INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

Introduction to IoT and its industrial applications (CS698T)

A smart irrigation system using IoT

Members:
Manjyot Singh Nanra,
Roll-20111038,
manjyots21@iitk.ac.in
Sharanya Saha,
Roll-20111056,
sharanya21@iitk.ac.in

Professor:
Dr. Priyanka Bagade
Department of Computer Science

1 Introduction

Aravind, an ambitious farmer, wanted 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 wanted to build an intelligent irrigation system for his field. He is good at farming but didn't have much experience building Machine learning models and IoT systems.

An IoT-based smart irrigation system is implemented for Aravind's farm, following his requirements. Sensors are deployed to determine whether it is daytime or nighttime. During the day, the temperature and humidity values are sensed with the help of sensors. The information from the sensors is further used to predict the water flow percentage for irrigation. Here is the link to the project simulation: `https://wokwi.com/arduino/projects/313964051374801472`

2 Irrigation System

This section explains the design of the irrigation system, components and the sensors used, and their roles.

2.1 Components Used:

- Arduino Mega 2560: Core Micro Controller. It fetches the value of temperature and humidity during the daytime from the sensors, predicts the water flow required and adjusts the water flow of servo motor using the weights from the trained ML model [4].
- LCD 1602: LCD output to show percentage of water flow for each unit. It is used in l2c configuration mode [2].

2.2 List of Sensors/Actuators Used:

- DHT22: It will sense the temperature and humidity and send the data to Arduino Mega[1].
- Servo Motor: A standard Micro Servo Motor to control water flow. Depending upon the water flow required, servo motors angle will be adjusted to attain the desired water output[5].
- Photoresistor-Sensor: Photoresistor (LDR) sensor module to sense whether its day or night [3].

2.3 Design:

The irrigation system has 4 units. Each unit have a DHT22 sensor and a Servo Motor. The units are independent from each other i.e the water level for Servo Motor in Unit-1 is in accordance to the temperature and humidity values sensed by the DHT22 of Unit-1. Each unit is connected to the Arduino Mega, the core micro-controller. For connections, both DHT22 and Servo Motor uses 3 pins. A LDR sensor is deployed to detect the level of light and is connected to the core controller using 3 pins(VCC, GND and AO).Additionally, a LCD display is also connected to the core micro-controller using 4 pins(GND, VCC, SDA and SCL) to display the water flow level in percentage for each unit. Note that Unit 1 and unit 2 are at top half of design, and unit 3 and unit 4 are at bottom part of design

