**1.INTRODUCTION**

Now as technology is increasing there are lots and lots of developments in the field of science and technology but even in these days as the population increases the resources become scarce as water is one of the most important resource .Even though earth constitutes 71% of water 97% of water is salty and only 0.03 is used for drinking and if population continues to increase at same rate then there will be a time where there might be no water available one way to counteract this is by optimal use of water and water conservation. In olden days ther use to be no concrete roads so the water from rainfall use to go deep into soil but as the roads are being constructed it is difficult to conserve rainfall water so in order to avoid this we came up with idea to predict based on humidity and temperature whether rainfall occurs or not when the rainfall occurs at areas where there are concrete roads we can use other techniques to conserve water so we can improve water quantity for future generations

**1.1Motivation**

Even though there is abundance of water on earth the climate is prone to change unlike in olden days when rainfall occurs seasonally due to pollution and other changes in the atmosphere Rainfall occurs even in summer winter and non-rainy seasons when rain occurs at that time. A software able to predict occurrence and conserve water during non-rainy season can help in increasing water resources by predicting weather and storing them using rainwater storing techniques

**1.2 Basic Definition**

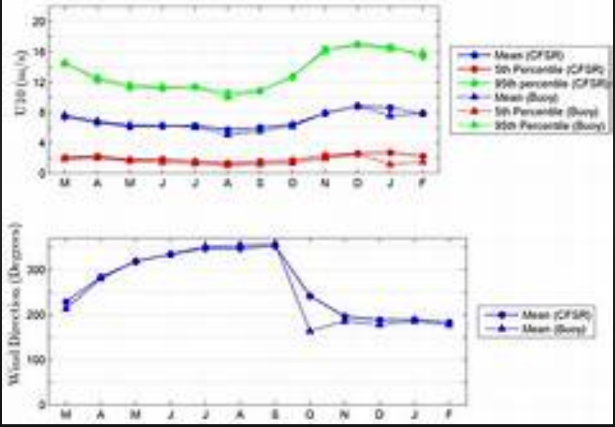
In the early 1950’s The prediction of future based on past data would have been considered impossible But Alan Turing proposed a theory saying that machines could think this was the beginning of artificial intelligence .In the early to late 1980’s when machine learning algorithms , increase in computation and data we are able to predict the future and as technology advanced we have microcontrollers and more efficient algorithms .This software uses to collect data from environment and analyse data and make predictions

**1.3 Problem Definition**

1. The normal data science websites like Kaggle will have clean data and data is past and we need to have data continuous and every interval so prediction will be correct
2. In this project there will be a IOT device connected with sensors in order to collect data and store in file using pyserial
3. The next process includes collect previous data and store in a file train.csv and collect data in test.csv
4. Now to write a python script to pre-process the data
5. Now selecting features as input and output (Target variables)
6. Then the next process is to configure the neural network which is used to train data and make predictions
7. Similarly we use another neural network to predict weather and configure train and test the neural network
8. We can check for accuracy and see the predictions if the accuracy is less then we can tune the hyper parameters and improve the accuracy
9. Then for incoming input we can predict the output this allows us to analyse if rainfall occurs or not then to take preventive measured

**1.4 Existing System**

In existing system of Rainfall prediction the data is collected and stored then the data is split as test or train based on attributes of data the analytics is done which include splitting of the same train and test data as weather is very versatile and change daily so there are no such software which collect data daily

****

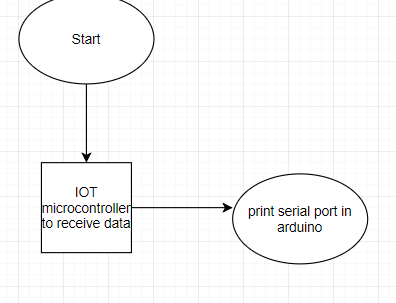
**Fig. 1.1:** Output from rainfall prediction from existing system.

**2.PROPOSED SYSTEM**

**2.1 Methodology**

The project titled RPID is a python application cum IOT application which basically a usage of IOT microcontroller with a Temperature sensor which collect temperature and humidity from a weather. In basic Rainfall prediction as there is no streaming data acquired, we use the same data for test and training and data at every point influences the weather and rainfall. So In this project it was decided to use the IOT to gather the data and use in synergy with a python script which uses deep learning models.

