A Novel Method for Monitoring and Control Greenhouse Farming Using IOT

Kavyashree T[1]
M.Tech Student: Department of Studies in Computer Science and Engineering
University B.D.T. College of Engineering
(A constituent college of Visvesvaraya Technological University)
Davangere, Karnataka, India

Naveen Kumar B[2]
Assistant Professor: Department of Studies in Computer Science and Engineering
University B.D.T. College of Engineering
(A constituent college of Visvesvaraya Technological University)
Davangere, Karnataka, India

Abstract— **Greenhouses structures are built to facilitate a controlled cultivation system in greenhouse farming; the structure must allow light penetration and maximum coverage. The objective of this project is to provide favorable growing conditions and to protect crops from unfavorable weather and various pests by using Internet of Things (IOT) with Machine Learning (ML) technique. The advantage of this intelligent system is that it enables effective management and additionally equipped offer the possibility to take complete control of the crop production inside greenhouse, which leads to faster growth and higher yields without human intervention. Incorporating ML technique with IOT system along with Thinkspeak based implementation will make an Intelligent Greenhouse Monitoring and Control System.**

Keywords— **Internet of Things (IOT), Machine Learning (ML), Thinkspeak IOT analytics, K-NEAREST NEIGHBOUR (KNN) algorithm.**

I. INTRODUCTION

Greenhouse farming is one of the popular and unique farming techniques where crops are grown within sheltered structures covered by a transparent material. Greenhouses are preferred to grow both seasonal and unseasonal fruits, vegetable and other crops. It are seen fully or partially covered in order to create the microclimate. Glass, wood, polyethylene sheets are most widely used materials to cover the house.

Controlling in climatic parameters plays a major role in greenhouse farming. Like heating is required during winter season. While in summer when ambient temperature rises above 35°C during daytime, the cooling of poly house is required. Micro irrigation or drip irrigation system for watering plants is needed. It is very hard for a farmer to have continuously monitored the parameters to keep greenhouse in the perfect condition.

The required conditions are:

- Optimum Temperature should be between 21°C to 30°C
- Humidity percentage should be between 84% to 95%
- Soil moisture percentage should always be more than 50%
- Light should be present for photosynthesis of plants
- Make sure to avoid extreme temperature which is likely to cause fire

It becomes complicated to maintain the above listed conditions manually. So there is a need of an automatic system which is intelligent enough to make predictions and to provide controlled favorable conditions that helps saplings to grow faster and healthier.

The main aim of our project is to make Internet of Things (IOT) to work with Machine Learning (ML) method to control the climatic factors like temperature and moisture level inside a greenhouse and protect the plants from the harsh external weather and pests. For instance, if temperature is high (say more than 30%) then exhaust fan will be ON which cools or decreases the temperature inside the house.

Our project is setup with 3 units, they are: hardware unit, software unit and a cloud database.

- Hardware Unit: It comprises of a microcontroller, input & output devices and a communication module. Hardware unit comprises of whole IOT system.
- Software Unit: Python programming and C++ programming which handshake themselves to monitor and control hardware unit is a software unit.
- Cloud Database: Thinkspeak is an online database that allows a channel to store data. It is an IOT analytics platform.

In the proposed system, machine learning algorithm is used to predict whether to water the plants or not, and it checks if cooling is required. Here we are using Arduino uno microcontroller which collects the data from sensors. The values acquired are then sent to Python ide. Python program which uses machine learning technique called KNN algorithm to predict the outcome and it also controls the output system. Wi-Fi module ESP8266 is used to save the sensor data on to Thinkspeak cloud database which can be used for future analysis.

II. Literature Survey

a) The implementation of a wireless sensor networks for greenhouse environment monitoring can be seen in the contribution [1]. Sensors are used to sense the parameters like temperature, humidity, soil moisture, color of leaves, fire and CO2 toxic gas detector. Microcontroller controls only water motor when the moisture level decreases and rest of the values will be messaged to farmer using GSM modem.

b) Monitoring and control more than one greenhouses using a web application with cloud technology is seen in the contribution [2]. The system is composed of two sections: sensors and actuators. The input of temperature, humidity, soil moisture and light intensity sensors are collected and sent to greenhouse web application. Then actuators are used to control the required output devices using fuzzy logic control. Similarly more than one greenhouse parameters are collected and are controlled remotely.