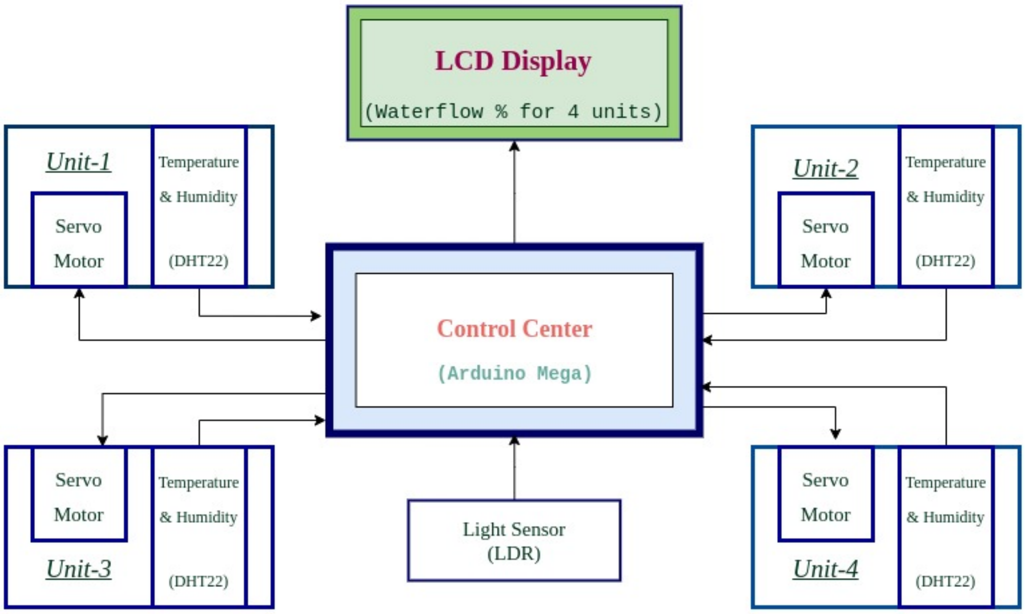


Figure 1: Block diagram of the final design

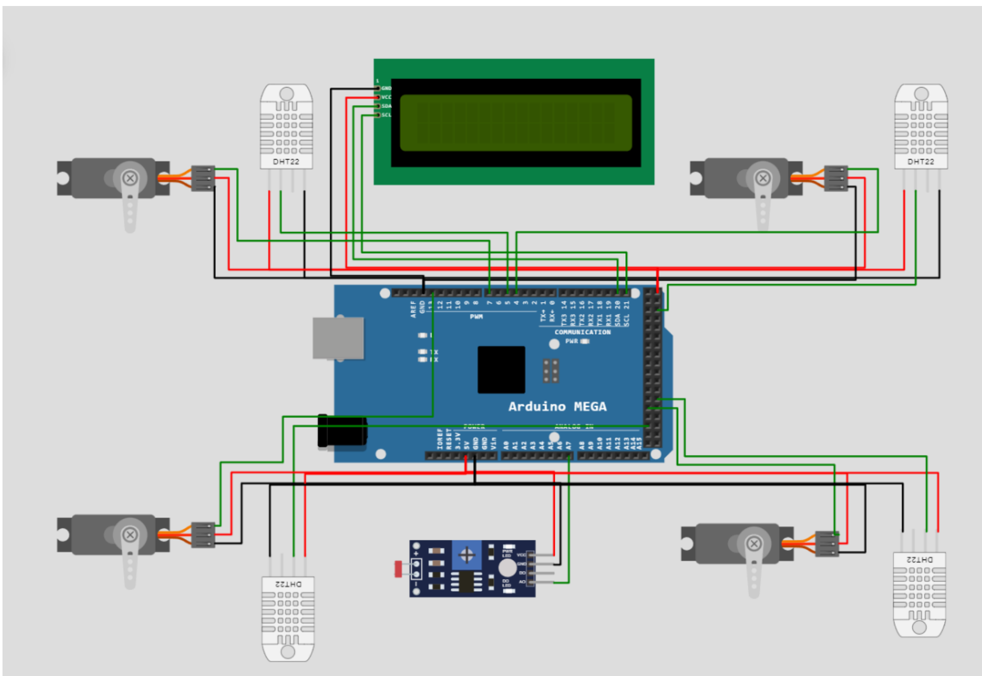


Figure 2: Snap shot of the final design implemented

Component Name	Pin Name	Description	Arduino Pins
DHT22			
	VCC	Positive Voltage	Unit 1: 5V.2, Unit 2: 5V.2, Unit 3: 5V, Unit 4: 5V
	SDA	Digital Data Pin (input/ output)	Unit 1: 5, Unit 2: 25, Unit 3: 50, Unit 4: 45
	GND	Ground	Unit 1: GND.1, Unit 2: GND.1, Unit 3: GND.2, Unit 4: GND.2
Servo			
	PWM	Servo control signal	Unit 1: 7, Unit 2: 4, Unit 3: 13, Unit 4: 46
	V+	Positive voltage (5V)	Unit 1: 5V.2, Unit 2: 5V.2, Unit 3: 5V, Unit 4: 5V
	GND	Ground	Unit 1: GND.1, Unit 2: GND.1, Unit 3: GND.2, Unit 4: GND.2
LDR			
	VCC	Positive power supply	5V
	GND	Ground	GND.2
	AO	Analog output	A7
LCD (l2c configuration)			
	GND	Ground	GND.1
	VCC	Supply voltage	5V.2
	SDA	I2C data line	20
	SCL	I2C clock line	21

Table 1: Description of the component pins and their location on arduino board

2.4 Working:

Our IoT based Smart farm will control flow for water in four units. That's why we will require 4 DHT22 sensors and 4 Servo Motors. Each DHT22 sensor will sense temperature in Celsius and humidity in percentage, this information will then be sent to the Arduino Mega Micro Controller. Arduino mega controller will use Photoresistor Sensor to find out whether its day time or night time. If the light value (lux) sensed by the sensor is less than 50 units it would imply that it is night time, and there should be no water flow. If lux is greater than 50 units, that means it is day time, in this case, we will use the temperature and humidity values and use a trained ML algorithm(this will be discussed in next section) to predict the required water flow percentage for that Unit. This water flow percent should then be used to control the degree of servo motor actuator so that correctly controls the water flow. The percentage of water flow should be proportional to the angle of servo motor. Percentage of water flow for every unit is also showed at the LCD display as output.

