical specifications of the sensor.

Table 2.1: Specifications of DHT 22 sensor [3].

|  |  |
| --- | --- |
| Model | DHT22 |
| Power supply | 3.3-6V DC |
| Output signal | Digital signal via single-bus |
| Sensing element | Polymer capacitor |
| Operating range | Humidity: 0-100%RH  Temperature: 40-80 |
| Accuracy | Humidity ± 2 % RH (Max ± 5 % RH);  Temperature < ± 0.5Celsius |
| Resolution or sensitivity | Humidity 0.1 % RH  Temperature 0.1 |
| Repeatability | Humidity ±1 % RH  Temperature ±0.2 |
| Sensing period | 2 sec |

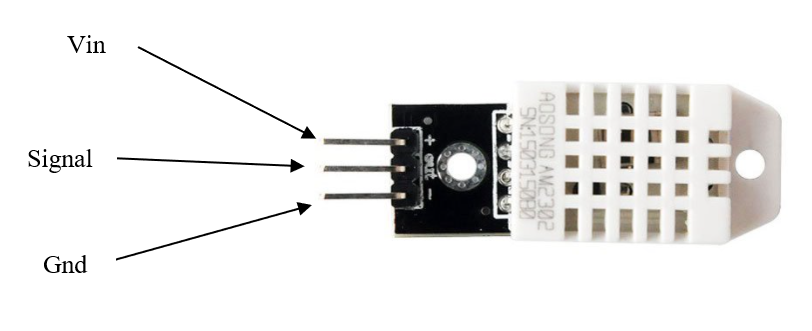


Fig 2.1: DHT 22 module with pinout configurations [3]

The sensor utilizes digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer. Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory. Single-bus data is used for communication between MCU and DHT22, it costs 5mS for single time communication. Data is comprised of integral and decimal parts. DHT22 send out higher data bit first, 8-bit integral relative humidity data then 8-bit decimal humidity data. The same is done for temperature data. It also includes 8-bit check-sum. If the data transmission is right, check-sum should be the last 8-bit of 8-bit integral relative humidity data, 8-bit decimal humidity data, 8-bit integral temperature data and 8-bit decimal temperature data. When microcontroller unit send start signal, DHT22 change from low-power-consumption-mode to running-mode. When microcontroller unit finishes sending the start signal, DHT22 will send response signal of 40-bit data that reflect the relative humidity. Small size & low consumption & long transmission distance (20m) enables DHT22 to be suited in all kinds of harsh application occasions [3].

**3. Display**

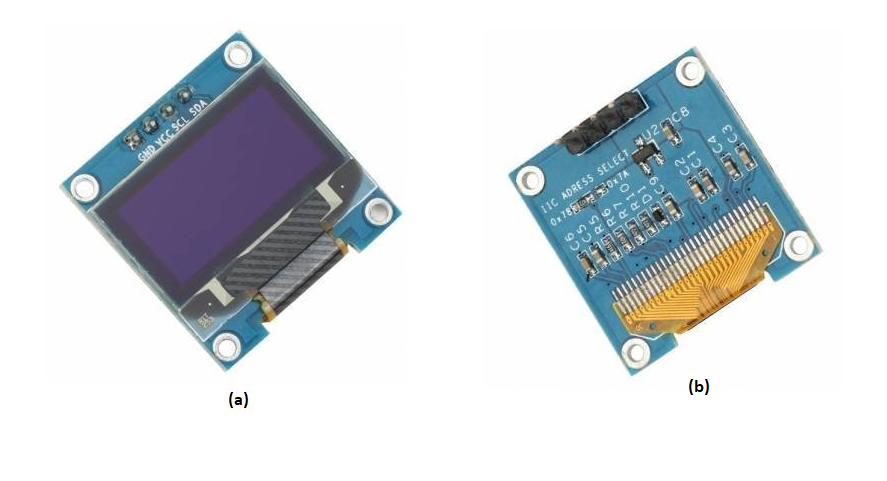
****

Fig 3.1: Image of I2C oled display (a) top view (b) bottom view [4].

For instantaneous on-board display of data, an I2C OLED display has been utilised. The display module consists of four pins, SCL, SDA, Vin and Gnd. The SDA and SCL pins are connected to the D1 and D2 pins of the Node MCU microcontroller board. The display is dot-matrix type with a resolution of 128 × 64 bits. It operates under a voltage range of 1.65V to 3.3V. The module comes with on-board SD1306Z oled display driver.

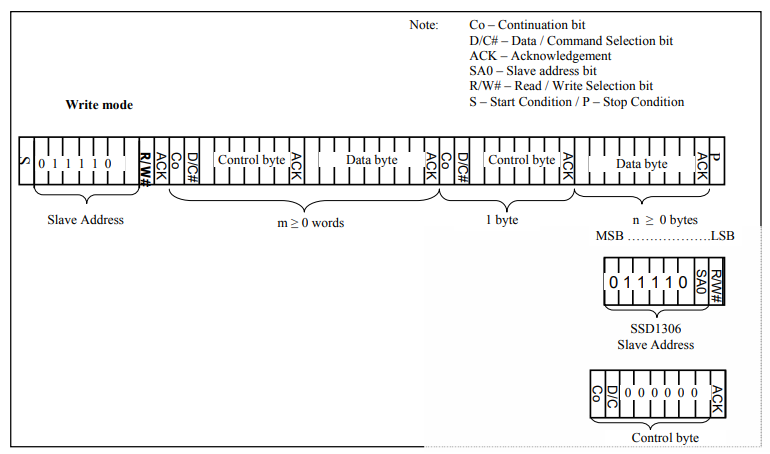


Fig 3.2: I2C-bus data format for oled display [4].

The I2C communication protocol uses only two lines to transmit data. At the starting condition of data transfer, the SDA line switches from a high voltage level to a low voltage level before the SCL line switches from high to low. Here the microcontroller acts as master and the display works as the slave device. The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level before switching the SCL line from high to low as shown in Fig 3.3.