c) Android application based greenhouse monitoring and control project is seen in [3] model. ATmega328 microcontroller is used to control the system. Based on values from sensors the temperature, humidity, and light and soil moisture level are controlled using coolers, sprayers, bulbs and pumps. HC05 Bluetooth application is used to display sensors data and the data is then messaged to farmer via SMS through GSM communication module.

d) From the study of reference paper [4] it is observed that the greenhouse monitoring is being made via mobile application which is a login – password based implementation. The greenhouse system comprises of raspberry pi microcontroller, input sensors - DHT11, LDR, MQ5 and soils moisture, output devise - fan, water pump / sprinkler, light bulb and alarm. Microcontroller is the heart of the system, if it finds any imbalance in the parameters then relay will trigger the corresponding output electronic component to turn ON in order to satisfy the requirement of resources in the greenhouse environment.

Existing System

Farming is often done by the conventional methods where he has to irrigate by monitoring the land time to time. There exist automated IOT systems that have been deployed in farming which are used to capture the data and irrigate the field, as soon as the moisture has reached a specific threshold, the irrigation will stop. In these automated systems programmer have to mention the conditions explicitly for the system to act upon.

III. Methodology

Machine learning algorithms are programs that can learn from data and improve from experience, without human intervention. It aims at developing algorithms that can learn and create statistical models for data analysis and prediction. These algorithms should be able to learn by themselves - based on data provided make accurate predictions, without having been specifically programmed for a given task. We are incorporating KNN classification method (that calculates distance of neighbors using Euclidean distance formula) to build our system.

III.I KNN CLASSIFIER

K-Nearest Neighbor is one of the Machine Learning algorithms based on Supervised Learning technique. The main idea behind the algorithm is that the elements that are close to each other would belong to the same category. The two properties that define KNN are:

1. Since it does not have a specialized training phase and uses all the data for training while classification, it is called as a lazy learning algorithm.
2. This algorithm is considered as a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

Pseudocode:

Step1: Calculate d(x, xi) i =1, 2, ….., n;

Where [d] is the distance calculated using Euclidean distance between the training data and a new instance x

Step2: Sort the calculated n distance values in ascending (smallest to highest) order.

Step3: Choose the [k] value where the value is positive integer i.e. $k \geq 0$, take the first [k] distances from the sorted list

Step4: In those [k] sorted neighbors find the corresponding target class.

Step5: Let [ki] be number of neighbor points belonging to the ith class and [kj] belongs to jth class among chosen neighbors. Then x will be assigned with the class which has maximum number of occurrences.

Working:

The algorithm uses the entire data set as the training set. While classifying a new data instance, the algorithm goes through the entire data set to find the k- nearest instances which are most similar to the new instance. K: It is a numeric value that indicates numbers of neighbors to be consider while classifying the new instance.

There are many ways to calculate distance between neighbors, we are incorporating Euclidean Method.

Euclidean Distance Formula

To measure the distance between points X and Y in a feature space, various distance functions have been used in the literature, in which the Euclidean distance function is the most widely used one. Let A and B are represented by feature vectors X = (x1, x2, …, xm) and Y = (y1, y2, …, ym), where m is the dimensionality of the feature space.

To calculate the distance between X and Y, the normalized Euclidean metric is generally used by

Equation1 Euclidean distance formula

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

d (x,y) - two points in Euclidean n-space

xi, yi - Euclidean vectors of ith instance

n - n-space

III.II TRAINING DATA SET TO KNN ALGORITHM

KNN makes predictions by calculating the similarity between an input sample and each new instance and finds nearest neighbors.