In this project the plan was to try storing the data accumulated from the IOT device and try storing in the file which can be accessed using python’s panda’s library then we can use the data frame and clean the data then try using the pre-processed data. Normally the pre-processed data gives more accurate results compared to any uncleaned data. The output of data when uncleaned uses to regression techniques to predict rainfall but on cleaning it either outputs 0 or 1 saying rainfall occurs or not.

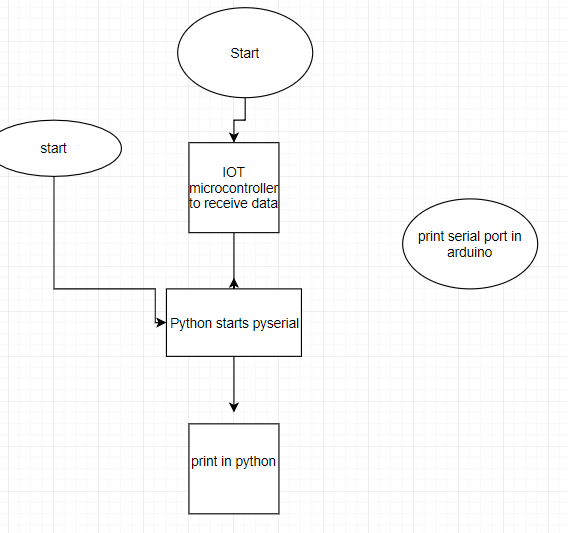


**Fig. 2.1.:** Flow chart design of small part of software

This project even uses weather prediction where in we use the previous data where the labels are given this type of techniques in deep learning/machine learning is called supervised learning where we show some labels and machine learns from it this allows machine to recognise which label is it i.e. what weather it is based on temperature or humidity given this might be useful in order to take some countermeasures whenever any disaster occurs due to heavy weather conditions such as cyclones and typhoon

**2.2 Architecture of The Proposed System**

The above is the architecture of the Parts of RPID which shows how the python application works. Here using Arduino code which is deployed into iot is used which in turn is connected to battery which powers the device this allows the microcontroller to send in data to a port which has been enabled onto the computer .This allows us to capture the data into our computer even though the data is arriving we must try logging the data else there is no use hence our initial approach was to use tera term software but since it doesn’t append data from previous log we rejected the idea and wanted to use pyserial.

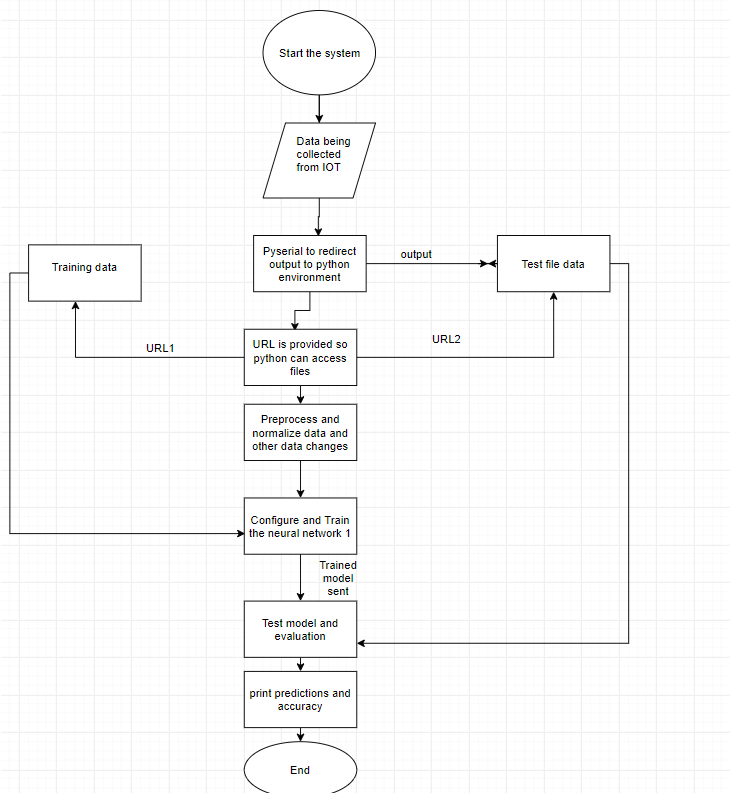


**Fig 2.2** Sending IOT data to python using Pyserial

In the above model we have seen only collecting data from Arduino but Arduino uses c++ language which is not suitable for data analytics case hence we use pyserial to serial the Arduino data into python script where we store the file the package for python to retrieve Arduino data is pyserial .The Pyserial package uses an Object instance to connect to the Arduino given the baud rate and port number then it is directed to python environment and stored in a file this is a csv file which stands for comma value separator a file used for python data analytics