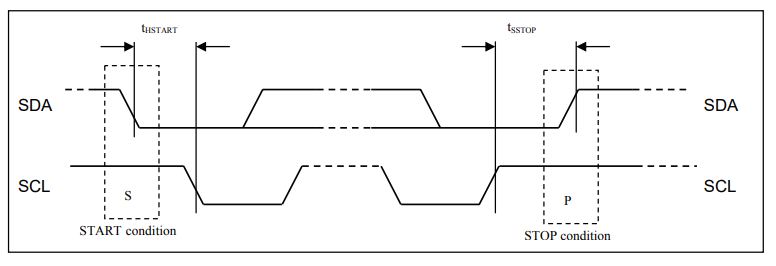


Fig 3.3: Definition of the Start and Stop Condition [4].

The master i.e., the microcontroller, sends the slave (i.e., display) 7-bit address of the slave it wants to communicate with, along with the read/write bit. The slave compares the address sent from the master to its own address. If the address matches, the slave returns an acknowledgement bit by pulling the SDA line low for one bit as shown in the Fig 3.4. If the address from the master does not match the slave’s own address, the slave leaves the SDA line high. The master sends or receives the data frame.

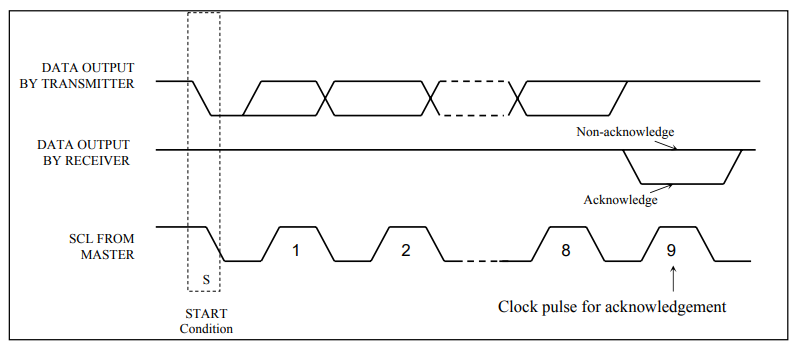


Fig 3.4: Definition of the acknowledgement condition [4].

After each data frame has been transferred, the receiving device returns another acknowledgement bit to the sender to acknowledge successful receipt of the frame. To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high as shown in Fig 3.5.

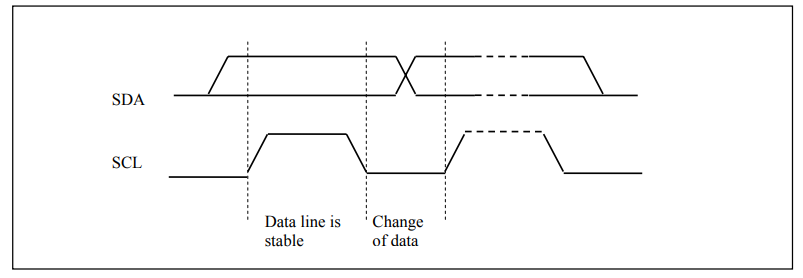


Fig 3.5: Definition of the data transfer condition [4].

**4. Battery**



Fig 4.1: Image of the 18/65 ICR battery

ICR18650-26F li-ion battery is used in this project to enhance the remote operating capabilities of the smart device. The battery has a nominal voltage of 3.7V and has a capacity of 2600 maH. The operating temperature is between 20 to 60. Table 4.1. shows the specifications of the battery [5].

Table 4.1: Battery specifications [5].

|  |  |
| --- | --- |
| **Item** | **Specification** |
| Nominal Capacity | 2600mAh (0.2C, 2.75V discharge) |
| Charging Voltage | 4.2 ±0.05 V |
| Nominal Voltage | 3.7V |
| Charging Method | CC-CV  (Constant voltage with limited current) |
| Charging Current | Standard charge: 1300mA  Rapid charge: 2600mA |
| Charging Time | Standard charge: 3hours  Rapid charge: 2.5hours |
| Max. Charge Current | 2600mA |
| Max. Discharge Current | 5200mA |
| Discharge Cut-off Voltage | 2.75V |
| Operating Temperature | Charge: 0 to 45℃ Discharge: -20 to 60℃ |
| Storage Temperature | 1 year: -20~25℃ (1\*)  3 months: -20~45℃ (1\*)  1 month: -20~60℃ (1\*) |

**5. Battery Management System (BMS)**



Fig 5.1: Battery management system module.

The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. Furthermore, the TP4056 can work within USB and wall adapter. This module enables the charging of the li-ion battery when there is a power source and protects the battery from over discharge when there is no power. The charging state is indicated by a glowing red led. When the battery is fully charge, the module lights up a blue led. The input voltage to the module is 5V and the module charges the battery at constant 4.2V. The internal schematics of the module is given in Fig 5.2 [6].

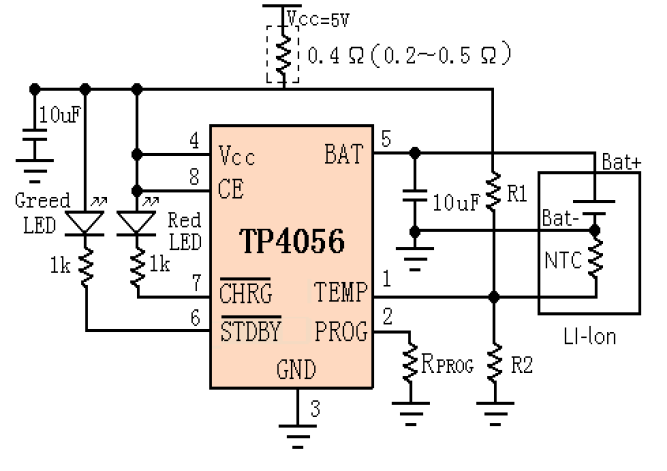


Fig 5.2: Schematics of TP4056 module [6].