Table1 below shows input training data used by KNN to predict the output

	A	B	C	D	E
1	Temperature	Humidity	Soil-Moisture	Water-Level	Target
2	28	75	76.34	300	no
3	28	63	76.34	330	no
4	26	77	76.44	290	no
5	27	78	76.54	300	no
6	27	60	75.12	333	no
7	26	72	43.37	250	yes
8	29	72	36.17	222	yes
9	26	79	27.27	200	yes
10	27	80	39.1	100	no
11	27	89	39.1	90	no
12	29	64	25.1	500	yes
13	27	65	50.18	600	yes
14	29	65	62.37	300	no
15	29	66	57.28	468	no
16	26	74	55.03	373	no
17	27	74	53.47	450	no
18	26	74	44.05	100	no
19	25	85	66.07	300	no

Training Data

Table1: Input training data

Field A: Temperature in percentage (%)

Field B: Humidity in percentage (%)

Field C: Soil moisture in Percentage (%)

Field D: Water level in Centimeter (cm)

Field E: Target Value

Accuracy of the training data is defined by model fit. Below snap shows accuracy percentage of the training data

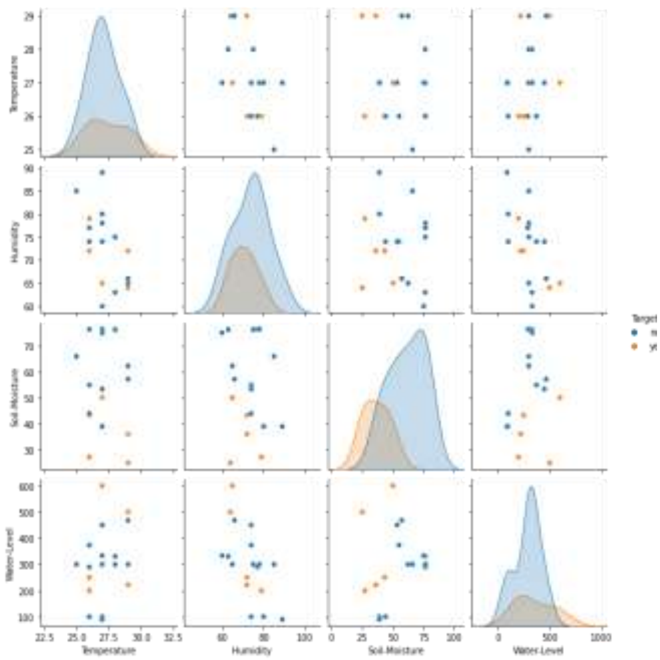
```
# evaluate predictions
accuracy = accuracy_score(y_test, yhat)
print('Accuracy: %.3f' % (accuracy * 100))

Accuracy: 66.667
```

Figure1: Shows accuracy percentage of the training data

III.III EXPLORATORY DATA ANALYSIS (EDA) OF TRAINING DATA:

Seaborn library in python is used to discover and visualize the relationships between the data. A pairs plot allows us to see both distribution of single variables and relationships between two variables.

The below graph contains scatter plots between each pair of feature columns against the target value column (i.e. yes/no). Feature columns are Temperature, Humidity, Soil moisture and Water level. The target values ‘yes’ and ‘no’ are indicated in orange and blue plots respectively.



Graph1: Feature Map Graph

We can infer from above graph by considering all numerical attributes, they are:

- ➢ When the temperature is between 26% and 28% the target value is most likely no.
- ➢ In humidity the target value is mostly yes when it is around 60% to 90%.
- ➢ When it comes to soil moisture, the target value is mostly yes when the moisture percentage is less than 60.
- ➢ The target value is yes when the water level is around 200 to 300cm

III.IV BLOCK DIAGRAM:

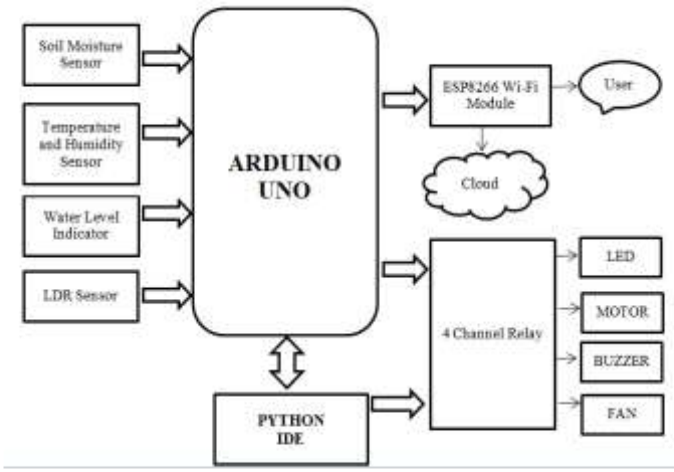


Figure.2: Block diagram of System

Figure.2 shows the System Block diagram which is composed of various elements, they are:

- The inputs devices: Temperature and Humidity sensor, Soil moisture sensor, Water level indicator and light intensity sensor.
- The output devices are: LED, Water motor, Buzzer, Exhaust fan.
- ESP8266 Wi-Fi module is a communication module.

