Introduction to IoT and Its Industrial Applications (CS698T)
Indian Institute of Technology Kanpur
Smart Irrigation System using IOT, Assignment 2

Member Names: Himanshu, Manish
Member Emails: himanshul21@iitk.ac.in, manishv21@iitk.ac.in
Member Roll Numbers: 21111403, 21111036
Date: November 3, 2021

Abstract

Day by Day, India's Population is increasing at a very high rate, therefore we need to strengthen our Agriculture to avoid food scarcity. In today's world, due to many environmental reasons, the farmers are dealing with shortage of rain and inadequacy of water. Hence, we require a modern solution for this problem, so as we can strengthen our agriculture sector.

1 INTRODUCTION

Agriculture undoubtedly, is the largest livelihood in India. India's major source of income is from agriculture sector. Around 70% of people in India depends on agriculture. The problem is that most of them uses traditional methods which are operated manually. The basic problem with traditional methods is that two scare and valuable resources that is water and energy are not utilised properly.

So, Our Problem Statement which is "Aravind is an ambitious farmer who wants to improve the yields from his farmland. While looking for ways to improve the yields, he came across tech buzzwords like the Internet of Things and Machine Learning. He wants to try them out to build an intelligent irrigation system for his field.

The irrigation system he wants to build contains servo motors that control the water supply to the farm, based on the present humidity, temperature levels in the farm. Using sensor data, Aravind wants to build a machine learning model that predicts how much water (in percentage) should be supplied to his farm. Aravind also does not want to water the plants during nighttime."

So, The irrigation system that we want to build should contain 4 sensors in such a way that each sensor can sense humidity and temperature level. These values are then provided to our ML model on the IOT board to predict the water flow in percentage. After this, the water flow percentage will be provided as input to the servo motor which will then map these values with the servo motor rotation(0 degree to 180 degree).

The proposed system makes use of microcontroller Arduino Mega and IOT which enables Aravind to monitor moisture and temperature levels.
One another thing is that Aravind also doesn't want to water the plants during the nighttime. So, We'll be using LDR sensor to detect day/night.