**6. AC to DC converter**

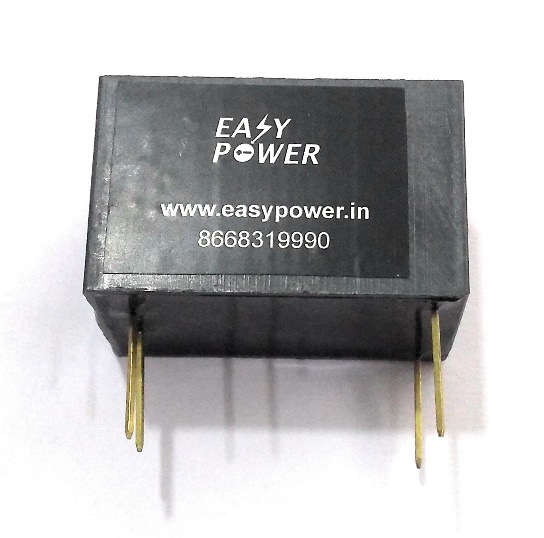


Fig 6.1: Image of AC to DC converter module.

This is a compact AC to DC converter which can convert 220 AC signal to 5V and 3.3V DC signal. It has a current range of 500 mA. This hardware is used to feed power to the IOT based smart weather monitoring device. Fig 6.2 shows the inner schematics of the device [7].

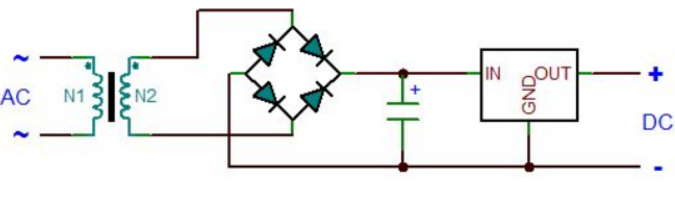


Fig 6.2: Schematics of AC to DC converter [7].

**Schematics and working of IOT based smart weather monitoring device**

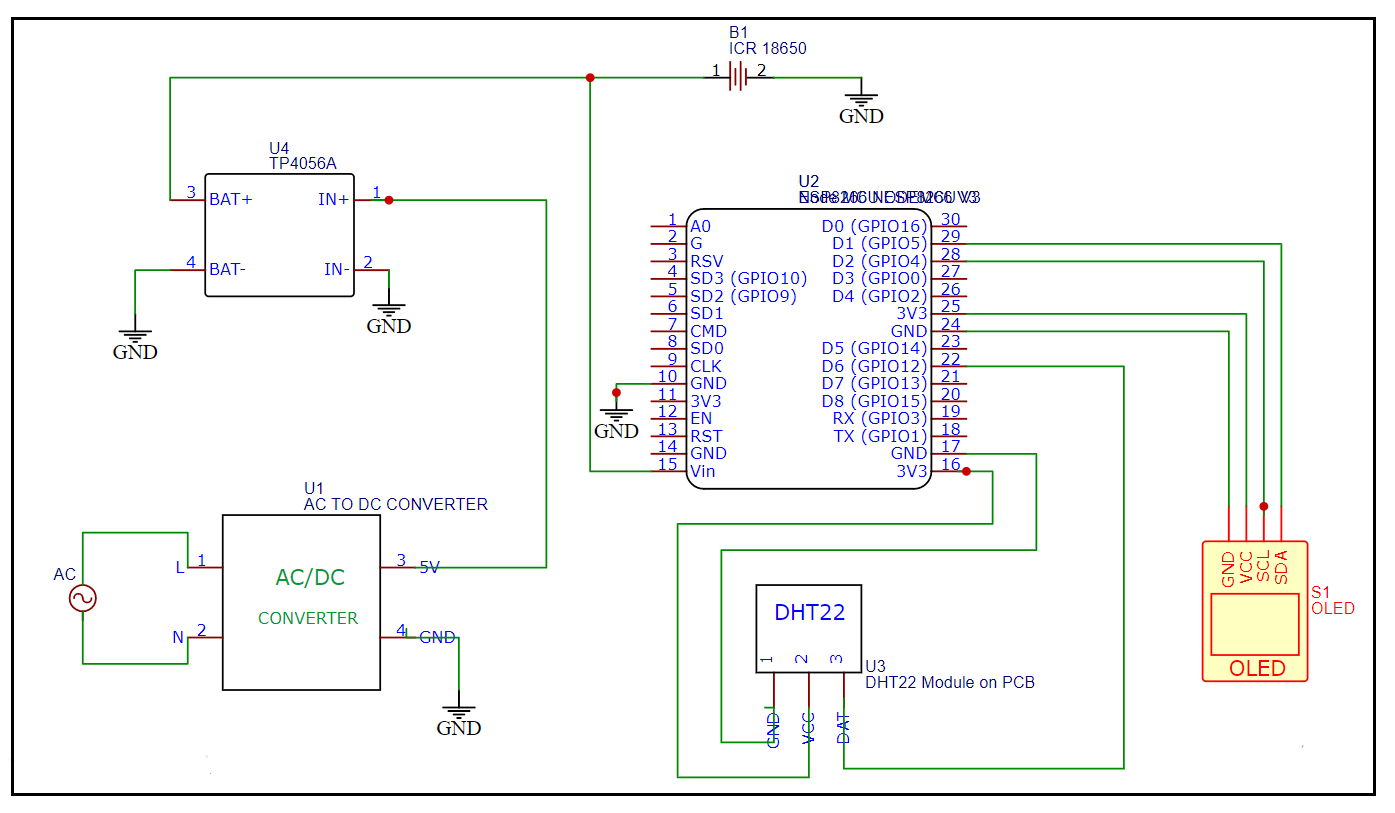
****

Fig : Working schematics of the weather monitoring device.

All the connections are made in accordance with the figure above. The digital pin of the DHT22 sensor is connected to digital pin D6 of the board. 3.3V (pin no. 25) and ground pins (pin no. 24) on the board are connected to the pins 1 and 3 respectively of the sensor. D1 (pin no. 29) and D2 (pin no. 28) pins of the board are connected to SDA and SCL pins of the display. Whereas 3.3V (pin no. 16) and ground (pin no. 17) pins are connected to the displays Vcc pin and ground pins. The output pins of the battery module BAT+ and BAT are connected to battery’s positive terminal and ground. The Vin pin (pin no. 15) is connected to battery’s positive terminal and GND pin (pin no. 10) is connected to ground. Upon powering up the whole setup with a 220V AC power supply, the device starts taking temperature and humidity data and display the data in the Oled screen. In presence of a Wi-Fi based internet connection, it sends the acquired data to “Thingspeak.com” server.