The microcontroller collects the data from input sensors and sends them to python ide through USB port. Python takes the input and predicts the output with the help of machine learning technique, KNN algorithm. Besides prediction python also controls the IOT output devices accordingly. 4 channel relay is used to switch ON/OFF output devices in the system.

III.IV HARDWARE AND SOFTWARE SPECIFICATIONS:

Hardware components used to implement the system:

1. ARDUINO UNO microcontroller
2. Soil moisture sensor
3. DHT11 sensor
4. Water level indicator
5. Breadboard and jump wires

6. Electronic output devices: Led, Buzzer, Exhaust fan and Water motor

<u>Software used:</u>

1. Arduino IDE version 1.8.12

   Arduino IDE is an open source environment where code is written and uploaded to Arduino board. It supports C and C++ programming languages.

2. Python 2.7 IDE

   Python is a high level programming language which is used to predict the watering, interact with Arduino ide and control the IOT system.

III.V FLOW CHART:

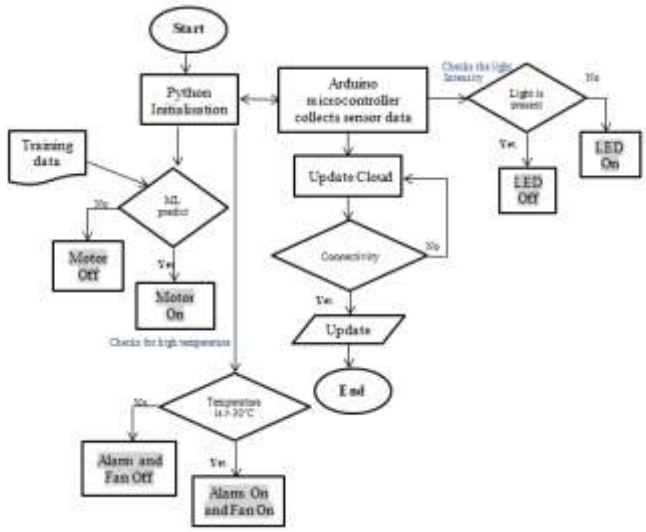


Figure.3: Dataflow Diagram

Figure.3 describes the dataflow diagram which is a method that gives information about the yields and contributions of every entity and the procedure. The flow of information from collecting input from sensors to processing them is depicted below.

The various steps involved are as follows:

1. The microcontroller collects current temperature percentage, humidity percentage, soil moisture percentage, water level in the tank and light intensity inside the greenhouse from the input sensors and sends the required data to python ide through USB port.

2. Python processes these inputs using with the help of machine learning technique. KNN algorithm uses training data that has been provided to make the prediction weather watering is required or not.

3. Python also controls other output devices like exhaust fan and buzzer if in case of high temperature inside greenhouse.

4. 4 channel relay is used as switch to ON/OFF the output devices.

5. ESP8266 Wi-Fi module updates cloud with the input data received by Arduino.

IV. CIRCUIT CONNECTIONS AND RESULTS:

IV.I CIRCUIT CONNECTIONS

IV.I.I ESP8266 Connection with Arduino Board

Figure.4: *Shows ESP8266 Wi-Fi connections with Arduino Uno board*

Above figure shows connection of wifi module with Arduino board. The connections are as follows:

- pin 1: Rx is connected to Rx of Arduino
- pin 2: GPIO 0 is connected to ground while uploading the code to arduino IDE
- pin 3: GPIO 2 is not used
- pin 4: GND is connected to ground of Arduino board
- pin 5: Tx is connected to Tx of Arduino
- pin 6: CH_PD(EN) is connected to 3.3v power supply
- pin 7: RST(reset) is not used
- pin 8: Vcc is supplied with 3.3v from Arduino.

IV.I.II Water Level indicator and water motor

Water level indicator is used to check the water level in the tank.