2.2.**1 The flow of the RPID**

The project requires the proper connection of IOT Microcontroller with the sensor and a power source so that the power is supplied so data is captured which then uses python to store the test data into csv file the train data can be taken from the internet and stored in train file then we can give URL of the file and they can be accessed then by cleaning the data and applying deep learning model we train the data then we test it against it against an incoming data to predict the occurrence of rainfall and predict weather conditions. This project mainly uses anaconda an opensource tools for data analytics and estimate the future weather conditions.

 **Fig. 2.3:**  Complete flow chart of proposed system

data from pyserial there are some garbage bytes occupying the file so we have to do this task manually

**3.REQUIREMENTS**

**3.1 Software Requirements**

**3.1.1Python:**

**Python** is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Applications of python:

1. Web and Network Programming
2. 3D Game programming
3. Data Science
4. Cryptography

**3.1.2Anaconda:**

Anaconda is a free and open-source[[5]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-5) distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system *conda*. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

**Anaconda distribution** comes with more than 1,400 packages as well as the [Conda](https://en.wikipedia.org/wiki/Conda_(package_manager)) package and virtual environment manager, called AnacondaNavigator [[8]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-8), so it eliminates the need to learn to install each library independently.

The open source packages can be individually installed from the Anaconda repository [[9]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-AnacondaRepo-9) with the **conda install** command or using the **pip install** command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in most cases they can work together.

Custom packages can be made using the **conda build** command, and can be shared with others by uploading them to Anaconda Cloud,[[10]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-AnacondaCloud-10) [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda [[11]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-11).

### **3.1.2.1Anaconda Navigator:**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator [[12]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-12):

1. JupyterLab
2. Jupyter Notebook
3. QtConsole
4. Spyder
5. Glue viz
6. Orange
7. RStudio
8. Visual Studio Code

**3.1.3 C++:**

C++ is a general-purpose programming language that was developed by Bjarne Stroustrup as an extension of the C language, or "C with Classes". It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++compilers, including the Free Software Foundation, Microsoft, Intel, and IBM, so it is available on many platforms.

C++ was designed with a bias toward system programming and embedded, resource-

constrained software and large systems, with performance, efficiency and flexibility of use as

its design highlights. C++ has also been found useful in many other contexts, with key

strengths being software infrastructure and resource-constrained applications including desktop applications, servers and performance-critical applications

**3.1.4 Arduino IDE:**

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages *Processing* and *Wiring*. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring

**3.2. HARDWARE REQUIREMENTS:**

**3.2.1 Arduino (Octabrix):**

**Arduino** is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control both physically and digitally. Its products are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits. The Arduino board which is used is called and Octabrix it is a device which was built by a start up called ib-hubs located in India.

**3.2.2 Jumper wires:**

A **jump wire** (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

**3.2.2.1 Types:**

There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

* Solid tips – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.
* Crocodile clips – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.
* Banana connectors – are commonly used on test equipment for DC and low-frequency AC signals.
* Registered jack (RJnn) – are commonly used in telephone (RJ11) and computer networking (RJ45).
* RCA connectors – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.
* RF connectors – are used to carry radio frequency signals between circuits, test equipment, and antennas.

**3.2.3Breadboard:**

A breadboard is a construction base for prototyping of electronics. Originally it was literally a bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

**3.2.4Resistors:**

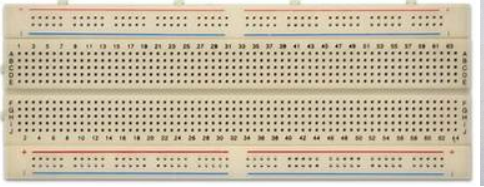
A **resistor** is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many watts of electrical power as heat, may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity. Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in electronic equipment. Practical resistors as discrete components can be composed of various compounds and forms. Resistors are also implemented within integrated circuits. The electrical function of a resistor is specified by its resistance: common commercial resistors are manufactured over a range of more than nine orders of magnitude. The nominal value of the resistance falls within the manufacturing tolerance, indicated on the component.