**Results and Discussion**

**7. Data Acquisition via “Thinkspeak.com” server**

ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyse live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics [8]. The figures below show the data acquisition and visualisation in the ThingSpeak™ server.

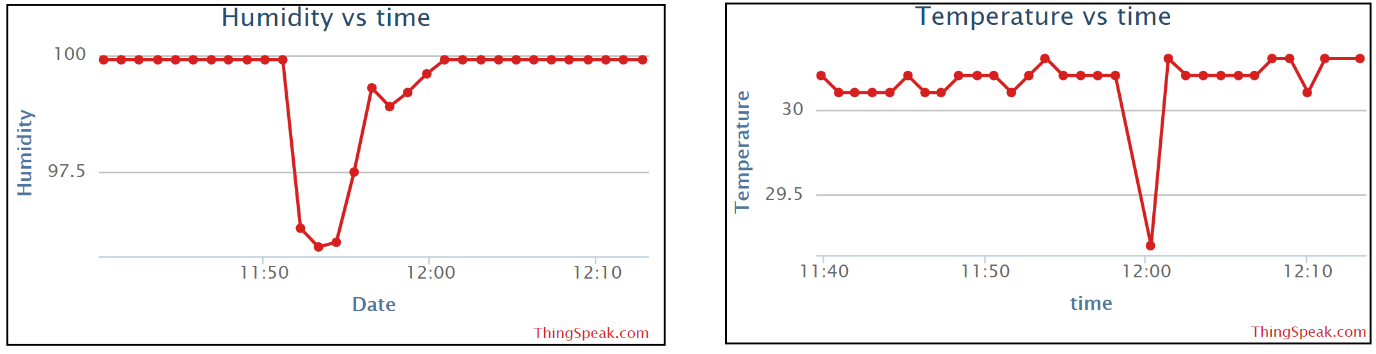


Fig 7.1: Data visualisation in “Thingspeak” server.

In order to record data in “ThingSpeak” server, firstly an account needs to be created in MATLAB website. After logging into the “ThingSpeak” server, a channel is needed to be created. After creation of the channel, different fields have to be created for different sets of data. In this case, two fields (i.e., humidity and temperature) are created. After creation of all the required fields, the channel will provide a “channel number” (1411111) and “API key” (R49YBNJI41BMLRK7) as shown in Fig 7.2. These two numbers are crucial in writing the program for ESP8266 board for proper recording of data in the “ThingSpeak” server. For a free account in ThingSpeak, the minimum time between consecutive data uploads is about fifteen seconds. This also has to be kept in mind while writing the program. We have put a 20s delay between each data uploads, therefore making the device recording data after every one minute.

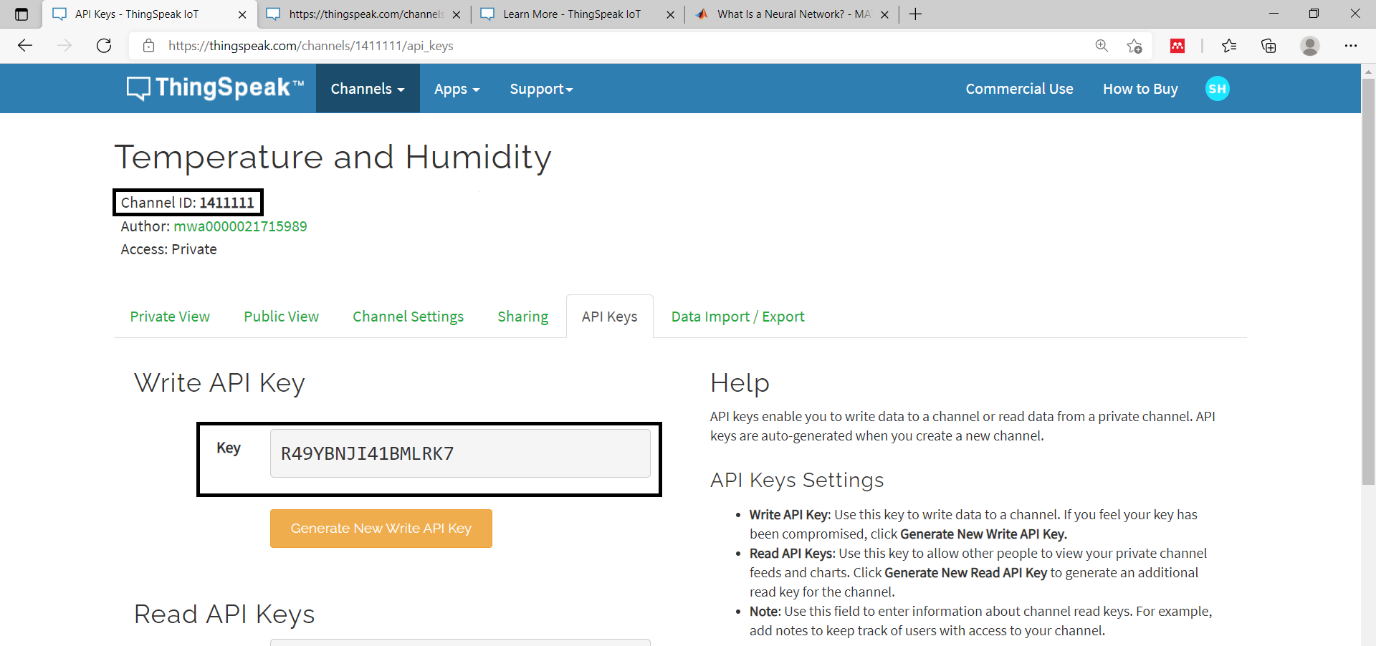


Fig 7.2: Image describing how to get API key and Channel number from “ThingSpeak”.