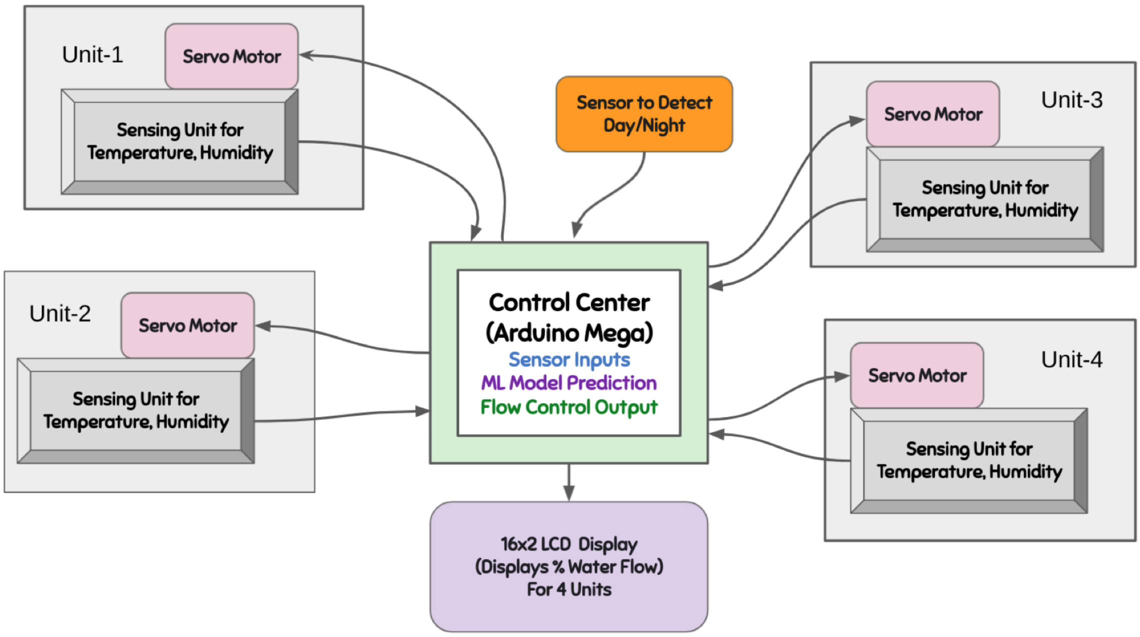

Figure 1: Block diagram of proposed System

2 SOFTWARE & HARDWARE REQUIREMENTS

2.1 Arduino Mega

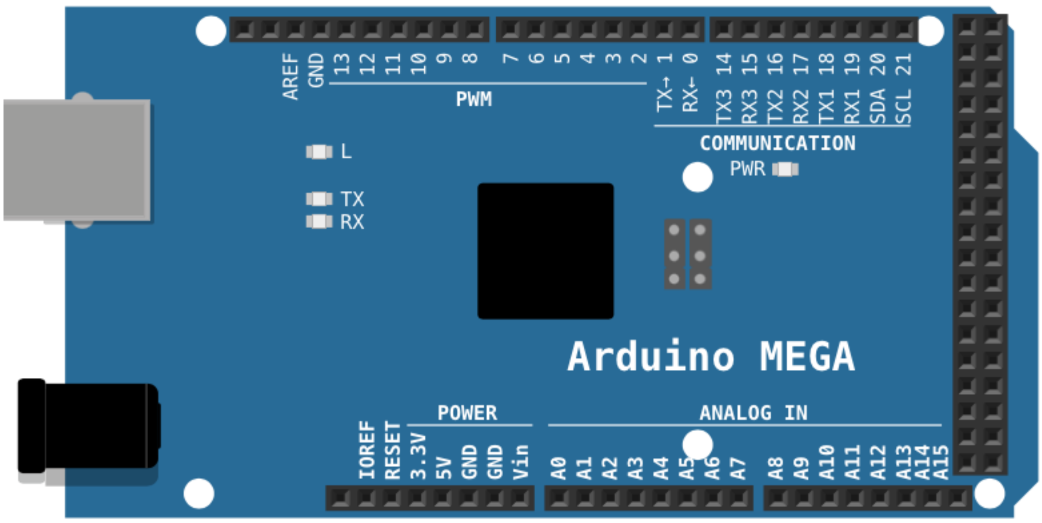

Figure 2: Arduino Mega

Powered by the ATmega2560 chip, Arduino Mega 2560 comes with a 256 KB of Flash program memory, 4KB of EEPROM and 8KB of SRAM. This board has 54 digital pins, 16 analog input pins, & 4 serial ports. It runs at a frequency of 16Mega HZ.

2.2 DHT22 Sensor

This Sensor is used to get the soil moisture levels and temperature. Soil moisture sensors are used for measuring the water content of the soil. They basically use capacitance to measure dielectric permittivity of the surrounding medium in soil, dielectric permittivity is a function of the water content.

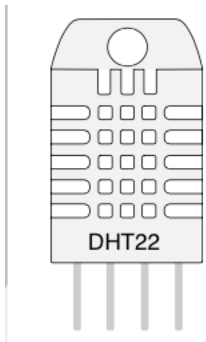

Figure 3: DHT22 Sensor

2.3 LDR sensor

The LDR (light-dependant resistor) sensor, also known as photo-resistor, is basically a photocell that works on the principle of photoconductivity. The passive component of this sensor is actually a resistor whose resistance value decreases when the intensity of light decreases.

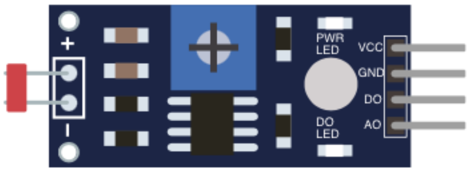

Figure 4: LDR sensor

2.4 Servo Motor

Servo motors are electronic devices and rotary or linear actuators that rotate and push parts of a machine with precision.