**Fig. 3.1:** Hardware Arduino



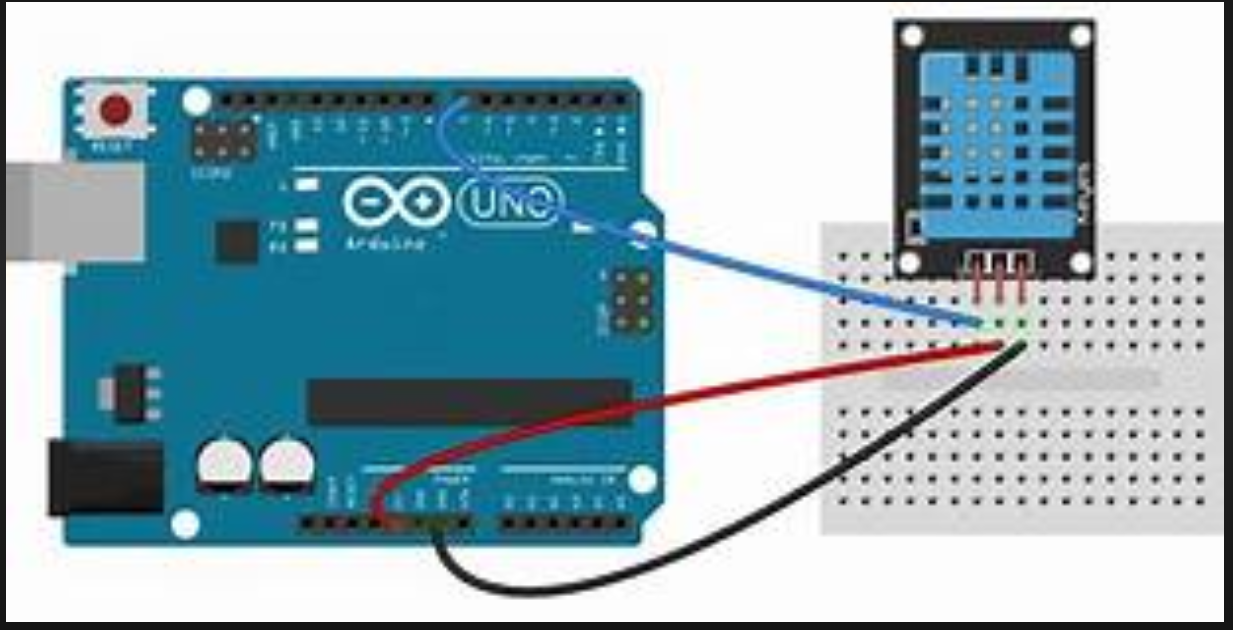
**Fig 3.2** **Resistors**



**Fig. 3.3:** Bread Board

****

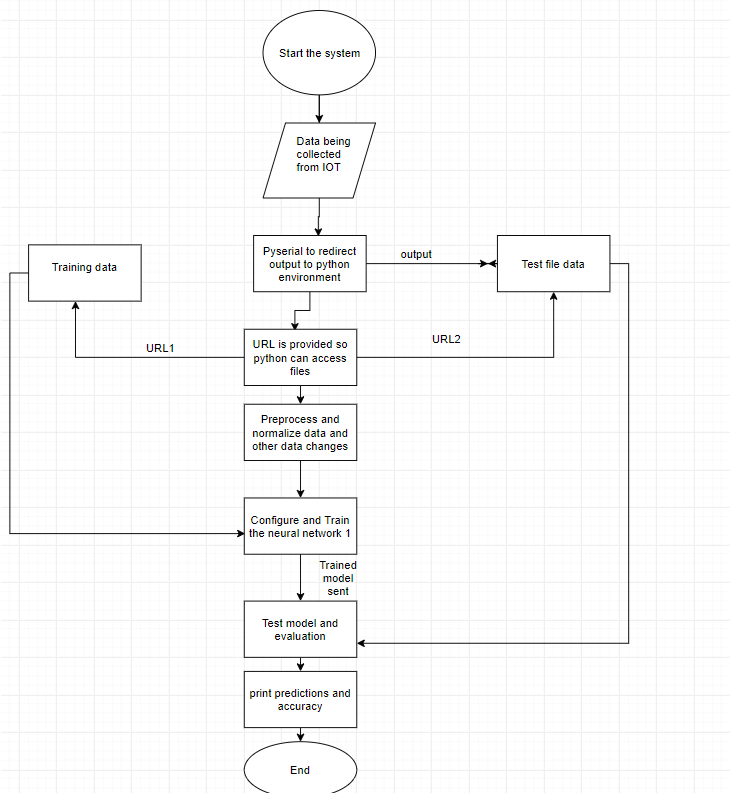
**Fig 3.4** Jumper wires

****

**Fig 3.5** DHT 11 Sensor to Arduino

**4.SYSTEM DESIGN**

**4.1 Flow chart**

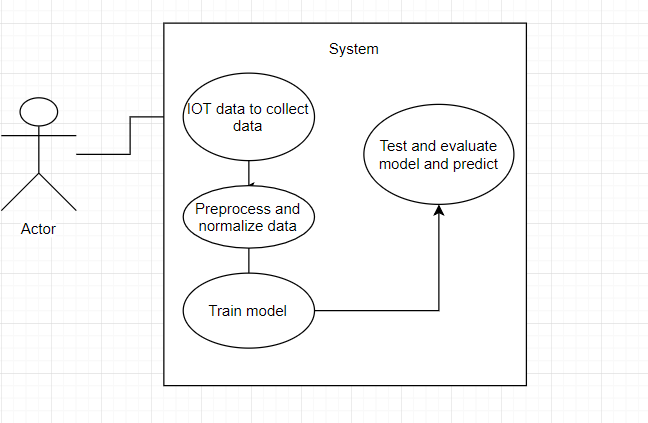
****

**Fig. 4.1:** Flow chart of the RPID

The above flow chart represents the complete flow of creation of the Mini project RPID. First we connect the iot microcontroller to the DHT11 sensor. Then we use the python package pyserial to redirect the data going into Arduino then data is stored in a file later the file is accessed and manipulated then a neural network is constructed and configured then it is trained, tested and evaluated then predictions are done on weather and rainfall.

**4.2 Use case diagram**

The below is the use case diagram of the RPID project. This represents how the application actually works by showing the tasks done by the software. First the user runs the IOT device and collect the data which he