**8. Data processing and prediction using artificial neural network**

Processing of acquired data for prediction is done through the artificial neural network toolbox of MATLAB®. For this project the “dynamic time series” app present in the neural network toolbox on MATLAB (version R2019a) is used. Predictive models are used for system identification or dynamic modelling, in which the user can build dynamic models of physical systems. These dynamic models are important for analysis, simulation, monitoring and control of a variety of systems. In this project, the Nonlinear Autoregressive with External input (NARX) modelling is used for weather prediction. Fig 8.1 shows the neural network used in this prediction model.

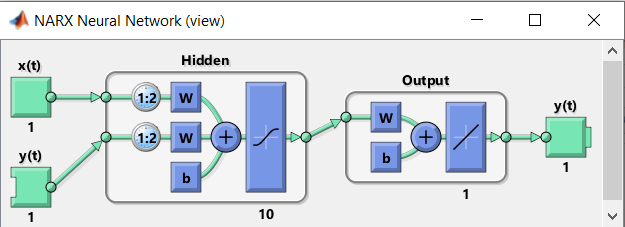


Fig 8.1: Diagram of the NARX model used.

Closed-loop networks can perform multistep predictions. When external feedback is missing, closed-loop networks can continue to predict by using internal feedback. In NARX prediction, the future values of a time series are predicted from past values of that series, the feedback input, and an external time series. We start the analysis by importing temperature, humidity and time data in the MATLAB environment to train the neural network with known values. The details of the program are described in Appendix A1. After loading all the values, we start the neural network app by typing “nntool” in MATLAB. Fig 8.2 shows the toolbox in MATLAB environment. Out of the three options as shown in Fig 8.2, we choose the first one as it is best suited for our purpose then click ‘next’ button.

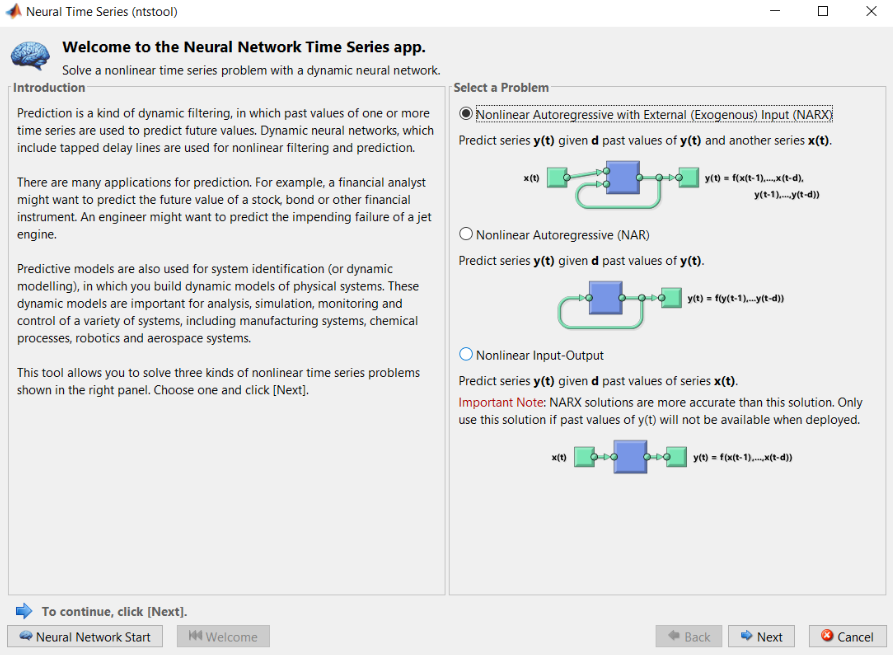


Fig 8.2: Image of neural network app in MATLAB.

After selection of the NARX predictive model, input data must be provided which in our case is date and output data i.e., humidity and temperature is selected. This is shown in Fig 8.3

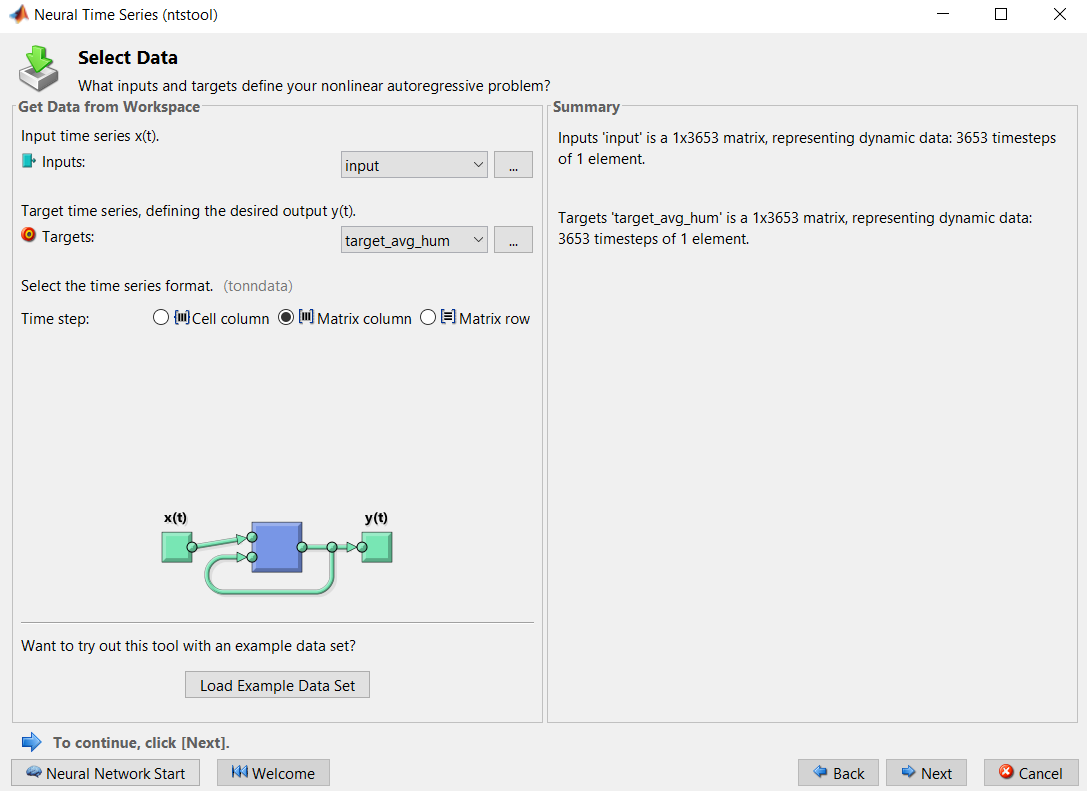


Fig 8.3: Initialization of input and output data of the neural network.