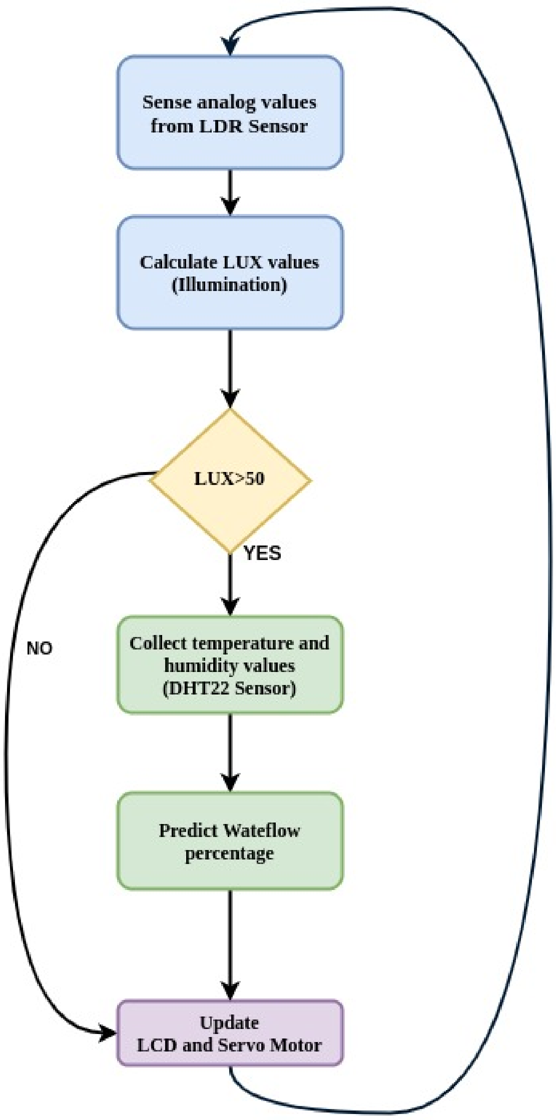
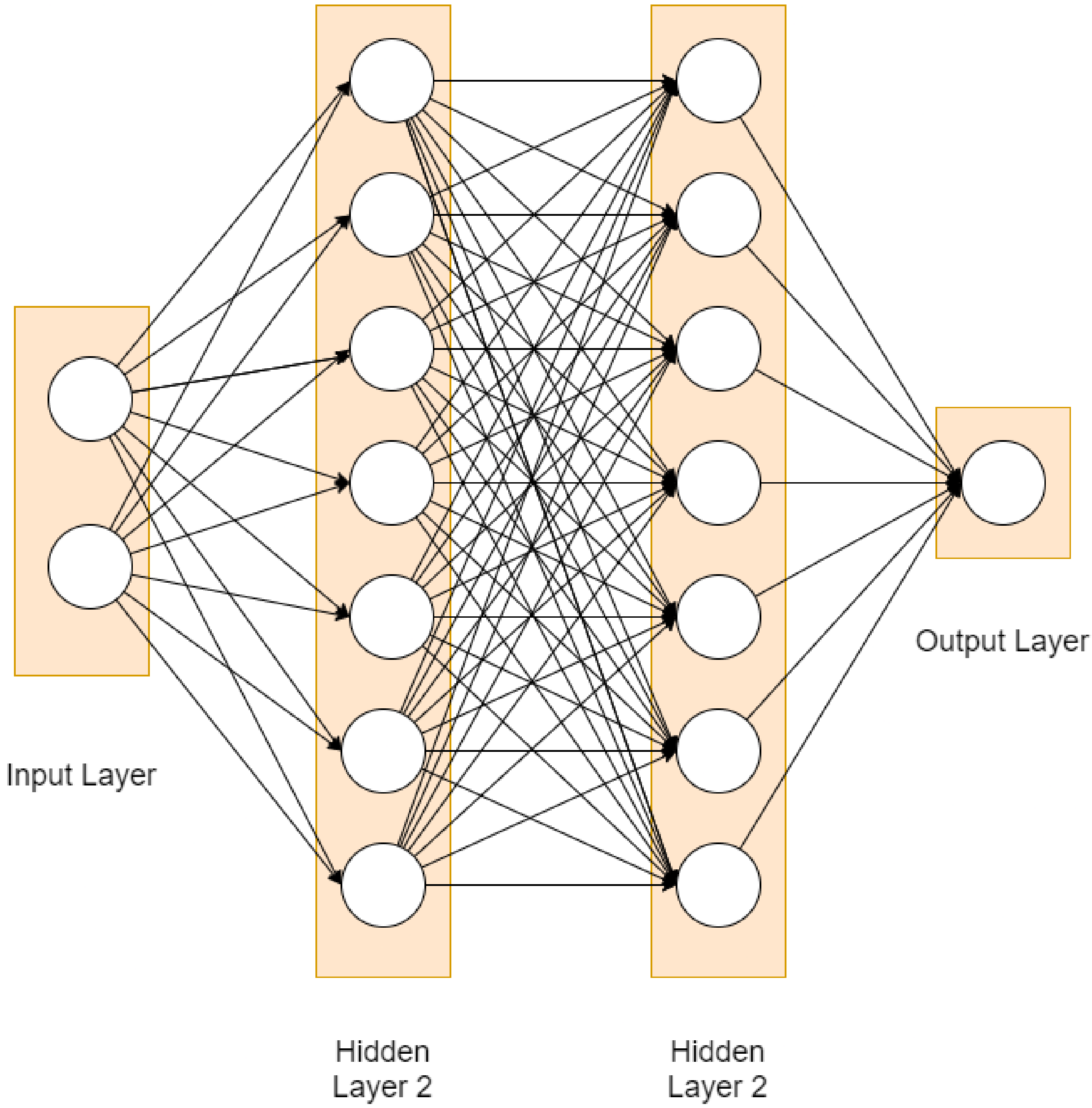