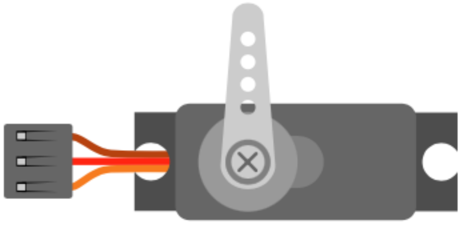

Figure 5: Servo Motor

2.5 LCD display

The LCD which we are using is LCD1602 which uses the Hitachi HD44780 LCD Controller chip. The chip comes with a built-in font, as well as, it has the ability to define up to 8 custom characters. It will display the % water flow for the the 4 units.

Figure 6: 16x2 LCD

3 ML Model

For the linear prediction problem, many researchers have previously suggested various machine learning and deep learning models. However, deep learning provides an end-to-end approach that is not commonly seen in the machine learning model. In this assignment, we use a neural network model to predict the percentage of the Workflow required to irrigate the agriculture field based on current atmospheric temperature (in degree C) and soil humidity (in percentage).

Why Neural Network

1. The biggest benefit of deep learning is that it is able to execute featuring engineering on its own. In a deep learning approach, the data is scanned by an algorithm in order to identify features that correlate and later combine them in order to promote fast learning.

2. The ability to work with insufficient knowledge:

3.1 Data Preprocessing

Neural networks require significantly less hand engineering, and the network itself does all the feature extraction part. Only Water Flow(%) column is changed during the preprocessing and normalised so that its value comes in 0 to 1. Normalisation is essential because the network's output will predict the probability of Water Flow, ranging from 0 to 1.

The dataset contains 200 samples that split into training and test in the ratio of 80:20 percent. That means the training set contains 160 samples, and the test set contains 40 samples. For splitting, the random state is fixed to 42.

3.2 Network Architecture

We are using three layers neural network with two hidden layers and one output layer. Each hidden layer has ten nodes, and the Relu activation function brings nonlinearity to the network, and the output layer uses the sigmoid activation function. The input layer has two nodes for temperature and humidity. The total number of the trainable parameters in the network is 151 and 0 non-trainable parameters.

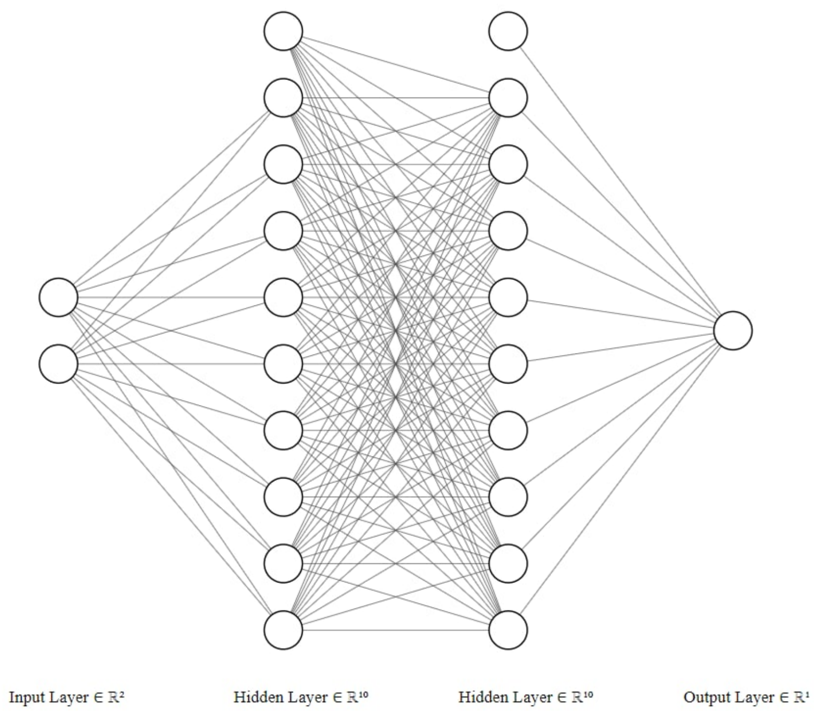

Figure 7: Network Architecture

3.3 Model Training

Model training is the phase in the machine learning development lifecycle where we try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

We trained our neural network for 100000 epoch with early stopping criteria, monitoring the mean square error for 20000 epochs. If the model does not successfully find any improvement over MSE for 20000 epochs, it will be declared converged and the training process will halt. Also, early stopping will be storing the best weights for which it found the least mean squared error.