After initialization of data, parameters for testing and validation are to be selected. By default, the toolbox sets 70% of data for training the neural network, 15% for validation and another 15% for testing the efficiency of the neural network. This is shown in Fig 8.4.

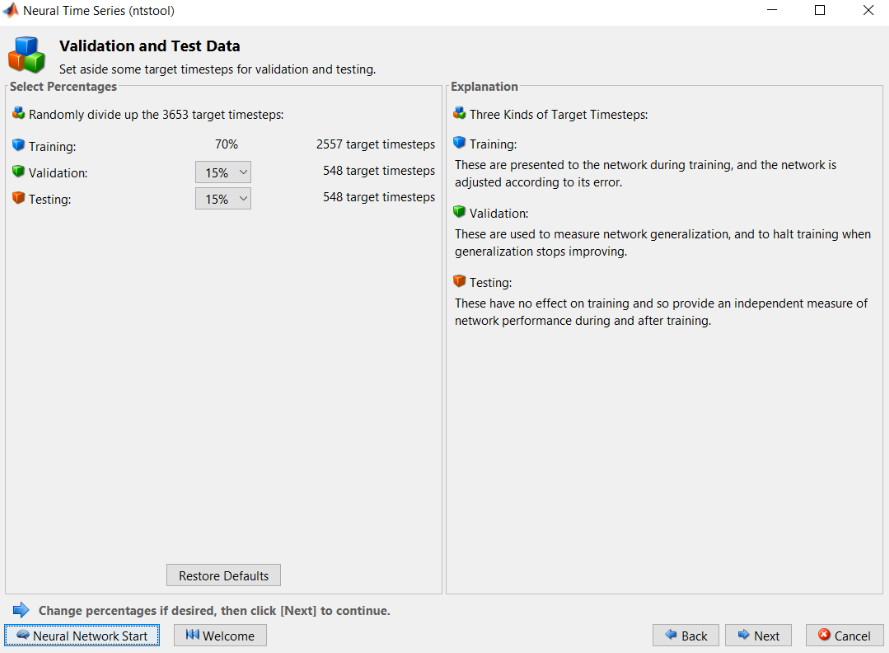


Fig 8.4: Determination of parameters for validation and testing.

After determination of parameters for validation and testing, the toolbox opens the ‘Network Architecture’ menu. Here the number of neurons and delays can be selected. In this case, it is set to 10 and 2 for neurons and delay respectively.

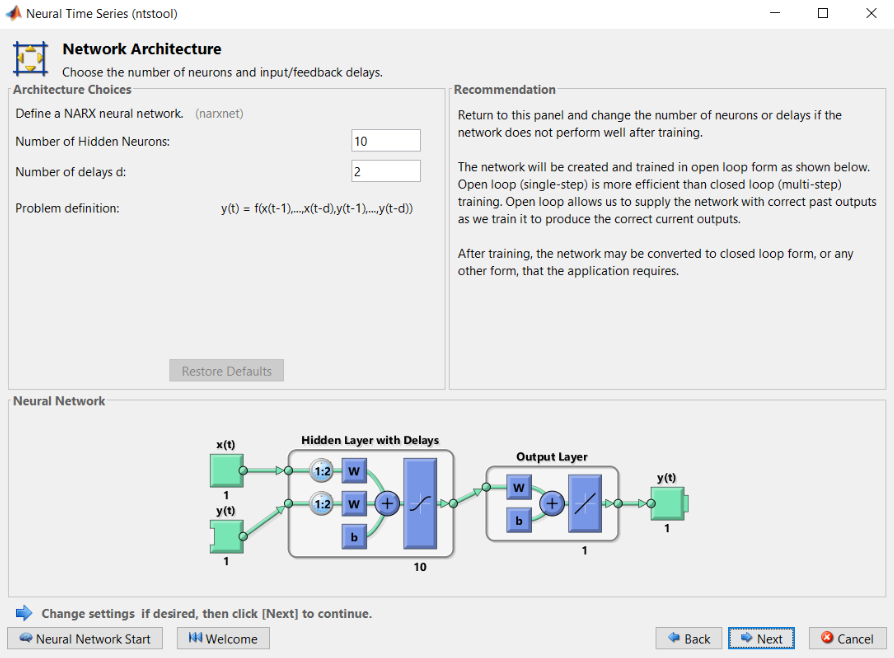


Fig 8.5: Determination of network architecture.

After determining all the parameters MATLAB can begin the training of the neural network as shown in the Fig 8.6.

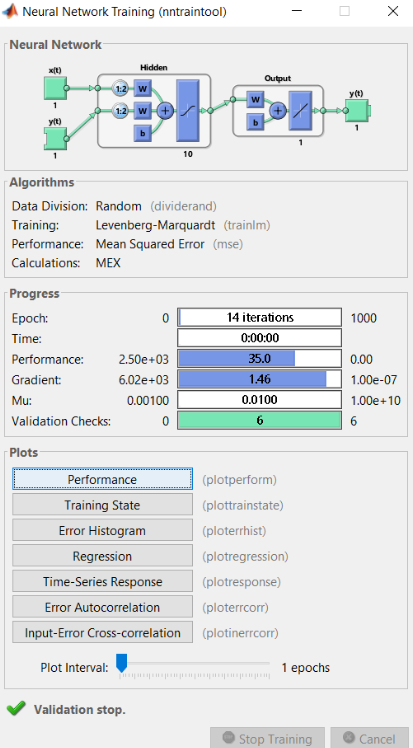


Fig 8.6: Training of the neural network.