Water motor is used to pump the water to plants. It is a submersible pump

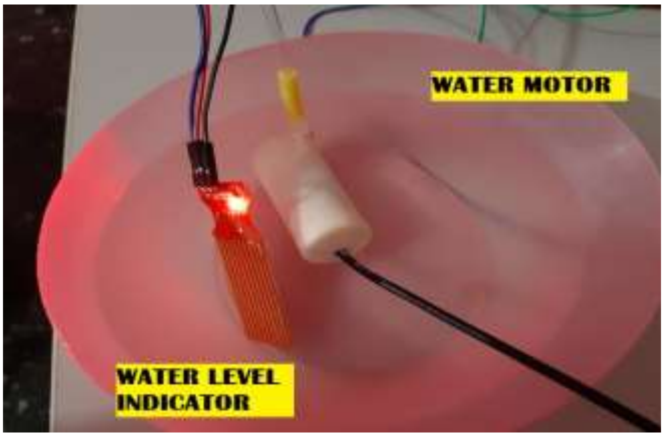


Figure5: Shows water level indicator and water motor in the tank

IV.I.III Prototype of Greenhouse Farming Model

Figure 6: Shows a prototype of greenhouse farming model.

IV.II RESULTS:

1. Below screenshot shows python receiving Arduino's input and its predictions values at different K- values

Case1: Prediction of Watering for K=5

```
>>> ================================ RESTART ===========
>>>
Hi!, Welcome to Greenhouse Farming

Input from Arduino is: (23.0, 95.0, 8.7, 663.0)
[yes] => 0
[no] => 1
Data=(23.0, 95.0, 8.7, 663.0), Predicted: 0
Watering is required, MOTOR ON
Temperature is high, FAN ON
Checking the light intensity
Light is absent
Updating Thinkspeak cloud
>>>
```

New Input values are: [23.0, 95.0, 8.7, 663.0] which corresponds to [Temperature, Humidity, Soil Moisture, Water Level] values. Soil moisture value is LOW, KNN predicts as watering is required.

Case2: Prediction of Watering for K=2

```
>>> ================================ RESTART ===========
>>>
Hi!, Welcome to Greenhouse Farming

Input from Arduino is: (23.0, 95.0, 8.31, 687.0)
[yes] => 0
[no] => 1
Data=(23.0, 95.0, 8.31, 687.0), Predicted: 0
Watering is required, MOTOR ON
Temperature is high, FAN ON
Checking the light intensity
Light is absent
Updating Thinkspeak cloud
```

New Input values are: [23.0, 95.0, 8.31, 6687.0] which corresponds to [Temperature, Humidity, Soil Moisture, Water Level] values. Soil moisture value is LOW, KNN predicts as watering is required.

Both in case1 and case2 it is observed that the prediction of watering did not change for different values of K.

2. Below screenshot shows python prediction of watering not required

```
>>> ================================ RESTART =========
>>>
Hi!, Welcome to Greenhouse Farming

Input from Arduino is: (23.0, 95.0, 52.49, 670.0)
[yes] => 0
[no] => 1
Data=(23.0, 95.0, 52.49, 670.0), Predicted: 1
No need of watering
Temperature is high, FAN ON
Checking the light intensity
Light is absent
Updating Thinkspeak cloud
>>> |
```

New Input values are: [23.0, 95.0, 52.49, 670.0] which corresponds to [Temperature, Humidity, Soil Moisture, Water Level] values. Soil moisture value is HIGH, KNN predicts as watering is not required.

✓ **Mathematical calculations for KNN predictions using Euclidean distance formula for the new instance [23.0, 95.0, 52.49, 670]: (K=5)**

A	B	C	D	E	F
Temperat	Humidity	Soil-Moist	Water-Le	**Euclidean distance for new instance [23.0, 95.0, 52.49, 67**	Target
28	75	76.34	300	370.93	no
28	63	76.34	330	342.37	no
26	77	76.44	290	381.19	no
27	78	76.54	300	371.32	no
27	60	75.12	333	339.59	no
26	72	43.37	250	420.7	yes
29	72	36.17	222	448.9	yes
26	79	27.27	200	470.95	yes
27	80	39.1	100	570.36	no
27	89	39.1	90	580.19	no
29	64	25.1	500	175.06	yes
27	65	50.18	600	76.29	yes
29	65	62.37	300	371.39	no
29	66	57.28	468	204.21	no
26	74	55.03	373	297.76	no
27	74	53.47	450	221.03	no
26	74	44.05	100	570.45	no
25	85	66.07	300	370.38	no