stores in a file manually (This is due to some bug in project which will be fixed later on). Then user gives URL of files and runs the software This automatically preprocess the dataset and select features (If dataset has different columns they are required to change the attributes in d[[]] command) Then training data is selected and a model is trained then the same model is used to predict on given data.Similar process occurs for weather prediction also. Then the accuracy of model is evaluated.

****

**Fig. 4.2:** Use case diagram of RPID application

**5.IMPLEMENTATION OF PROJECT**

In order to run this application, we need to get some tools and packages that are needed and also we need to perform some steps to execute this application which are explained below.

**5.1Required parameters:**

5.1.1 Arduino IDE:

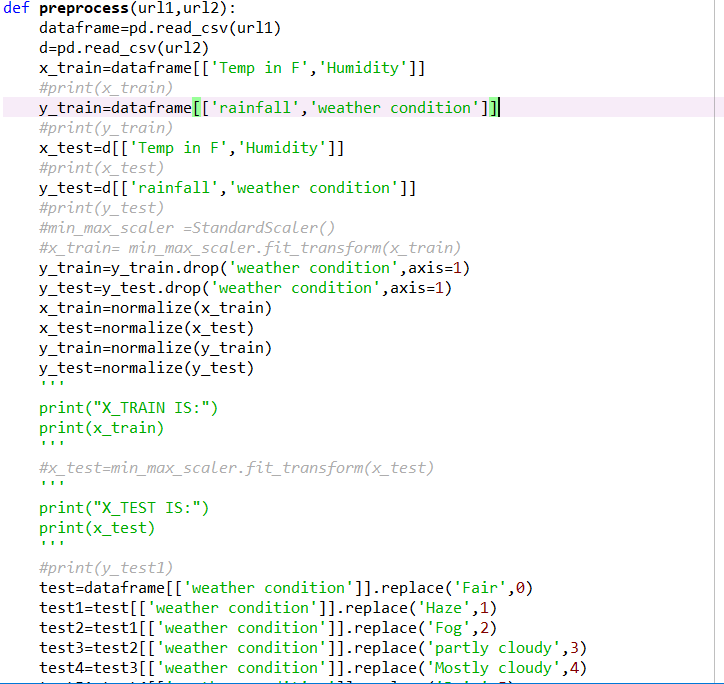
We need to install this from the chrome browser and the libraries of Arduino the main one being DHT11 sensor package to run the code in IOT Microcontroller The system must be configured beforehand.