Figure 3: Flow chart describing the working of the system

3 ML Model

We have used ML algorithm to predict the required water flow percentage for each unit. Training of the algorithm is done with Python language to get the trained weights. These weights are then used in Arduino Mega to implement forward pass and get the prediction.

3.1 Neural Network Architecture



We have implemented Two-layer perceptron machine learning model. This model is used because it has the ability to learn and model non-linear and complex relationships and has the ability to work with insufficient knowledge[8]. Also, forward pass for prediction isn't a computationally expensive task to do so it works well on arduino mega. Description about the layers are as follows:

- One Input layer: Input layer consists of 2 nodes, one for temperature input and one for humidity input.
- Two Hidden Layers: Each hidden layer consists of 7 nodes. Each hidden layer node is connected with every nodes from its previous layers. Each of this connection has a weight associated with it. Activation function used at hidden layers is ReLU. Each hidden layer node will also have a bias term.
- Output Layer: Output layer is sigmoid activated layer, which receives inputs from hidden

layer 2 and outputs water percentage ratio between 0 to 1. This ratio will be converted to percentage by multiplying it with 100. It is known that scaling output doesn't affect the performance of the network. Output layer also has a bias term

In our architecture, the weights that connect the nodes and biases on each node are trainable. For our architecture there is a total of: 14(hidden layer 1 weights) + 49(hidden layer 2 weights) + 7(output layer weights) + 7(hidden layer 1 bias) + 7(hidden layer 2 bias) + 1(output bias) = 85 trainable parameters.

3.2 Implementation:

The architecture is implemented using Tensorflow Keras Library [6]. Data is divided into training set and test set with ratio 0.9:0.1. Output layer has sigmoid activation function. Sigmoid function gives output between 0 to 1. For our training and prediction, we will be converting water flow percentage to water flow to fraction, by dividing it by 100, i.e 50% will become, 0.5. This way we can use sigmoid activation to get water flow fraction, so using this we scale the output to get water flow percentage by multiplying the output by 100. Loss function used is "mean squared error" to optimize the model, and Adaptive Moment Estimation(adam) optimizer is used as optimizer.
Model is trained for 300 epochs with a batch size of 10. Weights obtained after training the model are used to implement forward pass in the arduino board.

3.3 Test Cases:

Following are the few Test cases and water flow output produced by the model.

Temperature	**Humidity**	**Waterflow %**
80	11.5	75
59.5	16.5	44
51	24.5	10.08
34	40	0.3

Table 2: Test Cases

3.4 Results:

After training the Two Layer Perceptron machine learning model for 300 epochs, we tested the performance on held out test set. We are able to achieve a root mean square error(RMSE) of 11.461 and R^2 value[7] of 0.905

4 Conclusion and Future works:

The smart irrigation system has been implemented successfully in accordance to the requirements provided by Aravind. The ML model has been implemented on the dataset and the weights are used for further calculations. The system can reduce the burden from humans to some good extent, when installed correctly. Aravind is happy with the newly built smart irrigation system.

Currently the system only uses two factors to predict the water flow. It can be improved further by deploying additional sensors and collecting some more relevant data like soil texture, soil dryness, precipitation level etc. There are tons of scope for improvement of the system with additional sensors.

References

[1] Wokwi-dht22 reference. URL https://docs.wokwi.com/parts/wokwi-dht22.

[2] Wokwi-lcd1602 reference. URL https://docs.wokwi.com/parts/wokwi-lcd1602.

[3] Wokwi-photoresistor-sensor reference. URL https://docs.wokwi.com/parts/wokwi-photoresistor-sensor.

[4] Wokwi-arduino-mega reference. URL https://docs.wokwi.com/parts/wokwi-arduino-mega.

[5] Wokwi-servo reference. URL https://docs.wokwi.com/parts/wokwi-servo.

[6] Module: Tf.keras tensorflow core v2.6.0. URL https://www.tensorflow.org/api_docs/python/tf/keras.

[7] Coefficient of determination, Oct 2021. URL https://en.wikipedia.org/wiki/Coefficient_of_determination.

[8] J. Mahanta. Introduction to neural networks, advantages and applications, Jul 2017. URL https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207#:~:text=Key%20advantages%20of%20neural%20Networks%3A&text=ANNs%20have%20the%20ability%20to,linear%20as%20well%20as%20complex.