For the optimization algorithm, we are using "Adam" optimizer. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. The main advantage of Adam over other optimizers is its fast convergence.

The random seed for TensorFlow is set to 42 so that we can produce the same result again.

3.4 Model Accuracy

Calculating the accuracy of a regression problem is not very straightforward. In this assignment, we are using two metrics to calculate the model's accuracy, i.e., root mean squared error and r2 score. Accuracies are calculated against the training set, test set, and overall dataset.

RMSE - Root Mean Square Error

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \tag{1}$$

R2 Score - Coefficient of Determination

The coefficient of determination is a measurement used to explain how much variability of one factor can be caused by its relationship to another related factor. This correlation, known as the "goodness of fit," is represented as a value between 0.0 and 1.0.

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{2}$$

	Train Set	Test Set	Overall
RMSE	7.57	13.90	9.19
R$\hat{2}$ Score	0.95	0.87	0.93

Table 1: Accuracy of Model on Different Set

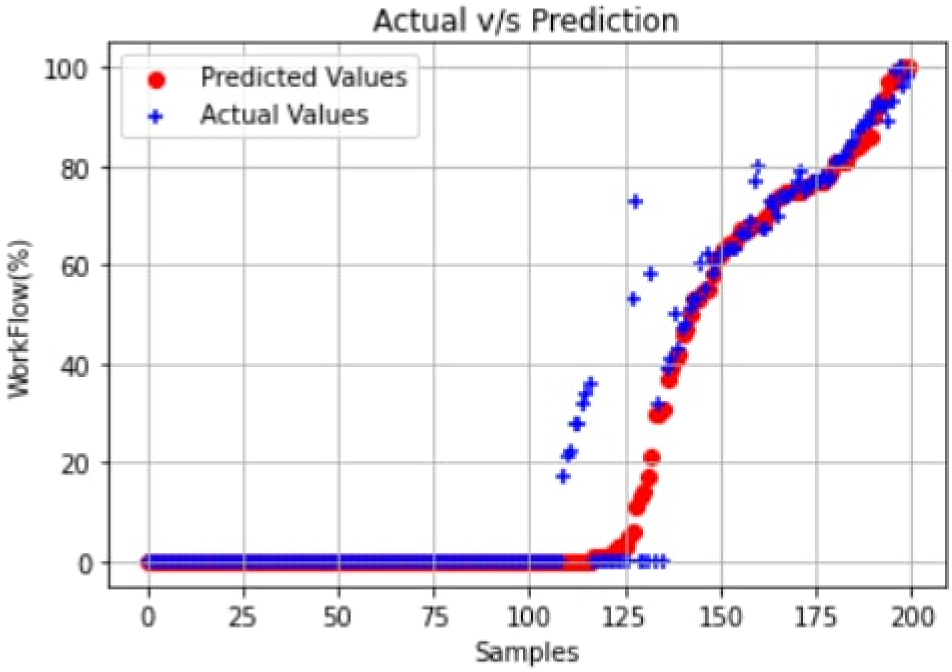

Figure 8: Actual vs predicted values

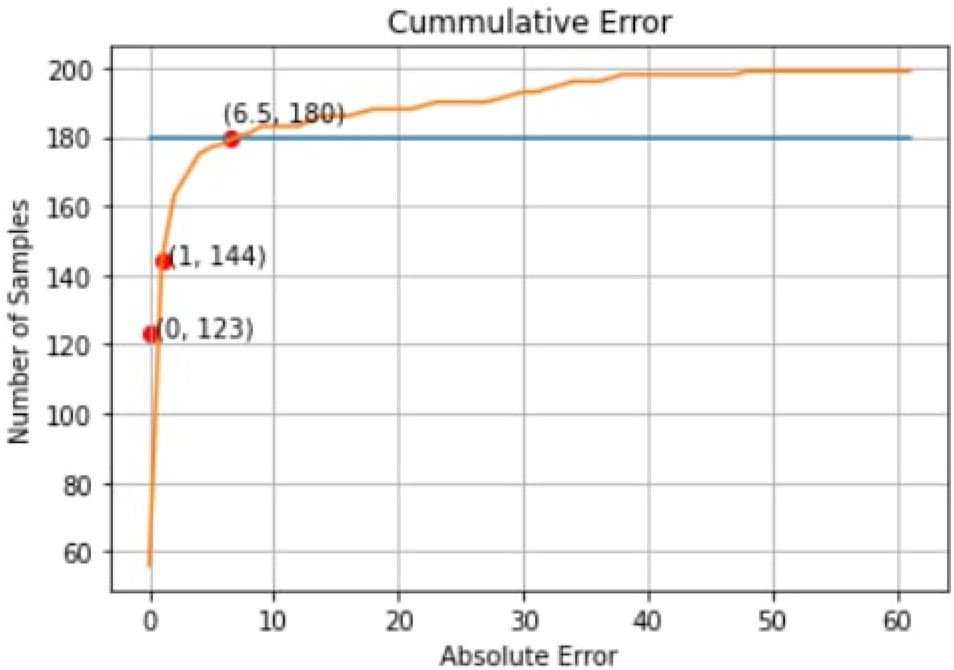

Figure 9: Cummulative Error