5.1.2 Keras & Anaconda:

Anaconda as said before is a tool which supports data science in R and Python Keras is Python library which uses TensorFlow as Backend and runs all deep learning Algorithms.

5.1.3 Pyserial:

It is a python library package which is used to redirect data from serial port in Arduino into python this allows us to store data in a file. First we need to Connect the IOT Microcontroller using jumper wires and resistors to the power source the jumper wires are mostly male-female type then we arrange the setup as show in fig() Then a code is written in c++ in Arduino IDE which is then deployed into micro controller as show in fig() Then pyserial dependencies are installed into system before that we install anaconda prompt and navigator Here we use a python IDE called Spyder here we write a python script which redirects the code from Arduino to python which is stored in a file then we write another python script to extract data from file to data frame and pre-process and split into parts .



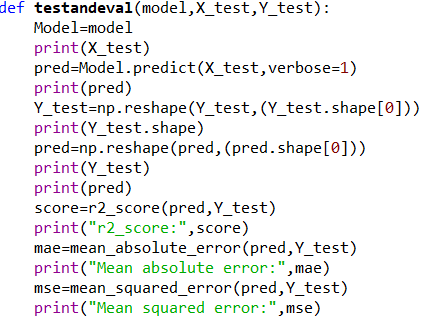
**Fig. 5.1:** pre-processing code

Then we need to write the code to configure a neural network using Keras and configure the model then we train the model on training data which returns a trained model then we write a function to send trained model test data and target parameters to predict the values and give us the output as mentioned in Fig. 5.2.



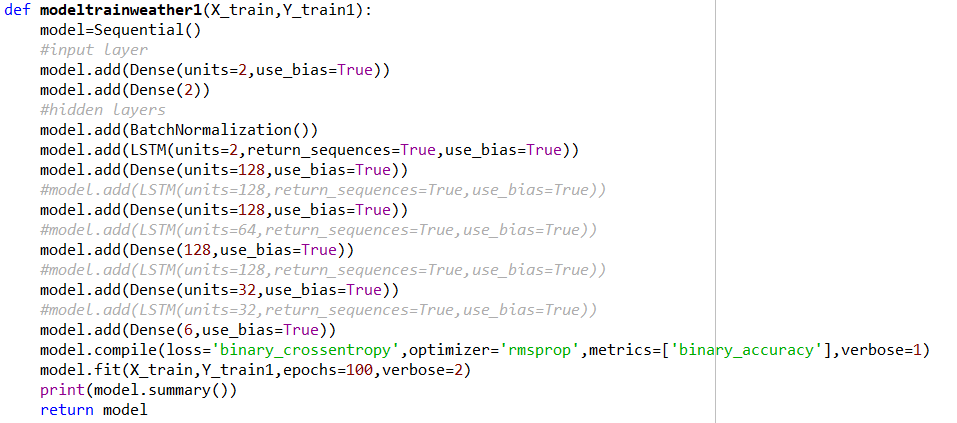
**Fig. 5.2**: Neural network training for rainfall prediction

Then we write another method which takes neural network but for weather prediction and then select a neural network algorithm or build layers using Keras. layers package. This again result in same sequence such as training and testing data but if data has not been pre-processed then It generally gives a low accuracy hence it is necessary to pre-process data before analysing the data and implement our required functions in it as shown in Fig. 5.3.