Nearest 5 values are:

Euclidean distance for new instance [23.0, 95.0, 52.49, 670]	Target
175.06	yes
76.29	yes
204.21	no
297.76	no
221.03	no

From the above mathematical calculation it is observed that among 5 nearest neighbors 3 neighbors belongs to class "NO" 2 other neighbors belongs to class "YES", hence on majority basis the new instance is classified as class "NO". From the above screen shot it is seen that the prediction is no. Hence the calculation is mathematically proved and it's correct.

3. Command prompt shows the output of WIFI updating Thinkspeak cloud successfully

```
ocommand => AT Success
1. at command => AT+CWMODE=1 Success
2. at command => AT+CWJAP="AndroidAPEA1A","kavyata7" Success
3. at command => AT+CIPMUX=1 Success
4. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Fail
0. at command => AT+CIPSEND=0,55 Success
2. at command => AT+CIPCLOSE=0
Autoscroll   Show timestamp        Both NL & CR   9600 baud   Clear output
```

Above output shows output result (success / failure) of ESP8266 Wi-Fi connections with Thinkspeak cloud. Success indicates AT command is executed. If any command fails, the Wi-Fi module retries to connect until sending of data to the cloud is successful.

4. ThinkSpeak Cloud showing graph of Temperature, Humidity, Soil Moisture and water level ate different time.

Field1: Soil Moisture – X axis shows time in hours and Y axis Moisture in percentage

Field2: Water Level – X axis shows time in hours and Y axis Water level in Centimeter

Field3: Temperature – X axis shows time in hours and Y axis Temperature in degree Celsius

Field4: Humidity – X axis shows time in hours and Y axis Humidity in percentage

CONCLUSION

The main aim of this contribution is to make irrigation process flexible, time saving and more efficient than the existing automated IOT systems. On basis of real time sensor data ML method will do predictions which enables continuous monitoring of water flow.

Overall this intelligent system eliminates much of the manual works which are a stress free for a farmer and mainly it avoids over irrigation and under irrigation state which in turn reduces the wastage of water and keeps constant soil moisture in the field. The main advantage is that the system's action can be implemented with low power, low cost, small size, robust and highly versatile. High sensitive sensors can be implemented i.e. soil moisture sensor that covers large area of acre to sense the moisture percentage, for large areas of land that uses greenhouse farming.

The future scope of this project would be as follows:

- Since the whole circuit works only on power supply, in case of power cuts or damage a battery or solar power unit can be implemented as an alternative source.
- The smoke sensors can used to send emergency information to user in case of fire in field or burning of motor.
- KNN machine learning method which is also called lazy learner can be replaced with eager learner method like decisions trees for more accurate and faster predictions.

REFERENCES

[1] Greenhouse Monitoring and Control Based on IOT Using WSN, Dept. of ECE, RRCE, Bangalore. http://www.irdindia.in/journal_itsi/pdf/vol4_iss3/15.pdf

[2] IoT Based Intelligent Greenhouse Monitoring and Control System, University of Basrah, Iraq. https://www.researchgate.net/publication/322539575

[3] Internet of Things Based Smart Greenhouse: Remote Monitoring and Automatic Control, Monash University, Australia. https://www.researchgate.net/publication/330859093

[4] Intelligent greenhouse monitoring and control scheme: An arrangement of Sensors, Raspberry Pi based Embedded System and IoT platform, Indian Journal of Science and Tech0logy. https://indjst.org/articles/intelligent-greenhouse-monitoring-and-control-scheme-an-arrangement-of-sensors-raspberry-pi-based-embedded-system-and-iot-platform

[5] Machine Learning by Tom Mitchell. https://books.google.co.in/books/about/Machine_Learning.html

[6] https://www.arduino.cc

[7] Arduino Soil Moisture Sensor: https://create.arduino.cc/projecthub/MisterBotBreak/how-to- use - a-soil-moisture-sensor-ce769b

[8] https://www.youtube.com

[9] https://www.google.com

[10] Python in practice, www.it-ebooks.info