From the above following conclusion can be drawn.

1. For more than 60% times, our model predicts the correct percentage of water flow with the precision of zero.
2. For 72% of the time, the deviation of predicted values from their actual value is less than

1.

3. For 90% of the time, the deviation of predicted values from their actual value is less than 6.5.

3.5 Instruction to run ML Model

3.5.1 Dependencies

1. Python - 3.8.8
2. Tensorflow - 2.3.0
3. Keras - 2.4.0
4. Numpy - 1.20.1
5. Pandas - 1.2.4
6. Matplotlib - 3.3.4
7. Sklearn - 0.24.1

3.5.2 Execution

1. All the code for ML model is in Deep_learning_model.ipynb.
2. To produce the same result use seed value = 42 in train_test_split and tensorflow.
3. To run ML Model you required all the above dependecies and jupyter notebook.

4 ARCHITECTURE AND WORKING OF THE MODEL

Our proposed system consists of Arduino Mega IOT board, 4 DHT22 sensor, LDR sensor, 4 Servo Motors and a LCD display for displaying % water flow. The Architecture of our proposed model is shown below.

The DHT22 sensors are connected to the IOT board through input pins 5,4,3,2 respectively. Similarly, Servo motors are connected to the input pins 13,12,11,10 respectively. We have LDR sensor also which would basically return analog signals which we then use to calculate Illumination value.