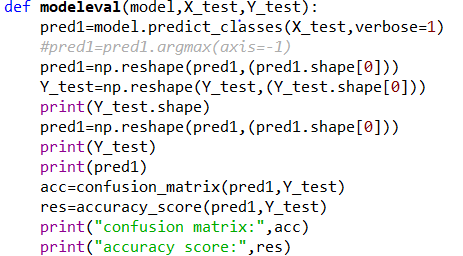


**Fig. 5.3:** Test and evaluation of rainfall predictor NN

.



**Fig. 5.4:** Neural network training for Weather prediction

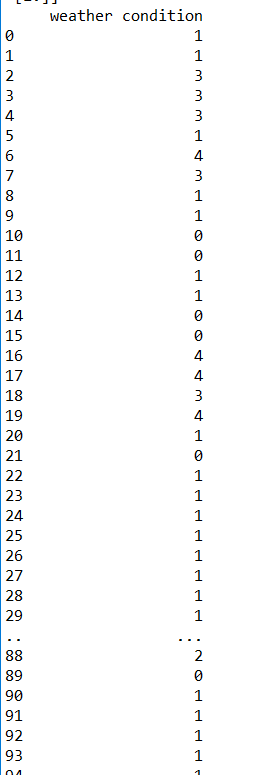


**Fig 5.5** Test and evaluation of weather prediction NN

**6.RESULTS**

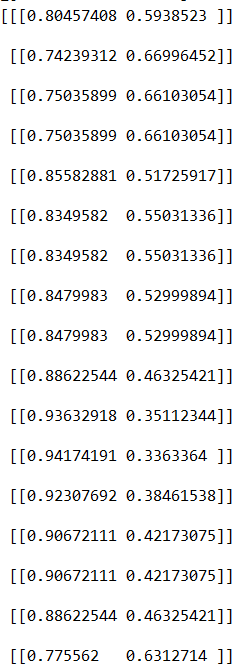
When we start the web application the first page you will see is a data being redirected form the IOT microcontroller to python script but due to some bug we can’t directly redirect hence manual techniques are used after storing give path of train and test files.After this we take the data into the python environment and use pandas library to pre-process data and normalise the data so it gives better accuracy

**Fig 6.1(SCREEN 1):** pre-processed data

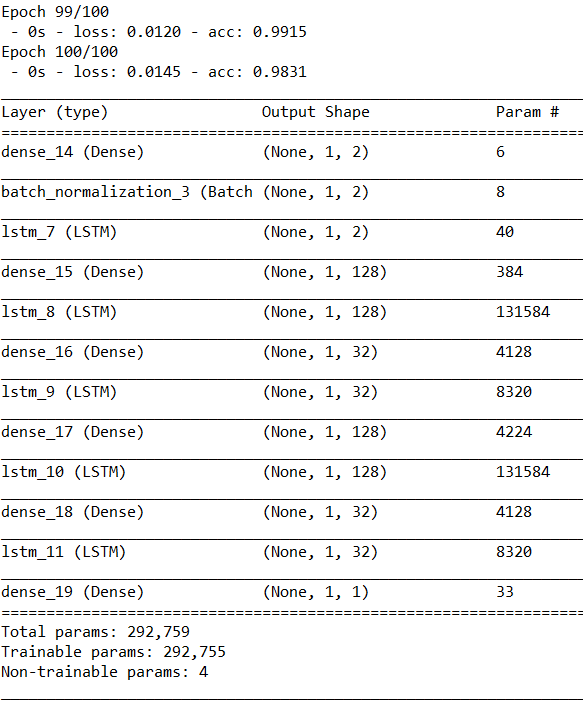


When Normalizing it results in and Since in Keras we are using a LSTM model which takes input in 3-d we use NumPy to rearrange and feed to train the network.