**9. Results**

After completion of the training, the output of the neural network (average temperature) with respect to the input (date) is shown in Fig 9.1. The regression values as shown in Fig 9.2 for training, validation and test are 0.86, 0.82, 0.80 respectively. The neural network created is named ‘net\_temp’. This neural network can be used to predict future temperature values

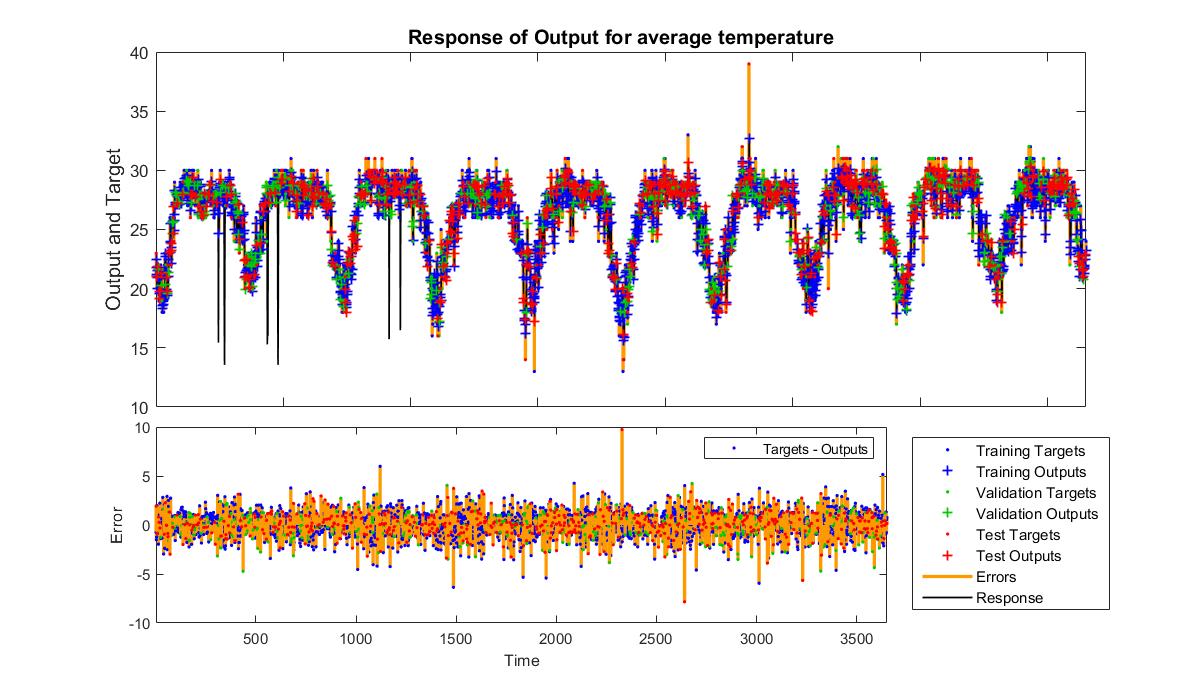
****

Fig 9.1: Time series data predicted by the neural network for average temperature.

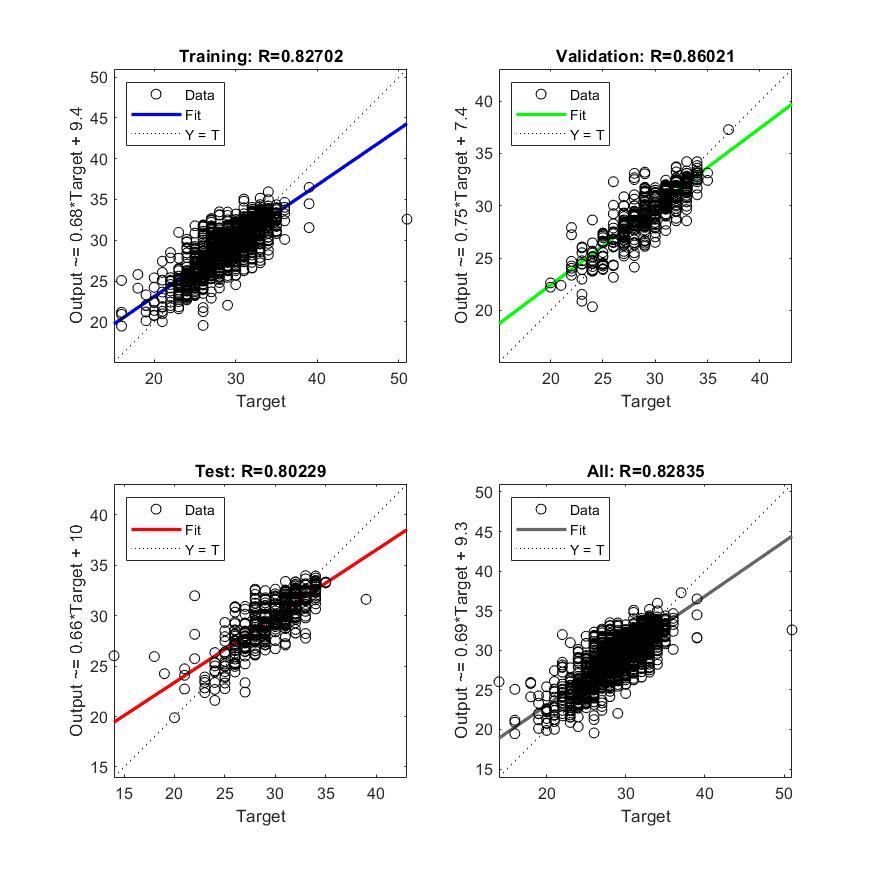


Fig 9.2: Regression diagrams of training validation and output datasets for average temperature.

Similarly, another neural network named ‘net\_hum’ is created to predict future temperature values. Fig 9.3 and 9.4 shows the time series data and the regression plots respectively.