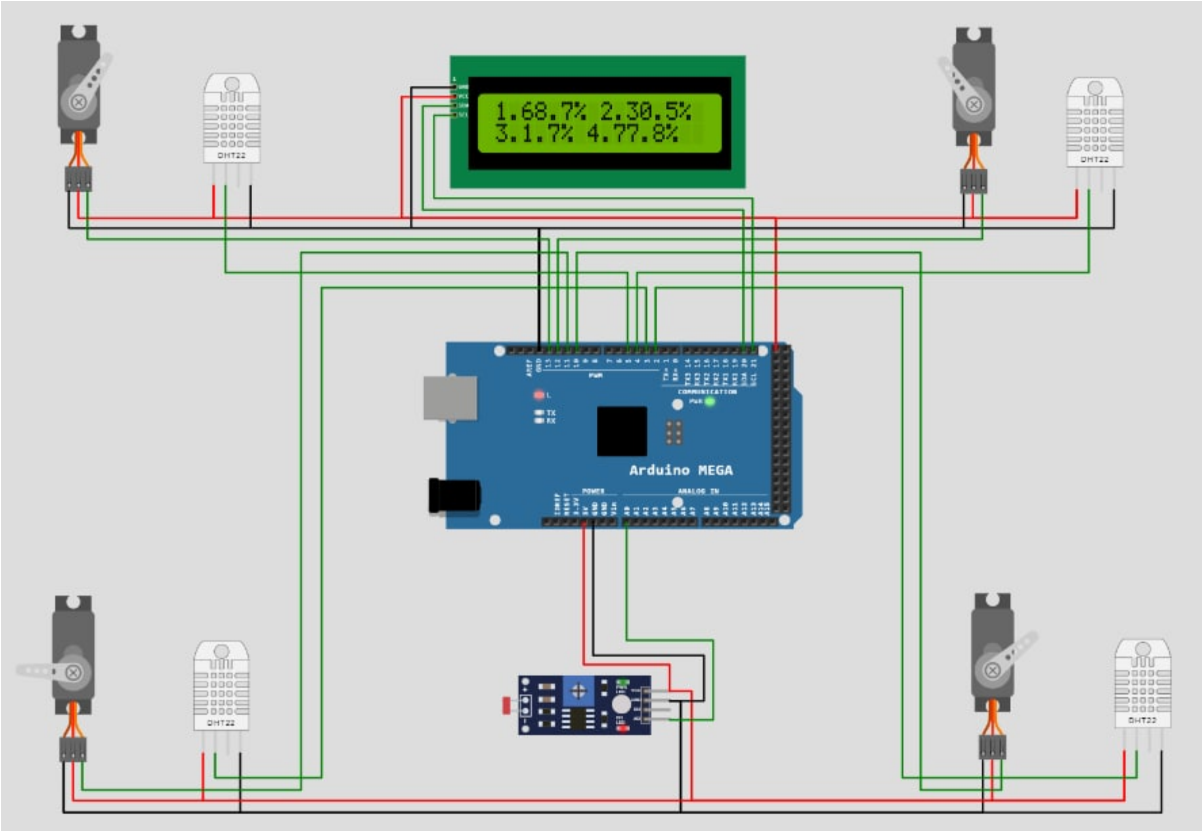

Figure 10: Architecture of Proposed System

4.1 Working

Now, Let us understand the working of our IOT system. There are 4 DHT22 sensors which will give us the temperature and soil moisture values. Each sensor unit has its own Servo motor for water flow. Then with the help of our deep learning algorithm which uses temperature and humidity values, we will set the values for %water flow. Then according to these values, the servo valves will open ranging from 0 degrees to 180 degrees depending on % water flow. And this % water flow values for all the four units will be displayed on the LCD screen.

With changing temperature and humidity levels, our % water flow changes which basically means that our servo motor handle angle will be changing depending on the humidity and temperature.

We are providing a link for our IOT model : Click here for the simulation

5 SUMMARY

We, first got the humidity and temperature values from the DHT22 sensor. Then we use Deep learning model to get % water flow. In our deep learning model, we used Relu's Activation function. We trained our model with 10000 iterations which gave us very good MSE. So, this low MSE helped us to predict very precise % water flow. At night time, we kept % water flow 0 as Aravind doesn't want to water his field at nights.

This proposed system can control flow the water pump depending on the soil moisture and temperature, thereby automating the whole process of irrigation. This IOT system uses information collected from soil moisture sensor and humidity sensor to irrigate soil which prevents clogging of soil, which in result prevents crop damage. From this IOT system, we can conclude that it is a considerable development in farming which is great for building a smart nation.

References

1. Neural Networks from Scratch in Python : Adarsh Menon
2. Getting Started with Arduino MEGA2560
3. Keras API reference
4. Welcome to Wokwi!
5. sklearn API Reference
6. Advantages of Neural Networks – Benefits of AI and Deep Learning