. **Fig 6.2** (**SCREEN 2):** Normalized data

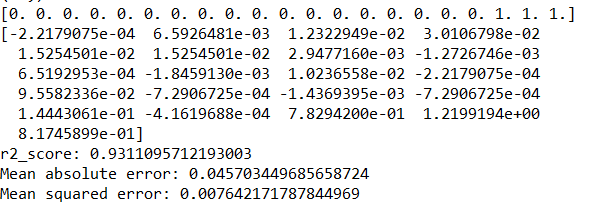


**Fig 6.3 (SCREEN 3):** Training and summary



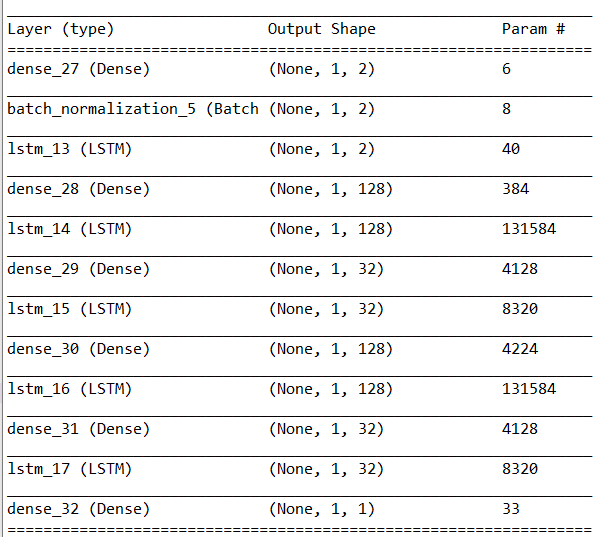
In the following method we use neural network in order to train on our training data this neural network is a hybrid model as it uses different layers and predict outputs for us.

**Fig 6.4 (SCREEN 4):** Testing and accuracy

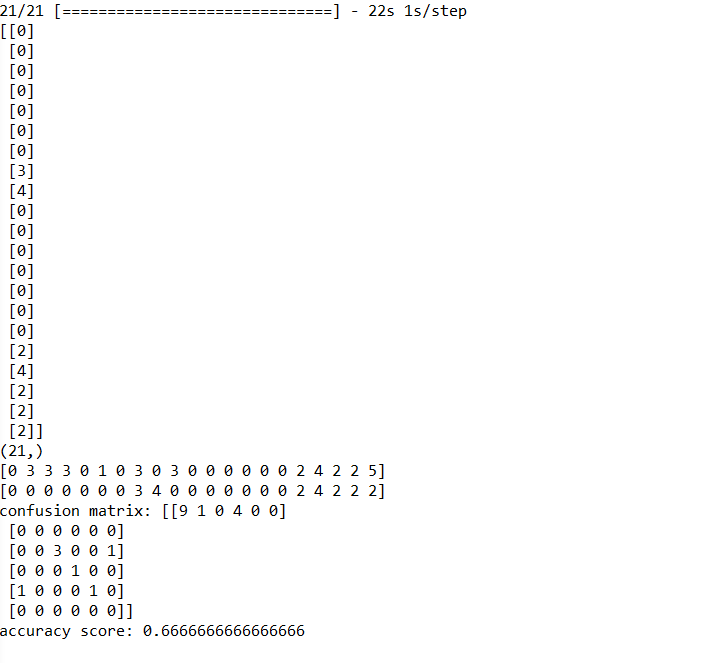


Here we use testing methods to predict output of our neural network to predict rainfall In the fig the r2 score is how accurate the model is if r2 is near to 1 then its good else its bad .

**Fig 6.5 (SCREEN 5):** Same process but training a neural network for weather prediction



**Fig 6.6 (SCREEN 6):** Output and accuracy



**7.CONCLUSION & FUTURE SCOPE**

**7.1CONCLUSION:**

The RPID Application has been computed successfully and was also tested but contains some bugs which will be resolved in future dates which were found by taking “test cases”.The software is developed using Arduino IDE, Anaconda ,Keras , TensorFlow in windows 10 environment.

The goals achieved by the software are:

1. Taking streaming data
2. Pre-process the data
3. Normalize the data
4. Train neural networks
5. Test and evaluate
6. Predict

**7.2FUTURE SCOPE:**

In future, The project will be improved so that it has its own battery source and transmit data wirelessly the accuracy of model can be further implemented and like to increase the count of microcontroller so we can predict the large climate changes like an ice age so we can take preventive measures and add user friendly interface.

**REFERENCES**

**[1]** <https://www.arduino.cc/>

**[2]**<https://www.anaconda.com/distribution/>

**[3]** <https://keras.io/>

**[4]**<https://machinelearningmastery.com/>

**[5]** <https://stackoverflow.com/>

**[6]**<https://www.instructables.com/>

**[7]** [https://www.wunderground.com](https://www.wunderground.com/)*/*