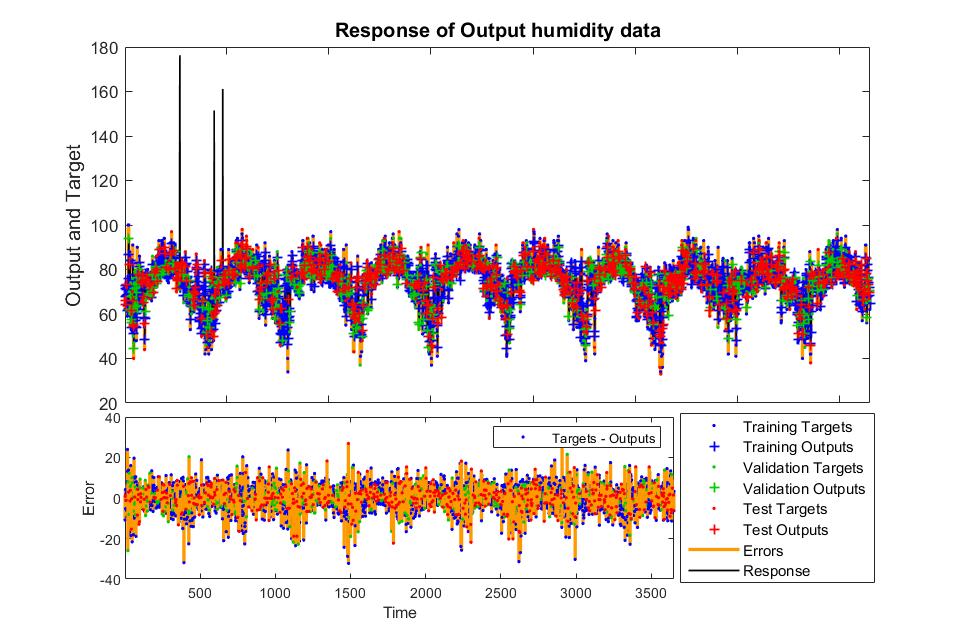


Fig 9.3: Time series data predicted by the neural network for average humidity.

MATLAB simulates the given input data and optimises the neural network to perform the task of prediction. The regression values for humidity predicting neural network are 0.82, 0.81 and 0.81 for training, validation and test datasets.

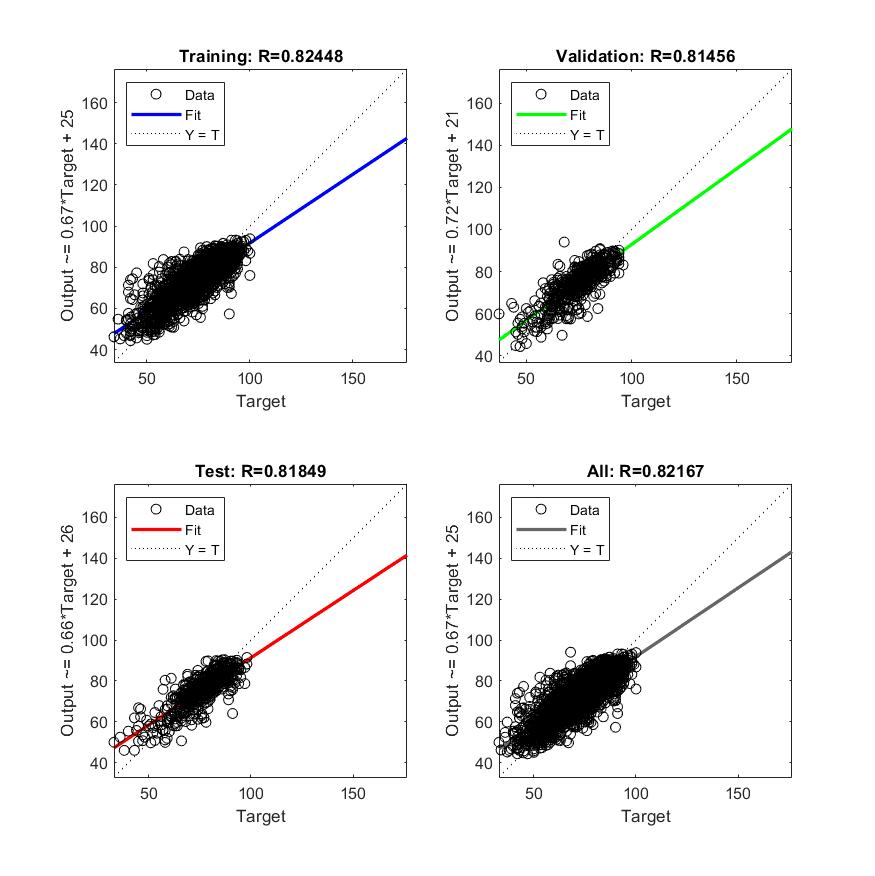


Fig 9.2: Regression diagrams of training validation and output datasets for average humidity.

**References**

[1] “Documentation.” [https://.readthedocs.io/en/release/search.html?q=esp8266](https://nodemcu.readthedocs.io/en/release/search.html?q=esp8266) (accessed Jun. 28, 2021).

[2] “ESP8266EX Datasheet,” 2015. [Online]. Available: http://bbs.espressif.com/

[3] T. Liu, “Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302) Capacitive-type humidity and temperature module/sensor.” [Online]. Available: http://www.Datasheet4U.com

[4] S. Systech, “SOLOMON SYSTECH SEMICONDUCTOR TECHNICAL DATA SSD1306 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller,” 2008. [Online]. Available: http://www.solomon-systech.com

[5] “Specification of product for Lithium-ion Rechargeable Cell Energy Business Division,” 2009.

[6] “NanJing Top Power ASIC Corp. TP4056 1A Standalone Linear Li-lon Battery Charger with Thermal Regulation in SOP-8 DESCRIPTION.”

[7] “Simple 6V to 12V DC Converter Circuit and Its Working.” https://www.efxkits.us/6v-to-12v-dc-converter-circuit-explanation/ (accessed Jun. 28, 2021).

[8] “Learn More - ThingSpeak IoT.” https://thingspeak.com/pages/learn\_more (accessed Jun. 28, 2021).